

**Information Management**



# **DB2 system topology and configuration for automated multi-site HA and DR**

February 2010

Aruna De Silva  
Steve Raspudic  
Sunil Kamath  
IBM Toronto Lab

## Table of contents

<b>Summary</b>	<b>1</b>
<b>Introduction – general concepts</b>	<b>2</b>
<i>High availability and disaster recovery</i>	2
<i>DB2 High Availability Disaster Recovery (HADR) configuration</i>	2
<i>Automating DB2 HADR – DB2 integrated HA solution</i>	3
<i>Before you begin</i>	4
<i>Command conventions used in this document</i>	6
<b>Introduction - technical solutions</b>	<b>7</b>
<i>Three-node multi-site cluster with arbitrator node in third site</i>	7
<i>Two-node cluster with shared disk tiebreaker in third site</i>	8
<b>Initial configuration – common to both topologies</b>	<b>10</b>
<i>Setting up the basic network</i>	10
<i>A note on host names</i>	11
<i>Port requirements to run a multi-site cluster</i>	12
<i>Configuring DB2 HADR</i>	13
<i>Importance of clock synchronization between HADR servers</i>	14
<i>Considerations for configuring the HADR_TIMEOUT and HADR_PEER_WINDOW</i>	15
<i>A note on HADR performance</i>	16
<i>Using Automatic Client Reroute (ACR) with DB2 HADR</i>	18
<i>Setting up logging – SYSLOGs</i>	19
<i>Creating the initial two-node DB2 HADR cluster using db2haicu</i>	20
<i>A note on RSCT configuration to address resource contention</i>	21
<b>System configuration: three-node DB2 HA cluster topology</b>	<b>22</b>
<i>Adding the arbitrator node to the cluster using db2haicu</i>	22
<i>Configure network equivalencies or “db2haicu networks”</i>	23
<i>Testing three-node DB2 HA cluster topology</i>	26
<b>System configuration: DB2 HA cluster topology with shared DISK tiebreaker</b>	<b>27</b>
<i>Adding the shared DISK tiebreaker to the cluster</i>	27
<i>Configure network equivalencies or “db2haicu networks”</i>	30
<i>Testing two-node DB2 HA cluster with shared DISK tiebreaker</i>	30
<i>A note about iSCSI security</i>	30
<i>Using a Fibre Channel LUN as the shared DISK tiebreaker device</i>	31
<b>Alternative multi-site DB2 HA topologies</b>	<b>32</b>
<i>A topology using a private inter-site network trunk</i>	32
<i>A cross-site subnet topology using a Virtual IP (VIP)</i>	32

<b>Managing long-distance multi-site DB2 HA clusters</b> .....	<b>34</b>
<i>Monitoring RSCT cluster communications</i> .....	34
<i>Tuning RSCT Communication Group parameters</i> .....	35
<i>Monitoring Cluster Quorum Status</i> .....	36
<i>Monitoring operational status of RSCT Topology/Group Services</i> .....	38
<i>Disabling High Availability</i> .....	39
<i>Manual takeovers</i> .....	39
<i>The db2haicu maintenance mode</i> .....	39
<i>Stopping and starting the entire RSCT domain</i> .....	40
<i>Removing a RSCT peer domain</i> .....	41
<i>Troubleshooting unsuccessful failovers</i> .....	42
<b>Appendix A - List of references for more information</b> .....	<b>43</b>
<b>Appendix B - DBA's checklist to cluster planning</b> .....	<b>44</b>
<b>Appendix C - Configuring EtherChannel</b> .....	<b>45</b>
<b>Appendix D - db2haicu XML input file</b> .....	<b>46</b>
<b>Appendix E - Configuring NetApp iSCSI shared LUN for the two AIX hosts</b> .....	<b>48</b>
<b>Appendix F - Testing three-node DB2 HA cluster topology</b> .....	<b>55</b>
<b>Appendix G - Testing two-node DB2 HA cluster with shared DISK tiebreaker</b> .....	<b>62</b>
<b>Appendix H - Tivoli SA policies and scripts used by DB2 Integrated HA solution</b> .....	<b>70</b>
<b>Notices</b> .....	<b>72</b>
<i>Trademark acknowledgments</i> .....	73



## Summary

---

Today's highly competitive business marketplace leaves little room for error in terms of availability, continuous operations, or recovery in the event of an unplanned outage. In today's connected world, an event that makes business data unavailable, even for relatively short periods of time, can have a major impact.

Today's mission-critical database systems must operate with the highest degree of availability possible. As databases increase in size, and ad hoc queries place greater demand on the availability of the system, the time and hardware resources required to back up and recover databases grow substantially while the maintenance windows either shrink drastically or disappear.

In this paper, we demonstrate how IBM® DB2® Version 9.5 for Linux®, UNIX®, and Windows® software (DB2 9.5), IBM Tivoli® System Automation for Multiplatforms software, and IBM Reliable Scalable Cluster Topology software (RSCT) are combined to provide highly available advanced database technical architectures to meet these demands. While the topologies we implemented in this paper are based on the DB2 9.5 release, you can also use IBM DB2 Version 9.7 for Linux, UNIX, and Windows software (DB2 9.7), without any changes.

In this paper, we demonstrate how to use these features to create automated High Availability Disaster Recovery solutions spanning long distances (even across multiple time-zones) to meet the increasing availability requirements demanded in today's business environment.

## Introduction – general concepts

---

This paper describes two distinct approaches to creating an automated High Availability Disaster Recovery (HADR) configuration using IBM DB2 Enterprise Server Edition Version 9.5 for Linux, UNIX, and Windows software (DB2 9.5 ESE) combined with Tivoli System Automation for Multiplatforms software and Reliable Scalable Cluster Topology (RSCT). The setup shown here is appropriate for maintaining a HADR site at a different location from the primary site, and will perform well even when sites are located across the globe in different time zones.

Two key approaches discussed in this paper are:

- **Three-node DB2 HADR configuration** (third cluster node located in site three is used as the arbitrator)
- **Two-node DB2 HADR configuration** (shared DISK located in site three is used as the tiebreaker)

*Note that both of the above configurations have three physical sites.* Both of these configurations are designed to recover from two types of disasters: 1) those that may affect an entire site, such as a site power failure or site communication outage, and 2) those that affect the node only, such as an operating system crash, machine failure, or machine network card failure. These configurations can also be used for planned shutdowns, such as for system or hardware changes to the primary system.

## High availability and disaster recovery

---

*High availability* is the term that is used to describe systems that run and are available to customers most of the time. For this to occur:

- During peak operating periods, transactions must be processed efficiently without sacrificing performance even during a loss of availability.
- Systems must be able to recover quickly when hardware or software failures occur or when disaster strikes.
- Software that powers the enterprise databases must be continuously running and available for transaction processing.

*Disaster recovery* is the ability to recover a data center at a different site, on different hardware, if a disaster destroys the primary site or renders it inoperable.

## DB2 High Availability Disaster Recovery (HADR) configuration

---

The DB2 HADR feature (DB2 HADR), is a database log replication feature that provides a high availability solution for both partial and complete site failures. HADR protects against data loss by continually replicating data changes from a source (primary) database, to a target (standby) database. Furthermore, one can seamlessly redirect clients that were using the original primary database to the standby database (which becomes the new primary database) by using Automatic Client Reroute (ACR) and reconnect logic in the application.

With DB2 HADR, you can choose a level of protection from potential loss of data by specifying one of three synchronization modes: *synchronous*, *near synchronous*, or *asynchronous*.

More information about DB2 HADR can be found here:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.ha.doc/doc/c0011267.html>

Figure 1 below illustrates the concept of HADR.

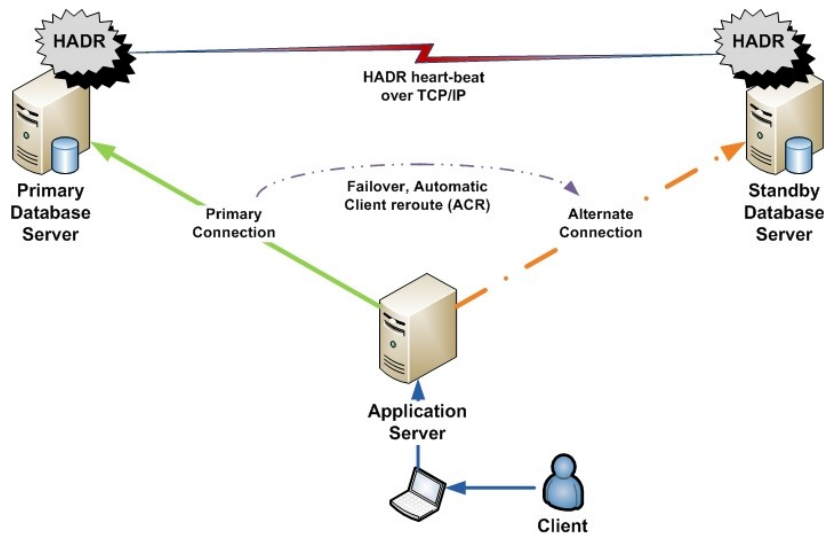


Figure 1. Concept of HADR in a DB2 9.5 environment

## Automating DB2 HADR – DB2 integrated HA solution

While HADR provides an elegant solution for maintaining a “hot-standby”, takeover operations and the subsequent reintegration process require manual intervention (i.e., someone has to issue the relevant HADR commands) in a disaster scenario. Hence, an automation framework (such as a cluster-manager-driven automation) or custom scripts can be used to provide HADR automated failover.

New to DB2 9.5, the DB2 High Availability (HA) feature enables integration between the IBM Data Server and cluster managing software and thus provides a unified HADR automation framework.

The DB2 9.5 HA feature provides infrastructure for enabling the database manager to communicate with the cluster manager when instance configuration changes, such as stopping a database manager instance, require cluster changes. In other words, it eliminates the need to perform separate cluster operations manually.

The DB2 HA feature is composed of the following key elements:

- **IBM Tivoli System Automation for Multiplatforms base component** is bundled with IBM Data Server on IBM AIX® and Linux as part of the DB2 HA feature, and integrated with the DB2 installer. We abbreviate this as “TSA” in this document.
- **The DB2 cluster manager API in the HA feature** defines a set of functions that enable the database manager to communicate configuration changes to the cluster manager. Therefore, for any database manager instance configuration and administration operation that requires cluster

changes, the database manager automatically issues a request to the underlying cluster manager to perform the required cluster configuration changes.

- **DB2 High Availability Instance Configuration Utility (db2haicu)** is a text-based utility that you can use to configure and administer your highly available databases in a clustered environment. db2haicu collects information about your database instance, your cluster environment, and your cluster manager by querying your system. You supply more information through parameters to the db2haicu call, an input file, or at runtime by providing information at db2haicu prompts.

See the white paper referenced in [Appendix A: Reference 1 - Automated Cluster Controlled HADR \(High Availability Disaster Recovery\) Configuration Setup using the IBM DB2 High Availability Instance Configuration Utility \(db2haicu\)](#) for more details about this solution.

That white paper discusses the automated HADR failover solution in a local site context and is critical to understanding the terminology and configurations described herein.

In [Appendix F – Tivoli SA policies and scripts used by DB2 integrated HA solution](#), we have included a brief description of the TSA policies and control scripts used under the covers by the DB2 integrated HA solution.

## Benefits of using DB2 9.5 integrated HA solution as an automated HA and DR solution

DB2 9.5 using the integrated HA solution offers the following features and benefits:

- High data availability.
- High performance.
- Flexible configurations.
- Automatic recovery from a component or link failure.
- Significantly reduced recovery time after a disaster.
- Increased integrity of recovery procedures.
- Reduced disaster recovery complexity, planning, testing, etc.
- Ability to perform well in a multi-site configuration that spans multiple time zones.

## Before you begin

---

Below you will find information about knowledge requirements, as well as hardware and software configurations used to set up the topologies depicted in the sections that follow. It is important that you read this section prior to beginning any setup.

### Knowledge prerequisites

- Understanding of DB2 9.5 and HADR.
- Understanding of high availability and disaster recovery concepts.
- Basic understanding of operating system concepts.

DB2 system topology and configuration for automated multi-site HA and DR

- A good understanding of TCP/IP networking concepts and the network infrastructure of your sites.

Information about configuring DB2 HADR can be found here:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.ha.doc/doc/t0051350.html>

## Software versions used in setup

For both topologies covered in detail in later sections, the same software configuration was used:

- DB2 9.5 Fix Pack 3
- TSA version 2.2.0.7 (included and bundled with DB2 9.5.0.3)
- AIX 5.300-09-02
- RSCT fileset version 2.4.10.0

## Recommended and minimum software levels

Your setup may have or need to use different levels than the ones we used. The following table shows which software versions can be used to successfully implement a similar solution.

Software levels	RSCT levels by OS (AIX 5.3)		RSCT levels by OS (AIX 6.1/ Linux)			DB2 9.5	DB2 9.7	TSA
	AIX 5.3	RSCT	AIX 6.1	Linux	RSCT			
<b>Minimum</b>	5.3-TL9-SP2	2.4.10.0	6.1-TL2-SP1	RHEL 5 Up2 SLES 10 Sp2 SLES 11	2.5.2.0	9.5 FP3	9.7 GA	2.2.0.7
<b>Current / Recommended</b>	5.3-TL10-SP1	2.4.11.4	6.1-TL3-SP2	RHEL 5 Up3 SLES 10 Sp2 SLES 11	2.5.3.4	9.5 FP5	9.7 FP1	3.1.0.4

### Notes:

- Refer to the DB2, Cluster, and System p<sup>®</sup> and AIX Information Centers for the latest software levels.
- For Linux, Solaris, and Windows platforms, TSA is bundled with RSCT filesets.
- Though our implementation was based on DB2 9.5, you can use DB2 9.7 instead, with the instructions we have provided in this paper to implement such cluster topologies.

## Planning for long-distance multi-site cluster topologies

Proper planning is the key to successful implementation and management of long-distance multi-site HA clusters. This white paper will provide extensive assistance, hints, and tips that an implementer will need to consider before embarking on the actual implementation tasks.

One key difference between a multi-site automated HADR cluster and the local automated HADR cluster is the increased requirement to understand and carefully consider the networking parameters and topology. We provide a quick reference checklist ([Appendix B - DBA's checklist to cluster planning](#)) to assist DB2 database administrators (DBAs) to understand what information they need to gather before proceeding to the setup and configuration phase.

## Command conventions used in this document

---

- Commands that must be run by root will be shown with a shell prompt of `root>`.
- Commands that must be run by the database user (the instance owner or database administrator) will be shown with a shell prompt of `dba>`.
- Commands **in boldface** are shown as the actual commands issued on one (or any number) of the systems.

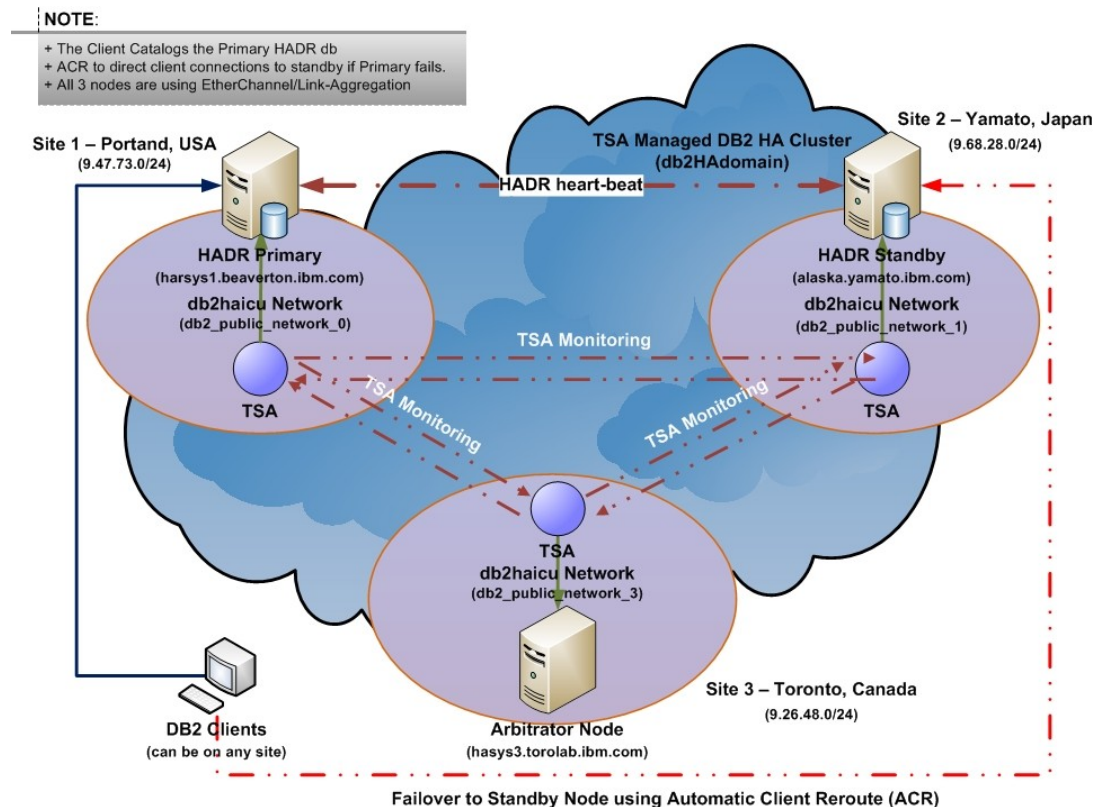
## Introduction - technical solutions

This section provides an overview of the two topologies, which are described in much greater detail in the remainder of this white paper. The two topologies differ technically in how the internal cluster quorum (sometimes called arbitration or tiebreaker) functionality is achieved. In the first case, an additional machine is used to achieve the arbitration task. In the second case, a shared resource (a shared disk) is used to achieve the arbitration task. *Note that the arbitrator (of either type) must not be physically collocated with either the HADR primary or HADR standby nodes.* In other words, it must be placed on a separate site.

### Three-node multi-site cluster with arbitrator node in third site

A key role of cluster software is to detect failures and take appropriate action. In order to be able to effectively and properly distinguish between various failures, a third cluster node can be used. This third cluster node arbitrates, or decides, the cluster state. This type of cluster configuration is commonly called a “*Majority Node Set*” cluster (or MNS for short). Note that this arbitration, once configured, is transparent to the end user or even to the applications running within the cluster. This kind of arbitration is especially useful when nodes in the cluster are separated by significant distances, as in extended distance clusters or metropolitan clusters. Arbitrator nodes may be configured to run non-clustered applications, or they can be set up purely as arbitrators, with no other applications running other than the clustering software (TSA in this case).

The use of an arbitrator node is shown in Figure 2. The server in site three is an arbitrator node.



**Figure 2. Three-node multi-site DB2 HADR cluster topology**

DB2 system topology and configuration for automated multi-site HA and DR

## Two-node cluster with shared disk tiebreaker in third site

As stated above, a key role of cluster software is to detect failures and take appropriate action. In order to be able to effectively and properly distinguish between various failures, a shared disk device can also be used. In this instance, a “*Quorum Device Tiebreaker*”, such as a shared disk, is used in place of the third cluster node. Note that the DISK tiebreaker type we used in our configuration is specific to AIX environments. For Linux environments, the tiebreaker type SCSI can be used instead. Shared disk tiebreaker is not supported on the Solaris operating system.

This tiebreaker type enables one to specify a SCSI or SCSI-like physical disk using an AIX device name, and assumes that the SCSI disk is shared by one or more nodes of the peer domain. Tiebreaker reservation is done by the SCSI reserve or persistent reserve command. Physical disks attached using Fibre Channel, iSCSI, and Serial Storage Architecture Connections are suitable.

The use of a DISK tiebreaker quorum device is shown in Figure 3. The single shared iSCSI LUN in site three is a DISK tiebreaker device.

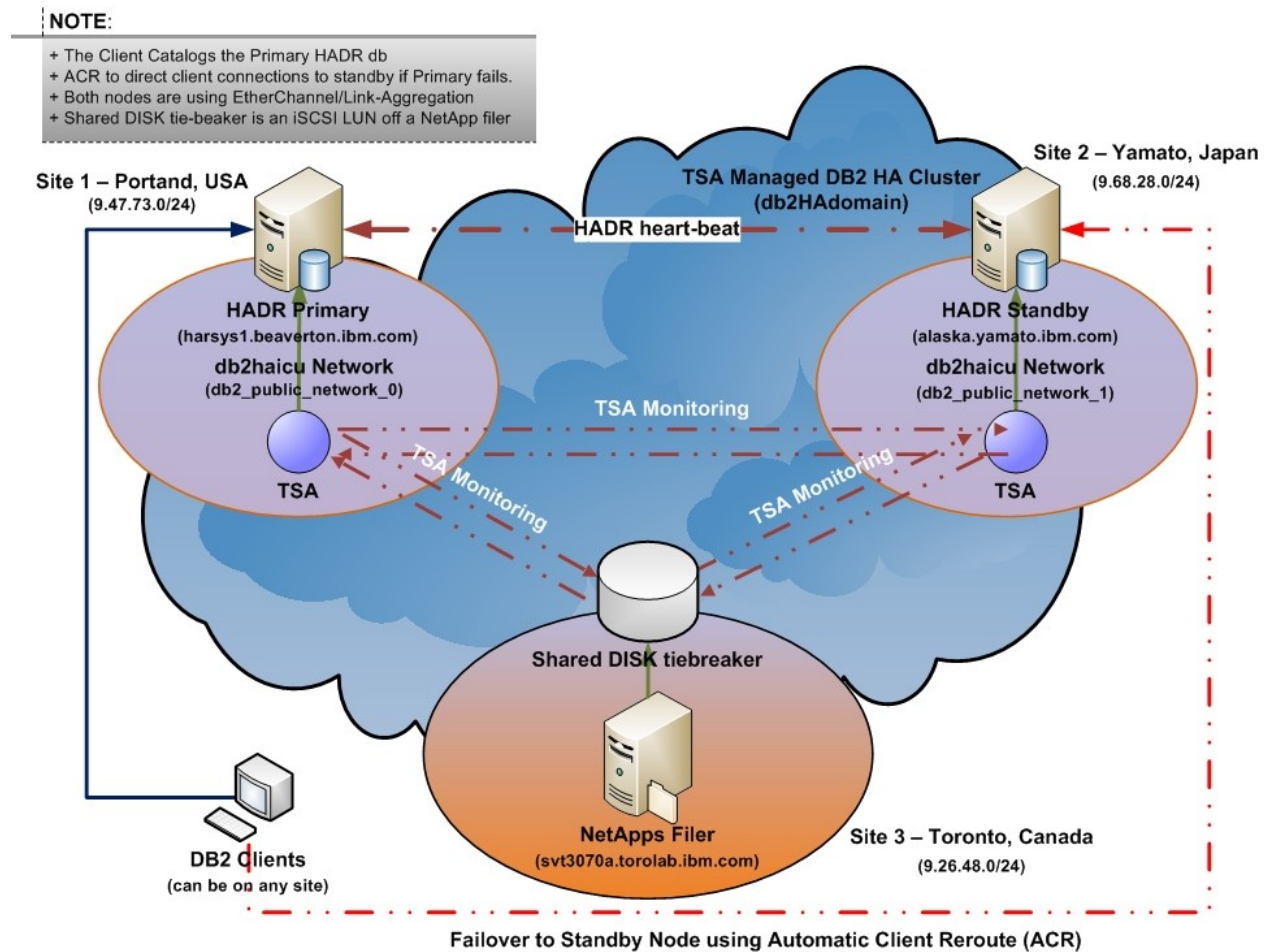


Figure 3. Two-node multi-site DB2 HADR cluster topology with shared DISK tiebreaker in third site

The usage of the disk tiebreaker is transparent to the end user (and to the applications using that cluster), but it can be helpful to briefly review the mechanism of the shared disk for cluster arbitration. In cases of potential node or network outage, the cluster software running on each of the active nodes will both attempt to acquire a SCSI reservation lock on the shared disk. Whichever node is able to achieve this lock first becomes the master. Any node that cannot acquire this lock is assumed to be out of quorum and will not be eligible to run cluster applications.

In this example of shared disk tiebreaker topology, we chose to use an iSCSI LUN from NetApp storage. However, the same concepts and procedures apply to storage attached using Fibre Channel or FCP LUNs as well. The only requirement in that case is the ability to do Storage Area Network (SAN) zoning across physical sites.

RSCT recently introduced support for SCSI-3 Persistent Reserve (PR) for DISK tiebreakers on AIX. However, to use SCSI-3 PR, the storage must support the required service actions and must be tested before being deployed into production environments. See the following IBM Cluster Information Center topic for more information about RSCT SCSI-3 PR support:

[http://publib.boulder.ibm.com/infocenter/clresctr/vrx/topic/com.ibm.cluster.rsct\\_5200\\_09.vsd.doc/b1506\\_undperr.html](http://publib.boulder.ibm.com/infocenter/clresctr/vrx/topic/com.ibm.cluster.rsct_5200_09.vsd.doc/b1506_undperr.html)

## **Using a network tiebreaker**

Network tiebreakers (or network quorum) can also be used in the absence of a third arbitrator node. While disk tiebreaker offers the fastest detection of node failure in the cluster thereby providing improved availability, a network tiebreaker is also a viable, reliable, and a fast alternative.

A network tiebreaker is a pingable IP address that is used to decide which node in the cluster will serve as the “active” node during a site failure, and which nodes will be offline. Note that the machine hosting this IP address does not need any particular software or operating system level installed; its primary requirement is that it can be pinged from all nodes in the cluster, and must remain pingable in the case of cluster node failures.

Typically, the default gateway of primary server site can be used as the network quorum device since it is reachable by both nodes.

## Initial configuration – common to both topologies

---

A significant portion of the configuration is common to both topologies. Hence, the following common configuration tasks are described under this section.

1. Setting up the basic network
2. Configuring DB2 HADR
3. Setting up logging – SYSLOGs
4. Creating the initial two-node HADR cluster using `db2haicu`

## Setting up the basic network

---

All machines used in both of our topologies contain two network interfaces each. For AIX environments, we will take the network adapters to be named `en0` and `en1` (same naming on each machine). Then we configure the network interface `en2` as the EtherChannel (or Ethernet Bonding in Linux) with `en0` as the primary adapter and `en1` as the backup adapter. We will take it that this EtherChannel adapter `en2` is the “public” adapter (which must be connected to the public network). Note that for Linux, the adapters are generally named `eth0` (and `eth1`), and for Solaris, adapters are generally named `hme0` (and `hme1`). Keep that in mind as you read the following example if you are working with Linux or Solaris environments.

1. Configure EtherChannel using the two network interfaces on each node. See [Appendix C – Configuring EtherChannel](#) for the details.

**Note:** You can use any supported Ethernet adapter in an EtherChannel. However, the Ethernet adapters must be connected to a switch that supports EtherChannel. See the documentation that came with your switch to determine if it supports EtherChannel (your switch documentation may refer to this capability also as link aggregation or trunking). Refer to the following URL for *EtherChannel configuration considerations*:

[http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.commadmn/doc/commadmndita/etherchannel\\_consider.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.commadmn/doc/commadmndita/etherchannel_consider.htm)

2. Note that the EtherChannel network interfaces (`en2`) are connected to each other through the external network cloud forming the public network.
3. Assign static IP addresses to the EtherChannel network adapters (`en2`) on the primary and standby nodes (and on the Arbitrator node for the three-node topology):

4. Make sure “short name” (i.e., `hostname -s`) is associated with “*Internet Network Extension*” (`inet0`). If required, you can set it by issuing the following command:

```
root> chdev -l inet0 -a hostname=`uname -a | awk '{print $2}'`
```

**Note:** On Solaris, `hostname -s` will set the host name, hence `uname -a` should be used instead.

5. Make sure that the primary and standby node names (and the Arbitrator node name for the three-node topology) are mapped to their corresponding public IP addresses in the `/etc/hosts` file:

- **For three-node cluster topology**  
9.47.73.22 harsys1 harsys1.beaverton.ibm.com # HADR Primary  
9.68.28.248 alaska alaska.yamato.ibm.com # HADR Standby  
9.26.48.48 hasys3 hasys3.torolab.ibm.com # Arbitrator
- **For two-node cluster with shared DISK tiebreaker topology**  
9.47.73.22 harsys1 harsys1.beaverton.ibm.com # HADR Primary  
9.68.28.248 alaska alaska.yamato.ibm.com # HADR Standby

6. Make sure that the DB2 nodes configuration file (`~/sql1lib/db2nodes.cfg`) for the instance residing on the server `harsys1` acting as primary is set up as follows:

```
0 harsys1 0
```

Also ensure that the DB2 nodes configuration file for the instance residing on the server `alaska` acting as standby has contents as follows:

```
0 alaska 0
```

7. Ensure that the short-name values (such as `alaska` or `harsys1`) are used to refer to these cluster nodes both in the XML input file and when prompted by the `db2haicu` setup tool.
8. Ensure that the `hostname` command when executed on any one of the cluster nodes will return the short-name value. For example, on the `alaska.yamato.ibm.com` machine `hostname` command must return `alaska`.
9. The primary and standby machines must be able to ping each other over both networks. Issue the following commands on both the primary and standby machines and make sure that they complete successfully:

```
root/dba> ping harsys1
root/dba> ping alaska
root/dba> ping harsys1.beaverton.ibm.com
root/dba> ping alaska.yamato.ibm.com
root/dba> ping 9.47.73.22
root/dba> ping 9.68.28.248
```

## A note on host names

---

You need to ensure that “short-name” is resolved before the fully qualified domain name (FQDN), which is the reason for the particular format of the `/etc/host` file in Step 5 above.

- **Before Version 9.5 Fix Pack 5:** If DB2 9.5 ESE was installed in a **HADR** configuration, only short-names are supported.
- **Version 9.5 Fix Pack 5:** If DB2 9.5 ESE was installed in a **HADR** configuration, both short-name and FQDN are supported. (Again, it must be in the same format throughout)

The short-name must be pingable from all nodes in the cluster and `hostname` command must return the short name on each node. In addition, when the initial cluster security is established by running the `preprnode` command, you must specify the short-name.

## Port requirements to run a multi-site cluster

Typically, most organizations implement some form of a Firewall Service at the entry point to each corporate site/subnet. While restrictions enforced by Firewall rules can be either mandated at corporate level or at individual site level, you need to ensure all ports required by DB2 HADR and cluster manager (RSCT in our topologies) are allowed for both inbound and outbound traffic between sites. A common issue seen when implementing and/or maintaining multi-site clusters is communication problems on one or more required ports due to incorrect Firewall rules.

We used the following port-IP-application mapping to configure Firewall rules in our three sites:

hostname [source IP address]	Application name (/etc/services)	Port number	Remark (usage)
alaska.yamato.ibm.com [9.68.28.248]	cthats	12347/udp	RSCT cluster port for CTHATS daemon
	cthags	12348/udp	RSCT cluster port for CTHAGS daemon
	rnc	657/tcp, 657/udp	RSCT cluster port for RMC daemon
	DB2_db2inst1	60000/tcp	DB2 listener port (SVCENAME)
	hadr_local_svc	50001/tcp	HADR heartbeat ports
	hadr_remote_svc	50002/tcp	HADR heartbeat ports
harsys1.beaverton.ibm.com [9.47.73.22]	cthats	12347/udp	RSCT cluster port for CTHATS daemon
	cthags	12348/udp	RSCT cluster port for CTHAGS daemon
	rnc	657/tcp, 657/udp	RSCT cluster port for RMC daemon
	DB2_db2inst1	60000/tcp	DB2 listener port (SVCENAME)
	hadr_local_svc	50002/tcp	HADR heartbeat ports
	hadr_remote_svc	50001/tcp	HADR heartbeat ports
hasys3.torolab.ibm.com [9.26.48.48]	cthats	12347/udp	RSCT cluster port for CTHATS daemon
	cthags	12348/udp	RSCT cluster port for CTHAGS daemon
	rnc	657/tcp, 657/udp	RSCT cluster port for RMC daemon
svt3070a.torolab.ibm.com [9.26.51.34]	iscsi-target	3260/tcp, 3260/udp	Port for iSCSI NetApp filer
hasys2.torolab.ibm.com [9.26.48.47]	DB2_db2inst1	60000/tcp	DB2 listener port (SVCENAME)

**Note:** The Firewall rules for the Arbitrator node (`hasys3.torolab.ibm.com`) were only used in the three-node topology while the NetApp filer ports were only required for the two-node shared DISK topology. The node `hasys2.torolab.ibm.com` is our DB2 client located in Toronto site. Since these ports requirements for RSCT may change over time, refer the *Reliable Scalable Cluster Technology (RSCT) Administration Guide: Appendix A. RSCT network considerations* for updated information (or go to the URL <http://publib.boulder.ibm.com/infocenter/clresctr> and click the links for port usage).

DB2 system topology and configuration for automated multi-site HA and DR

## Configuring DB2 HADR

HADR can be initialized through the command line processor (CLP), or by the Set Up High Availability Disaster Recovery (HADR) wizard in the Control Center. We illustrate the use of CLP to initialize HADR for the first time.

### Prerequisites

- Determine the host name, IP address, and the service name or port number for each of the HADR databases.
- If a host has multiple network interfaces, ensure that the HADR host name or IP address maps to the intended one.
- You need to allocate separate HADR ports in `/etc/services` for each protected database. These cannot be the same as the ports allocated to the instance.
- Check that the host name is mapped to only one IP address.

**Note:** Using the same instance names for the primary and standby databases is strongly recommended.

### HADR configuration and initialization steps

1. Create the HADR standby database. Typically, this is done either by restoring a backup (online/offline) of the primary database or by initializing a split-mirror copy.
2. Set the HADR configuration parameters on the primary and standby databases.

**Note:** It is very important that you set the following configuration parameters after the standby database has been created: `HADR_LOCAL_HOST`, `HADR_LOCAL_SVC`, `HADR_REMOTE_HOST`, `HADR_REMOTE_SVC`, `HADR_REMOTE_INST`. If they are set prior to creating the standby database, the settings on the standby database will reflect what is set on the primary database.

**We used the following HADR configuration in both of our topologies:**

DBM cfg parameter	Primary node	Standby node
HADR_LOCAL_HOST	harsys1.beaverton.ibm.com	alaska.yamato.ibm.com
HADR_LOCAL_SVC	50001	50002
HADR_REMOTE_HOST	alaska.yamato.ibm.com	harsys1.beaverton.ibm.com
HADR_REMOTE_SVC	50002	50001
HADR_REMOTE_INST	db2inst1	db2inst1
HADR_SYNCMODE	NEARSYNC	NEARSYNC
HADR_TIMEOUT	180	180
HADR_PEER_WINDOW	180	180

3. Start HADR on the standby database:

```
dba> db2 START HADR ON DB hadb AS STANDBY
```

4. Start HADR on the primary database (which will start HADR on both nodes):

```
dba> db2 START HADR ON DB hadb AS PRIMARY
```

5. Monitor the HADR status:

```
dba> db2pd -hadr -db hadb
```

**Note:** Ensure that HADR pair is able to establish `Peer` mode status.

You can also use `db2 get snapshot for database` command to get the HADR status.

## Importance of clock synchronization between HADR servers

---

In the integrated HA solution, any unplanned failover of primary HADR server is handled by executing `db2 takeover by force PEER WINDOW` only on the standby node. Hence, it is extremely critical that clocks on primary and standby servers are synchronized within 5 seconds of each other to avoid any potential `HADR_PEER_WINDOW` expired conditions.

1. HADR peer window end time is based on the clock of the Primary database. Hence, to check for potential clock skew between two HADR servers located in two different time zones, you need to covert clocks on both sites to Greenwich Mean Time (GMT):

- Issue the command `date` on both servers

- **Now convert the time to GMT and compare.** On AIX and Linux, this GMT conversion can be done by issuing the command `TZ=GMT date`.

- For example, in our cluster,

- Primary node in PDT time zone:  
10:28:47 Friday August 7, 2009 (PDT)  
17:28:47 Friday August 7, 2009 (GMT)
- Standby server in CDT time zone:  
12:32:01 Friday August 7, 2009 (CDT)  
17:32:01 Friday August 7, 2009 (GMT)

We see there is ~3 minute clock skew between the two sites in the above example.

2. Update the time. You can use `smitty date` on AIX or `hwclock` on Linux to update the date/time.
3. The best way to ensure that the clocks are synchronized is to configure both nodes to use NTP (Network Time Protocol). Note that each server must synchronize to an NTP server in their respective time zones.
4. Periodically ensure that the system clocks are synchronized.
5. In the event of a takeover failure due to clock-skew between two servers, you may see something similar to the following DB2 diagnostic log (usually in `~/sqlllib/db2dump/db2diag.log`) snippet:

```
2009-08-07-10.33.45.885625-300 I90023A599          LEVEL: Error
PID       : 127462                TID    : 7969          PROC   : db2sysc 0
INSTANCE: db2inst1              NODE   : 000          DB     : HADB
APPHDL   : 0-4566                APPID  : *LOCAL.db2inst1.090807153344
AUTHID   : DB2INST1
EDUID    : 7969                  EDUNAME: db2agent (HADB) 0
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrTakeoverHdrRouteIn, probe:55610
RETCODE : ZRC=0x8280001D=-2105540579=HDR_ZRC_NOT_TAKEOVER_CANDIDATE_FORCED
"Forced takeover rejected as standby is in the wrong state or peer window has expired"

2009-08-07-10.33.45.959409-300 I90623A897          LEVEL: Error
PID       : 417904                TID    : 1            PROC   : db2gcf
INSTANCE: db2inst1              NODE   : 000
EDUID    : 1
FUNCTION: DB2 Common, Generic Control Facility, gcf_start, probe:280
DATA #1 : String, 59 bytes
TAKEOVER BY FORCE PEER WINDOW ONLY failed for database HADB
DATA #2 : signed integer, 4 bytes
```

```
0
DATA #3 : signed integer, 4 bytes
-1770
CALLSTCK:
 [0] 0x09000000018E95B0 oss_log__FP9OSSLogFacUiN32U1N26iPPc + 0x1B0
 [1] 0x09000000018E937C ossLog + 0x7C
 [2] 0x09000000045BAC40 gcf_start + 0x24C0
 [3] 0x0900000004544C80 start__9GcfCallerFP12GCF_PartInfoU1P11GCF_RetInfo + 0x1A0
 [4] 0x0000000100000EC8 main + 0x548
 [5] 0x0000000100000290 __start + 0x98
 [6] 0x0000000000000000 ?unknown + 0x0
 [7] 0x0000000000000000 ?unknown + 0x0
 [8] 0x0000000000000000 ?unknown + 0x0
 [9] 0x0000000000000000 ?unknown + 0x0
```

In addition to DB2 `HADR_PEER_WINDOW` considerations, the time-of-day clocks on all nodes within the peer domain must be synchronized to within a reasonable tolerance of each other – typically, only a few seconds. Too great a variance among the clock settings on the nodes in the peer domain can cause RSCT control messages to time out and could even cause a node reboot.

## Considerations for configuring the `HADR_TIMEOUT` and `HADR_PEER_WINDOW`

---

When designing a multi-site HADR configuration spanning long distances, give special consideration to the following two HADR database configuration parameters:

- **`HADR_TIMEOUT`:** If HADR heartbeat communication between the two nodes is lost for longer than the `HADR_TIMEOUT` time period, then the HADR database concludes that the connection with the partner database is lost. If the database is in peer state when the connection is lost, then it moves into *disconnected peer state* when the `HADR_PEER_WINDOW` database configuration parameter is set to a value that is greater than zero.
- **`HADR_PEER_WINDOW`:** The `HADR_TIMEOUT` parameter determines whether the database goes into *disconnected peer state* after the connection is lost, and how long the database should remain in that state. HADR will break the connection as soon as a network error is detected during send, receive, or poll on the TCP socket (which is polled every 100 ms). This allows HADR to respond quickly to network errors detected by the operating system. In the worst-case scenario, a database application that is running at the time of failure can be blocked for a period of time equal to the sum of `HADR_TIMEOUT` and `HADR_PEER_WINDOW`.

### Calculating suitable `HADR_TIMEOUT` and `HADR_PEER_WINDOW` values

Setting `HADR_TIMEOUT` and `HADR_PEER_WINDOW` to a small value would reduce the time that a database application must wait in the event of a communication loss between the HADR pair. But the `HADR_TIMEOUT` should be set to a value that is long enough to avoid false alarms on the HADR connection caused by short, temporary network interruptions. Hence, we chose 180 seconds as the value of `HADR_TIMEOUT` database configuration parameter.

The `HADR_PEER_WINDOW` should be set to a value that is long enough to allow the system to perform automated failure responses. If, for example, TSA/RSCT cluster manager detects primary database failure before *disconnected peer state* ends, a failover to the standby database takes place. Data is not lost in the failover because all data from the old primary is replicated to the new primary. If the `HADR_PEER_WINDOW` is too short, the HA system may not have enough

time to detect the failure and respond. Therefore, for long-distance automated DB2 HA solutions such as the two topologies described in this white paper, the following process can be used to properly calculate the `HADR_PEER_WINDOW` database configuration parameter.

1. Three factors are generally involved in deciding `HADR_PEER_WINDOW` value in a TSA/RSCT automated DB2 HA cluster solution:

- **Time to detect node failure** (in sec) = **Sensitivity x (Period x 2)** of the Communication Group (`CommGroup`) the nodes in the cluster belong to.

**Notes:**

- A NIC/IP can belong to *only one* Communications Group at a time per single node – i.e., if there are multiple NICs on one or more nodes in a cluster, they will be placed in different Communication Groups.

- **Sensitivity:** The number of missed heartbeats that constitute a failure

- **Period:** The number of seconds between heartbeats

If the cluster is already operational, you can query the Cluster `CommGroups` to find out these two values. For example, these are the specific values in our three-node multi-site TSA/RSCT cluster topology.

```
root> lscomg
Name Sensitivity Period Priority Broadcast SourceRouting NIMPathName NIMParameters Grace
CG1 10 3 1 Yes Yes 30
```

Hence in our cluster, time to detect node failure =  $10 \times (3 \times 2) = 60$  sec

- **Time for TSA/RSCT to perform all the resource group moving**  $\approx 60$  sec

- **Ping Grace Period.** Whenever a node loses connection with rest of the cluster nodes, the RSCT Topology Services subsystem will issue an ICMP echo to check whether the system is still reachable. If that node responds within the time period set by Ping Grace Period, the cluster will not detect this as a node failure. Note that Ping Grace Period is not really meant for network glitches, but for cases where daemons get blocked because of memory starvation or other factors. We set this value to 30 seconds in both of our cluster topologies.

2. Calculating the `HADR_PEER_WINDOW` value:

- **`HADR_PEER_WINDOW`  $\geq$  Time to detect node failure + Time for TSA/RSCT to perform all the resource group moving + Ping Grace Period**

- For our cluster, `HADR_PEER_WINDOW`  $\geq 60 + 60 + 30$  Sec  $\geq 150$  sec

- Hence, we settled for 180 seconds allowing an additional 30-second grace period

Reference: DB2 Information Center topic “*Considerations for configuring the `HADR_TIMEOUT` and `HADR_PEER_WINDOW` database configuration parameters*”  
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.ha.doc/doc/c0056394.html>

## A note on HADR performance

---

Configuring different aspects of your topology, including network bandwidth, CPU power, memory usage, response time for log I/O and buffer size, can improve the performance of your DB2 HADR configuration.

For optimum HADR performance, consider the following recommendations:

- Network bandwidth must be greater than the database log generation rate. Incorrect provisioning of bandwidth for HADR solutions can adversely affect production performance and can invalidate the overall solution. Dedicated network trunk for HADR traffic between sites is therefore highly recommended.
- Apart from bandwidth, it is important to have good network latency to send the log data from primary to standby. You can use the `simhadr` tool to test the bandwidth and latency in order to understand bandwidth and latency. This tool can be downloaded from [http://www.ibm.com/developerworks/wikis/display/data/HADR\\_sim](http://www.ibm.com/developerworks/wikis/display/data/HADR_sim)
- Network delays affect the primary database only in `SYNC` and `NEARSYNC` modes.
- The slowdown in system performance as a result of using `SYNC` mode can be significantly larger than that of the other synchronization modes – i.e., the log write on the standby database plus round-trip messaging. Hence we used `NEARSYNC` mode in both of our topologies and generally recommend this setting for any automated long-distance multi-site HADR configurations.
- In `NEARSYNC` mode, the primary database writes and sends log pages in parallel and then waits for an acknowledgement from the standby. The standby database acknowledges as soon as the log pages are received into its memory. Since the acknowledgement might have already arrived by the time the primary database finishes local log write, on a fast network, the overhead to the primary database is minimal.
- For each log write on the primary, the same log pages are also sent to the standby. The size of a write operation (called a flush) is limited to the log buffer size database configuration parameter (`logbufsz`) on the primary database. The exact size of each flush is non-deterministic; hence a larger log buffer does not necessarily lead to a larger flush.
- Fast log disk device on both primary and standby database is also critical to performance of HADR-enabled databases. Typically, the response time for log IO to disk should be in the low millisecond range.
- The standby database should be powerful enough to replay the logged operations of the database as fast as they are generated on the primary. Hence, identical primary and standby hardware is highly recommended.
- If there is high memory usage, (i.e., significant paging activity), not only will HADR performance suffer, but the possibility of critical RSCT and/or system daemons blocking increases. Note that under certain conditions, significant delays in RSCT daemons may result in the node getting rebooted.

## Network congestion from HADR traffic

If the standby database is too slow replaying log pages, its log-receiving buffer might fill up, thereby preventing the buffer from receiving more log pages. For `SYNC` and `NEARSYNC` modes, the network pipeline consisting of the primary machine, the network, and the standby database can usually absorb a single flush, and congestion will not occur. However, the primary database remains blocked waiting for an acknowledgement from the standby database on the flush operation.

Increasing the size of the standby database log-receiving buffer can help to reduce congestion, although it might not remove all of the causes of congestion. By default, the size of the standby database log-receiving buffer is two times the size of the primary database log-writing buffer. The DB2 registry variable `DB2_HADR_BUF_SIZE` can be used to tune the size of the standby

database log-receiving buffer. If this registry variable needs to be tuned, it should be configured on both primary and standby databases at the same time so that the tuning is consistent on both database systems.

Congestion is reported by the `hadr_connect_status` monitor element.

## Tuning TCP socket buffer size for HADR

To maximize network and HADR performance, the TCP socket buffer sizes may require tuning. HADR log shipping workload, network bandwidth, and transmission delay are important factors to consider when tuning the TCP socket buffer sizes. Prior to DB2 9.5 Fix Pack 2, changing the TCP socket buffer size for HADR connections could only be done at the operating system level and the settings would be applicable across all TCP connections on the machine. Setting a large system level socket buffer size may consume a large amount of memory.

Two DB2 registry variables, `DB2_HADR_SOSNDBUF` and `DB2_HADR_SORCVBUF`, were introduced in Fix Pack 2 to allow tuning of the TCP socket send and receive buffer size for HADR connections only. These two variables have the value range of 1024 to 4294967295 bytes and default to the socket buffer size of the operating system, which will vary depending on the operating system.

For more details, refer to the DB2 9.5 Information Center topic “*High availability disaster recovery (HADR) performance*”:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.ha.doc/doc/c0021056.html>

## Using Automatic Client Reroute (ACR) with DB2 HADR

---

Until this point in the discussion, the focus has been on server-side failover. Issues with clients are critical as well. There are various techniques to fail over applications. In both of our topologies, the DB2 Automatic Client Reroute (ACR) feature will be used to achieve this. Note that *Virtual IP (VIP) failover is not used in either configuration in this white paper*.

### Configuring ACR

1. Identify the numeric value of the DB2 listener port number on each HADR node.

```
dba> awk "/^db2 get dbm cfg | awk '/SVCENAME/ {print $6}'"/  
/etc/services | head -1  
DB2_db2inst1      60000/tcp
```
2. Since we are using ACR, on each DB2 HADR server node set the "*alternate Server*" parameter. Use the port number values from the preceding step to set this:
  - **On primary** (`harsys1.beaverton.ibm.com`):

```
dba> db2 -v update alternate server for db hadb using hostname  
alaska.yamato.ibm.com port 60000
```
  - **On standby** (`alaska.yamato.ibm.com`):

```
dba> db2 -v update alternate server for db hadb using hostname  
harsys1.beaverton.ibm.com port 60000
```

3. On the DB2 clients, we need to catalog TCP/IP node and HADR database for only the primary database. DB2 Automatic Client Reroute (ACR) feature will redirect connections to the active server in case of a failover.

```
dba> db2 catalog tcpip node HAPNODE remote harsys1.beaverton.ibm.com
server 60000
dba> db2 catalog db hadb at node HAPNODE
```

Verify that you can connect to the primary database. Upon the first connection, client database catalogs are populated with alternate server information.

```
dba> db2 list db directory

System Database Directory

Number of entries in the directory = 1

Database 1 entry:

Database alias           = HADB
Database name           = HADB
Node name                = HAPNODE
Database release level  = c.00
Comment                  =
Directory entry type    = Remote
Catalog database partition number = -1
Alternate server hostname = alaska.yamato.ibm.com
Alternate server port number = 60000
```

4. Run your DB2 client applications/workload against the HADR pair and perform few “role-switch” operations to validate that ACR is working.
  - Also take note of the total time to complete the HADR takeover operation:

```
dba> date; db2 takeover hadr on db hadb; date
Mon Aug 24 14:27:39 PDT 2009
DB20000I The TAKEOVER HADR ON DATABASE command completed successfully.
Mon Aug 24 14:28:14 PDT 2009
```
  - From the above result, in our three-node topology the delay is 35 sec.

## Setting up logging – SYSLOGS

---

We need to enable system logging (SYSLOG) to monitor the status of cluster operations. Logging will help you to troubleshoot any potential issues that might arise over the operational life cycle of a cluster.

### Configuring AIX SYSLOGS

1. Edit the SYSLOG configuration file `/etc/syslog.conf` and add the following lines:

```
*.debug /tmp/syslog.out rotate size 10m time 1w files 10 #10 files, 10MB each, weekly rotate
kern.debug /var/log/kern rotate files 12 time 1m compress #12 files, monthly rotate, compress
```
2. If the target log file does not exist, you need to create it first:

```
root> touch /tmp/syslog.out
root> touch /var/log/kern
```
3. Refresh SYSLOG daemon:

```
root> refresh -s syslogd
```

## Creating the initial two-node DB2 HADR cluster using db2haicu

We start the configuration of both cluster topologies with the same starting point – i.e., two-node DB2 HADR cluster. The `db2haicu` tool has two modes of operation: 1) *interactive*, 2) *XML input file*. We will use XML input file to create this cluster and will show how to add the third node using the `db2haicu` interactive mode to configure the three-node topology.

### Prerequisites:

- Install the same RSCT filesets on both nodes. In our configuration we used RSCT version 2.4.10.0.
- Complete all the tasks described under **Setting up the basic network**.
- Install DB2 9.5 Fix Pack 3 (which is bundled with TSA 2.2.0.7) on the two HADR nodes.
- If DB2 9.5 is already installed without TSA, you need to use `installSAM` command and manually install Tivoli SA 2.2.0.7 software (TSA). TSA install source is located in “<imageTop>/db2/<platform>/tsamp” of your DB2 install media. See the URL: <https://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.ha.doc/doc/t0051364.html>  
Note that while DB2 Fix Pack will have the latest TSA code base, you will still need to get the license file from the DB2 base install image and install using:  
`samlcm -i <imageTop>/db2/<platform>/tsamp/license/sam22.lic.`  
In addition, after completing the TSA install you need to copy the latest SA policy scripts used by DB2 integrated HA feature using `db2cptsa` tool.  
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.cmd.doc/doc/r0052706.html>
- Verify that HADR pair configuration is completed and the two nodes are in *peer state*.
- Add the following line to `~/.profile` of root to enable infrastructure mode:  
`export CT_MANAGEMENT_SCOPE=2`

## Creating the DB2 HADR cluster with db2haicu using XML input file

1. For using TSA in a multi-node cluster configuration, you need to run `preprnode` (prepare the security settings between the machines) on each-node first to ensure that TSA will have required permissions to access any TSA managed resources:

```
root> preprnode harsys1 alaska
```

**Note:** The command `preprnode <nodeA>` should be run on `nodeB` to enable `nodeA` to access `nodeB` when creating a cluster. Say, when creating a cluster on nodes `nodeA`, `nodeB`, and `nodeC` then the minimum requirement for `preprnode` is to run it as follows:

```
node B : preprnode nodeA
node C : preprnode nodeA
```

Then proceed to issue `mkrpdomain` from `nodeA`. It is valid to run an "all-to-all" setup:

```
node A : preprnode nodeB nodeC
node B : preprnode nodeA nodeC
node C : preprnode nodeA nodeB
```

2. Modify the sample `db2haicu` HADR configuration XML input file to suit your settings.

- **Note:** Skip defining `db2haicu` networks, and Quorum devices at this stage.
- The XML input file (`db2ha_HADR.xml`) used in our setup is included in [Appendix D – db2haicu XML input file](#) for your reference.

- Run db2haicu and create the cluster objects first on the standby node (alaska) and then on the primary node (harsys1):

```
dba> db2haicu -f db2ha_HADR.xml
```

- Verify that the cluster, resource groups, etc. were created successfully:

```
root/dba> lssam
```

```
$ lssam
Online IBM.ResourceGroup:db2_db2inst1_alaska_0-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_alaska_0-rs
'- Online IBM.Application:db2_db2inst1_alaska_0-rs:alaska
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADB-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs
'- Offline IBM.Application:db2_db2inst1_db2inst1_HADB-rs:alaska
'- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs:harsys1
Online IBM.ResourceGroup:db2_db2inst1_harsys1_0-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_harsys1_0-rs
'- Online IBM.Application:db2_db2inst1_harsys1_0-rs:harsys1
$
```

## A note on RSCT configuration to address resource contention

In heavily loaded systems, contention for resources such as memory, I/O, or CPU may result in RSCT daemons not being able to make progress in a timely manner. This contention may result in false node failures, or in RSCT daemons being recycled.

To minimize the possibility that the daemons are prevented from accessing system resources, the Topology Services, Group Services, and Configuration Resource Manager daemons now run with a fixed real-time CPU priority. Fixed real-time CPU priority should allow them to access CPU resources even when several other processes in the system are running.

**Note:** The use of a real-time fixed CPU priority will not result in the RSCT daemons using additional CPU resources. The priority only ensures that the daemons can access the CPU whenever needed. The real-time fixed CPU priority and memory pinning is applicable to AIX and Linux environments only.

The second step in improving the daemons' resilience to resource contention involves locking (or *pinning*) their pages in real memory. Once the pages are brought into physical memory, they are not allowed to be paged out. This minimizes the possibility of daemons becoming blocked or delayed during periods of high paging activity. Because the daemons' pages are locked in memory, the corresponding physical pages are dedicated to the daemons and cannot be used by other processes in the system. Therefore, the amount of physical memory available for other processes is slightly reduced.

By default, the daemons will use a fixed CPU priority and lock the pages in memory. This behavior can be changed, however, with the use of the `cthatstune` command.

The following command will direct the RSCT daemons not to use a fixed CPU priority:

```
root> /usr/sbin/rsct/bin/cthatstune -p 0
```

For the Group Services daemon, the setting will only take effect the next time RSCT Peer Domain is online on the node.

The following command will direct the RSCT daemons not to lock their pages in memory:

```
root> /usr/sbin/rsct/bin/cthatstune -m NONE
```

The setting will only take effect the next time RSCT Peer Domain is online on the node.

[http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.rsct\\_6100\\_02.admin.doc/bl503\\_contention.html](http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.rsct_6100_02.admin.doc/bl503_contention.html)

Next we move on to the first configuration in our white paper, the three-node DB2 HA cluster.

DB2 system topology and configuration for automated multi-site HA and DR

## System configuration: three-node DB2 HA cluster topology

---

Now we proceed to the setup and configuration tasks more specific to the three-node cluster topology. Complete the steps described in “[Initial configuration – common to both topologies](#)” before moving on to the scenario-specific configuration described below.

### Adding the arbitrator node to the cluster using db2haicu

---

We used the following process to add the Arbitrator node (hasys3.torolab.ibm.com) to the two-node HADR cluster we created previously.

1. Install Tivoli SA 2.2.0.7 software (TSA) on the Arbitrator node using `installSAM` command. TSA install source is located in “<imageTop>/db2/<platform>/tsamp” of your DB2 install media. Note, that there is no need to neither install DB2 on this third node nor copy the SA policy scripts used by DB2 integrated HA feature.

<https://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.ha.doc/doc/t0051364.html>

2. You need to run `preprnode` on each node again to ensure that TSA will have required permissions to access any TSA managed resources on the new node:

- On each HADR node : root> `preprnode hasys3`
- On the arbitrator (new) node : root> `preprnode hasys1 alaska hasys3`

3. Launch `db2haicu` in the *maintenance mode* from any one of the HADR nodes:

```
dba> db2haicu
```

```
Select an administrative task by number from the list below:
```

1. Add or remove cluster nodes.
2. Add or remove a network interface.
3. Add or remove HADR databases.
4. Add or remove an IP address.
5. Move DB2 database partitions and HADR databases for scheduled maintenance.
6. Create a new quorum device for the domain.
7. Destroy the domain.
8. Exit.

4. Select the option to **Add/Remove a Cluster node** from the menu options, review existing cluster nodes, and specify the host name of the third node. **Note:** Ensure short-name is specified for cluster node name:

```
Select an administrative task by number from the list below:
```

1. Add or remove cluster nodes.
2. Add or remove a network interface.
3. Add or remove HADR databases.
4. Add or remove an IP address.
5. Move DB2 database partitions and HADR databases for scheduled maintenance.
6. Create a new quorum device for the domain.
7. Destroy the domain.
8. Exit.

```
Enter your selection:
```

```
1
```

```
Do you want to review the status of each cluster node in the domain before you begin? [1]
```

```
1. Yes
```

```
2. No
```

```
1
```

```
Domain Name: db2HADomain
Node Name: alaska --- State: Online
Node Name: harsys1 --- State: Online
Do you want to add or remove cluster nodes to or from the domain? [1]
1. Add
2. Remove
1
Enter the host name of a machine to add to the domain:
hasys3
Adding node hasys3 to the cluster ...
Adding node hasys3 to the cluster was successful.
Do you want to add another node to the domain? [1]
1. Yes
2. No
2
Do you want to make any other changes to the cluster configuration? [1]
1. Yes
2. No
2
All cluster configurations have been completed successfully. db2haicu exiting ...
```

5. Verify that the new node is part of the cluster and that it is online:

```
dba> lsrpnode
Name      OpState RSCTVersion
hasys3    Online  2.4.10.0
harsys1   Online  2.4.10.0
alaska    Online  2.4.10.0
```

**Note:** If the add node operation fails, ensure that the clustering software is installed on the third node. The clustering software can be installed directly from the DB2 installation media using the `installSAM` command.

## Configure network equivalencies or “db2haicu networks”

---

The main use of creating db2haicu networks/TSA network equivalencies is to enable NIC monitoring. Since we had configured EtherChannel binding all Ethernet adapters available on each server, we do not need to create these network equivalencies. To reiterate, *there is no requirement to create these networks for cases where there is exactly one active network adapter per node* (as is the case in this example). However, there is no harm in creating these network equivalencies either, and we will show how they serve a more useful purpose in other variations of our three-node HA cluster topology, later in this paper.

### Prerequisites:

- Create TSA/RSCT cluster including all three nodes using "db2haicu" utility.

1. List the RSCT Communication Group that the three cluster nodes belong to. Take note of the network interface on each node that is defined in that `CommGroup`.

```
root> lscomg
Name Sensitivity Period Priority Broadcast SourceRouting NIMPathName NIMParameters Grace
CG1 4 1 1 Yes Yes
(Default)

root> lscomg -i CG1
Name NodeName IPAddress Subnet SubnetMask
en2 hasys3 9.26.48.48 9.26.48.0 255.255.255.0
en2 harsys1 9.47.73.22 9.47.73.0 255.255.255.0
en2 alaska 9.68.28.248 9.68.28.0 255.255.255.0
```

## 2. Launch db2haicu in the maintenance mode :

```
dba> db2haicu
```

```
Select an administrative task by number from the list below:
```

1. Add or remove cluster nodes.
2. **Add or remove a network interface.**
3. Add or remove HADR databases.
4. Add or remove an IP address.
5. Move DB2 database partitions and HADR databases for scheduled maintenance.
6. Create a new quorum device for the domain.
7. Destroy the domain.
8. Exit.

## 3. Select the option to **Add/Remove a Network Interface** from the menu options.

•Specify `en2` as the first network interface on the first node. **Note:** Ensure short-name is specified and proceed.

•Since we did not create any network equivalencies, select the option to **Create a new public network**.

```
Select an administrative task by number from the list below:
```

1. Add or remove cluster nodes.
2. **Add or remove a network interface.**
3. Add or remove HADR databases.
4. Add or remove an IP address.
5. Move DB2 database partitions and HADR databases for scheduled maintenance.
6. Create a new quorum device for the domain.
7. Destroy the domain.
8. Exit.

```
Enter your selection:
```

```
2
```

```
Do you want to view a list of the existing networks and the associated network interface cards and continue to configure the domain? [1]
```

```
1. Yes
```

```
2. No
```

```
1
```

```
No networks were found in the cluster.
```

```
Do you want to add or remove network interface cards to or from a network? [1]
```

```
1. Add
```

```
2. Remove
```

```
1
```

```
Enter the name of the network interface card:
```

```
en2
```

```
Enter the host name of the cluster node which hosts the network interface card en2:
```

```
harsys1
```

```
Enter the name of the network for the network interface card: en2 on cluster node: harsys1
```

```
1. Create a new public network for this network interface card.
```

```
2. Create a new private network for this network interface card.
```

```
Enter selection:
```

```
1
```

```
Are you sure you want to add the network interface card en2 on cluster node harsys1 to the network db2_public_network_0? [1]
```

```
1. Yes
```

```
2. No
```

```
1
```

```
Adding network interface card en2 on cluster node harsys1 to the network db2_public_network_0 ...
```

```
Adding network interface card en2 on cluster node harsys1 to the network db2_public_network_0 was successful.
```

```
Do you want to add another network interface card to a network? [1]
```

```
1. Yes
```

```
2. No
```

```
1
```

```
Enter the name of the network interface card:
```

```
en2
```

```
Enter the host name of the cluster node which hosts the network interface card en2:
```

```

alaska
Enter the name of the network for the network interface card: en2 on cluster node: alaska
1. db2_public_network_0
2. Create a new public network for this network interface card.
3. Create a new private network for this network interface card.
Enter selection:
2
Are you sure you want to add the network interface card en2 on cluster node alaska to the
network db2_public_network_1? [1]
1. Yes
2. No
1
Adding network interface card en2 on cluster node alaska to the network
db2_public_network_1 ...
Adding network interface card en2 on cluster node alaska to the network
db2_public_network_1 was successful.
Do you want to add another network interface card to a network? [1]
1. Yes
2. No
1
Enter the name of the network interface card:
en2
Enter the host name of the cluster node which hosts the network interface card en2:
hasys3
Enter the name of the network for the network interface card: en2 on cluster node: hasys3
1. db2_public_network_1
2. db2_public_network_0
3. Create a new public network for this network interface card.
4. Create a new private network for this network interface card.
Enter selection:
3
Are you sure you want to add the network interface card en2 on cluster node hasys3 to the
network db2_public_network_2? [1]
1. Yes
2. No
1
Adding network interface card en2 on cluster node hasys3 to the network
db2_public_network_2 ...
Adding network interface card en2 on cluster node hasys3 to the network
db2_public_network_2 was successful.
Do you want to add another network interface card to a network? [1]
1. Yes
2. No
2
Do you want to make any other changes to the cluster configuration? [1]
1. Yes
2. No
2
All cluster configurations have been completed successfully. db2haicu exiting ...

```

#### 4. List the network equivalency created in Step 3:

```
root> lsequ -Ab -l -s "Name like 'db2_public_network_%'"
```

```
Displaying Equivalency information:
```

```
All Attributes
```

```
Equivalency 1:
```

```

Name = db2_public_network_2
MemberClass = IBM.NetworkInterface
Resource:Node[Membership] = {en2:hasys3}
SelectString = ""
SelectFromPolicy = ANY
MinimumNecessary = 1
Subscription = {}
ActivePeerDomain = db2HADomain
Resource:Node[ValidSelectResources] = {en2:hasys3}
Resource:Node[InvalidResources] = {}
ConfigValidity =
AutomationDetails[CompoundState] = Automation

```

```
Equivalency 2:
```

```

Name = db2_public_network_1
MemberClass = IBM.NetworkInterface
Resource:Node[Membership] = {en2:alaska}
SelectString = ""
SelectFromPolicy = ANY
MinimumNecessary = 1
Subscription = {}
ActivePeerDomain = db2HADomain
Resource:Node[ValidSelectResources] = {en2:alaska}
Resource:Node[InvalidResources] = {}
ConfigValidity =
AutomationDetails[CompoundState] = Automation

```

Equivalency 3:

```

Name = db2_public_network_0
MemberClass = IBM.NetworkInterface
Resource:Node[Membership] = {en2:harsys1}
SelectString = ""
SelectFromPolicy = ANY
MinimumNecessary = 1
Subscription = {}
ActivePeerDomain = db2HADomain
Resource:Node[ValidSelectResources] = {en2:harsys1}
Resource:Node[InvalidResources] = {}
ConfigValidity =
AutomationDetails[CompoundState] = Automation

```

## Testing three-node DB2 HA cluster topology

After implementing this three-node DB2 HA cluster topology, we successfully conducted the following eight test cases to validate the operational status of the cluster.

Test case	Description
Graceful "role-switch" or planned outage	Switch primary and standby HADR roles. Useful when doing "rolling-upgrades".
Forced "role-switch"	Simulate primary DB2 instance crash and how to recover from it.
Forced "role-switch" using PEER WINDOW ONLY	Validate the HADR_PEER_WINDOW setting. This is the default behavior of the <code>hadrV95_start.ksh</code> script when moving resources to the standby.
Primary node failure	Simulate a brief loss of primary node, and how cluster resources are moved to standby.
Standby node failure	Simulate a brief loss of standby and subsequent reintegration into the cluster after coming back online.
Unexpected primary node failure - hardware/power/single-site failure	Simulate primary node failure for a long period (such as hardware issue) and subsequent reintegration into the cluster as standby after coming back online.
Network interface card (NIC) failures	How NIC failures are handled by switching to the backup adapter in EtherChannel.
Unexpected failure of arbitrator node or failure of third site	Simulate a brief loss of arbitrator node in the third site and its implications to the cluster.

Refer to the [Appendix F - Testing three-node DB2 HA cluster topology](#) for detailed test procedures for all of the above test cases.

## System configuration: DB2 HA cluster topology with shared DISK tiebreaker

---

Now we proceed to the setup and configuration tasks more specific to the two-node shared DISK tiebreaker cluster topology. Complete the steps described in [“Initial configuration – common to both topologies”](#) before moving on to the scenario-specific configuration described below.

### Adding the shared DISK tiebreaker to the cluster

---

We used the following process to add the shared DISK tiebreaker quorum device to the two-node HADR cluster we created previously.

#### Prerequisites:

- Create the initial TSA/RSCT cluster using "db2haicu" utility.
- Configure shared disk LUNs and serve them to two AIX hosts. Refer to [Appendix E – Configuring NetApp iSCSI shared LUN for the two AIX hosts](#) for details on how to configure iSCSI LUNs for AIX hosts using a NetApp filer.

1. On both nodes identify the `hdisk #` of the shared LUN that we will use as the DISK tiebreaker:

- On HADR primary (harsys1.beaverton.ibm.com): **hdisk3**

```
root> lsdev -Ccdisk
hdisk0 Available 01-08-00-3,0 16 Bit LVD SCSI Disk Drive
hdisk1 Available 01-08-00-4,0 16 Bit LVD SCSI Disk Drive
hdisk2 Available 01-08-00-5,0 16 Bit LVD SCSI Disk Drive
hdisk3 Available MPIO Other iSCSI Disk Drive
```

- On HADR standby (alaska.yamato.ibm.com): **hdisk4**

```
root> lsdev -Ccdisk
hdisk0 Available 57-08-00-8,0 16 Bit LVD SCSI Disk Drive
hdisk1 Available 57-08-00-9,0 16 Bit LVD SCSI Disk Drive
hdisk2 Available 57-08-00-10,0 16 Bit LVD SCSI Disk Drive
hdisk3 Available 57-08-00-11,0 16 Bit LVD SCSI Disk Drive
hdisk4 Available MPIO Other iSCSI Disk Drive
```

2. Verify that the iSCSI LUN is accessible by both hosts. For example, on HADR standby:

```
root> lspath -l hdisk3
Enabled hdisk3 iscsi0
```

3. Create a DISK tiebreaker device using the iSCSI LUN provisioned from NetApp filer:

```
root> mkrsrc IBM.TieBreaker Name="tb" Type=DISK NodeInfo='{"harsys1",
"DEVICE=/dev/hdisk3"}, {"alaska", "DEVICE=/dev/hdisk4"}'
HeartbeatPeriod=30
```

**Note:** If the `hdisk#` in AIX ODM happens to be the same, say `hdisk2`, for example, the above command can be simplified as follows:

```
root> mkrsrc IBM.TieBreaker Name="tb" Type=DISK
DeviceInfo="DEVICE=/dev/hdisk2" HeartbeatPeriod=30
```

In Linux environments, you would create a SCSI type tiebreaker device instead of a DISK type tiebreaker used in AIX environments.

In addition, the command used to create the tiebreaker device is slightly different. After obtaining the SCSI identifiers for the shared disk device (using `sginfo -l`), you can configure the tiebreaker device using a command similar to the following:

```
root> mkrsrc IBM.TieBreaker Name="tb" Type=SCSI DeviceInfo="ID=4 LUN=0" NodeInfo='{"node1",
"HOST=0,CHAN=0"}, [{"node2", "HOST=1 CHAN=2"}]' HeartbeatPeriod=30
```

[http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.rsct\\_6100\\_02.admin.doc/bl503\\_cscsi.html](http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.rsct_6100_02.admin.doc/bl503_cscsi.html)

4. Test SCSI-2 reservation (i.e., issue lock and unlock requests) from both nodes on the shared LUN. When a device is configured as a quorum disk tiebreaker, ConfigRM will be controlling the SCSI reservation locks (i.e., at any given time ConfigRM may release a lock). Hence, you need to test SCSI reservation before configuring this shared LUN as the quorum device. Two RSCT tools `disk_reserve` and `tb_break` can be used for this task.

•**Note:** You need to issue lock commands at the same time on both nodes to observe how reservation conflicts are handled.

•**Reserve (lock) issued on each node simultaneously:**

```
root@harsys1> /usr/sbin/rsct/bin/tb_break -v -l -t DISK "DEVICE=/dev/hdisk3"
Initializing DISK tie-breaker (DEVICE=/dev/hdisk3 -v)
Reserving tie-breaker (DEVICE=/dev/hdisk3 -v)
tb_reserve status 0 (errno=0)
root@alaska> /usr/sbin/rsct/bin/tb_break -v -l -t DISK "DEVICE=/dev/hdisk4"
Initializing DISK tie-breaker (DEVICE=/dev/hdisk4)
Reserving tie-breaker (DEVICE=/dev/hdisk4)
tb_reserve status 1 (errno=16)

root@alaska> /usr/sbin/rsct/bin/disk_reserve -v -l -d /dev/hdisk4
do_scsi_reserve(hdisk=/dev/hdisk4), forced_reserve=0 Entered
do_scsi_reserve(hdisk=/dev/hdisk4), Leaving, ecode=0
/usr/sbin/rsct/bin/disk_reserve cmd='l' (/dev/hdisk4) is successful
root@harsys1> /usr/sbin/rsct/bin/disk_reserve -v -l -d /dev/hdisk3
do_scsi_reserve(hdisk=/dev/hdisk3), forced_reserve=0 Entered
do_scsi_reserve(hdisk=/dev/hdisk3), Leaving, ecode=16
Operation is FAILED on disk (/dev/hdisk3), rc=16
```

•**Release (unlock) issued on each node:** Both of the above tools can be used for this.

```
root> /usr/sbin/rsct/bin/disk_reserve -v -u -d /dev/hdisk4
do_scsi_release(disk=/dev/hdisk4), Entered
do_scsi_release(hdisk=/dev/hdisk4), Leaving, errno=0
/usr/sbin/rsct/bin/disk_reserve cmd='u' (/dev/hdisk4) is successful
```

The tiebreaker will work correctly if the tiebreaker disk can be reserved and unlocked from either node and if the disk cannot be reserved while it is locked by the other node.

5. Configure the above disk tiebreaker device as quorum device in the cluster "db2HADomain":

```
root> chrsrc -c IBM.PeerNode OpQuorumTieBreaker="tb"
```

6. Verify that the tiebreaker device and Quorum device are correctly defined in the cluster:

```
root> lsrsrc -s "Name='tb'" IBM.TieBreaker
Resource Persistent Attributes for IBM.TieBreaker
resource 1:
  Name           = "tb"
  Type           = "DISK"
  DeviceInfo     = ""
  ReprobeData    = ""
  ReleaseRetryPeriod = 0
  HeartbeatPeriod = 30
  PreReserveWaitTime = 0
  PostReserveWaitTime = 0
  NodeInfo       = [{"harsys1", "DEVICE=/dev/hdisk3"}, {"alaska", "DEVICE=/dev/hdisk4"}]
  ActivePeerDomain = "db2HADomain"
```

7. List DB2 HA cluster status. You should now see all the HADR resource groups and the shared DISK tiebreaker quorum device information.

```

dba> db2pd -ha
      DB2 HA Status
Instance Information:
Instance Name           = db2inst1
Number Of Domains      = 1
Number Of RGs for instance = 2

Domain Information:
Domain Name            = db2HADomain
Cluster Version        = 2.4.10.0
Cluster State          = Online
Number of nodes        = 2

Node Information:
Node Name              State
-----
harsys1                Online
alaska                 Online

Resource Group Information:
Resource Group Name    = db2_db2inst1_db2inst1_HADB-rg
Resource Group LockState = Unlocked
Resource Group OpState = Online
Resource Group Nominal OpState = Online
Number of Group Resources = 1
Number of Allowed Nodes = 2
  Allowed Nodes
  -----
  harsys1
  alaska

Member Resource Information:
Resource Name          = db2_db2inst1_db2inst1_HADB-rs
Resource State         = Online
Resource Type          = HADR
HADR Primary Instance = db2inst1
HADR Secondary Instance = db2inst1
HADR DB Name           = HADB
HADR Primary Node     = harsys1
HADR Secondary Node   = alaska

Resource Group Name    = db2_db2inst1_alaska_0-rg
Resource Group LockState = Unlocked
Resource Group OpState = Online
Resource Group Nominal OpState = Online
Number of Group Resources = 1
Number of Allowed Nodes = 1
  Allowed Nodes
  -----
  alaska

Member Resource Information:
Resource Name          = db2_db2inst1_alaska_0-rs
Resource State         = Online
Resource Type          = DB2 Partition
DB2 Partition Number   = 0
Number of Allowed Nodes = 1
  Allowed Nodes
  -----
  alaska

Network Information:
No network information found.

Quorum Information:
Quorum Name              Quorum State
-----
tb                        Online
Fail                      Offline
Operator                  Offline

```

## Configure network equivalencies or “db2haicu networks”

---

Since we had configured EtherChannel/bonding on all available Ethernet adapters on each server, there is really no need to create these network equivalencies. If you want to create network equivalencies, see the required steps in the three-node topology configuration.

Now we have successfully implemented our DB2 HA two-node shared DISK tiebreaker cluster topology.

## Testing two-node DB2 HA cluster with shared DISK tiebreaker

---

After implementing this two-node DB2 HA cluster topology with shared DISK tiebreaker, we successfully conducted the following eight test cases to validate the operational status of the cluster.

Test case	Description
Graceful "role-switch" or planned outage	Switch primary and standby HADR roles. Useful when doing “rolling-upgrades”.
Forced "role-switch"	Simulate primary DB2 instance crash and how to recover from it.
Forced "role-switch" using PEER WINDOW ONLY	Validate the HADR_PEER_WINDOW setting. This is the default behavior of the <code>hadrV95_start.ksh</code> script when moving resources to the standby.
Primary node failure	Simulate a brief loss of primary node, and how cluster resources are moved to standby.
Standby node failure	Simulate a brief loss of standby and subsequent reintegration into the cluster after coming back online.
Unexpected primary node failure - hardware/power/single-site failure	Simulate primary node failure for a long period (such as hardware issues) and subsequent reintegration into the cluster as standby after coming back online.
Network interface card (NIC) failures	How NIC failures are handled by switching to the backup adapter in EtherChannel.
Loss of quorum DISK tiebreaker	Simulate a loss of the shared DISK tiebreaker device located in the third site, and its implications to the cluster.

Refer to the [Appendix G - Testing two-node DB2 HA cluster with shared DISK tiebreaker](#) for detailed test procedures for all of the above test cases.

## A note about iSCSI security

---

Note that in our Network topology, we did not have the luxury of having a separate network pipe to the NetApp iSCSI filer. However, in a scaled-down topology of our implementation (for

DB2 system topology and configuration for automated multi-site HA and DR

example, a metropolitan area network), you may already have a separate link available to carry iSCSI traffic. Wherever possible, having a separate and a secure network link to the iSCSI filer is recommended to minimize the risk of security vulnerabilities.

For the most part, iSCSI operates as a clear-text protocol. Though for the purpose of disk tiebreaker data security is irrelevant (since we do not save data on this LUN), it'll expose a security hole that can be exploited. For security reasons, it is advisable to configure the iSCSI filer security controls to provide secure initiator login.

## **Using a Fibre Channel LUN as the shared DISK tiebreaker device**

As we mentioned previously, there is no difference between the setup process and the use of a Fibre Channel LUN compared to an iSCSI LUN. However, the lack of inter-site Fibre Channel SAN infrastructure, basically limited us to using an iSCSI LUN (from a NetApp filer).

While iSCSI LUNs are easier to set up and do not require an expensive intra-site SAN infrastructure, a Fibre Channel LUN will provide more robust DISK tiebreaker operations. The reason is that Fibre Channel, by definition, will be using a different communication path (from Fibre Cabling to SAN switches) from the Ethernet traffic. Hence, a Fibre Channel shared disk LUN has the ability to provide quorum in case of a complete network failure between sites. On the other hand, you will need to set up and manage a complex cross-site SAN zoning infrastructure.

## Alternative multi-site DB2 HA topologies

---

Now we discuss how the two topologies we have designed, tested, and discussed in this paper can be modified to address specific needs based on the network topology available. Note that our treatment of these alternative topologies will be brief, but should provide you with a good idea about how to plan and configure your setup.

For more information about how to implement these two topology variants, refer to *Automated Cluster Controlled HADR (High Availability Disaster Recovery) Configuration Setup using the IBM DB2 High Availability Instance Configuration Utility (db2haicu) and DB 9.7 Information Center*.

### A topology using a private inter-site network trunk

---

In addition to the public cloud between the sites and clients, this topology has a “private” network trunk between the sites. Typically, this type of configuration is limited to a metropolitan area or within a state/province, and will not span across continents like our topologies.

This private trunk (which can be either physical NICs or EtherChannel/bonded NICs) is used for HADR heartbeat while the public trunk is used by database clients and applications. Network adapters used by both can be either single physical NICs or EtherChannel/bonded NICs. However, an EtherChannel/bonded NIC for the public link is strongly recommended to be able to provide seamless link failover capabilities.

This configuration is useful only if the private network path (including switches, routers, Ethernet cabling, etc.) is completely independent of the public network path. Otherwise, it is not very useful since a network failure at one site will bring down both private and public links.

The advantage of this topology is cluster communication redundancy. By configuring db2haicu networks/TSA network equivalencies between the public and private network interfaces on each node, we can configure TSA to monitor NICs and do a NIC failover in case of loss of a network path. Note that these network equivalencies can only be created to group NICs local to the node. The DB2 integrated HA solution does permit network equivalencies between subnets.

Similar to our implementations in this paper, using a Service IP is not supported in this topology. The public interfaces where the clients and application will connect are located on separate physical networks; this restricts TSA’s ability to float the VIP between those sites.

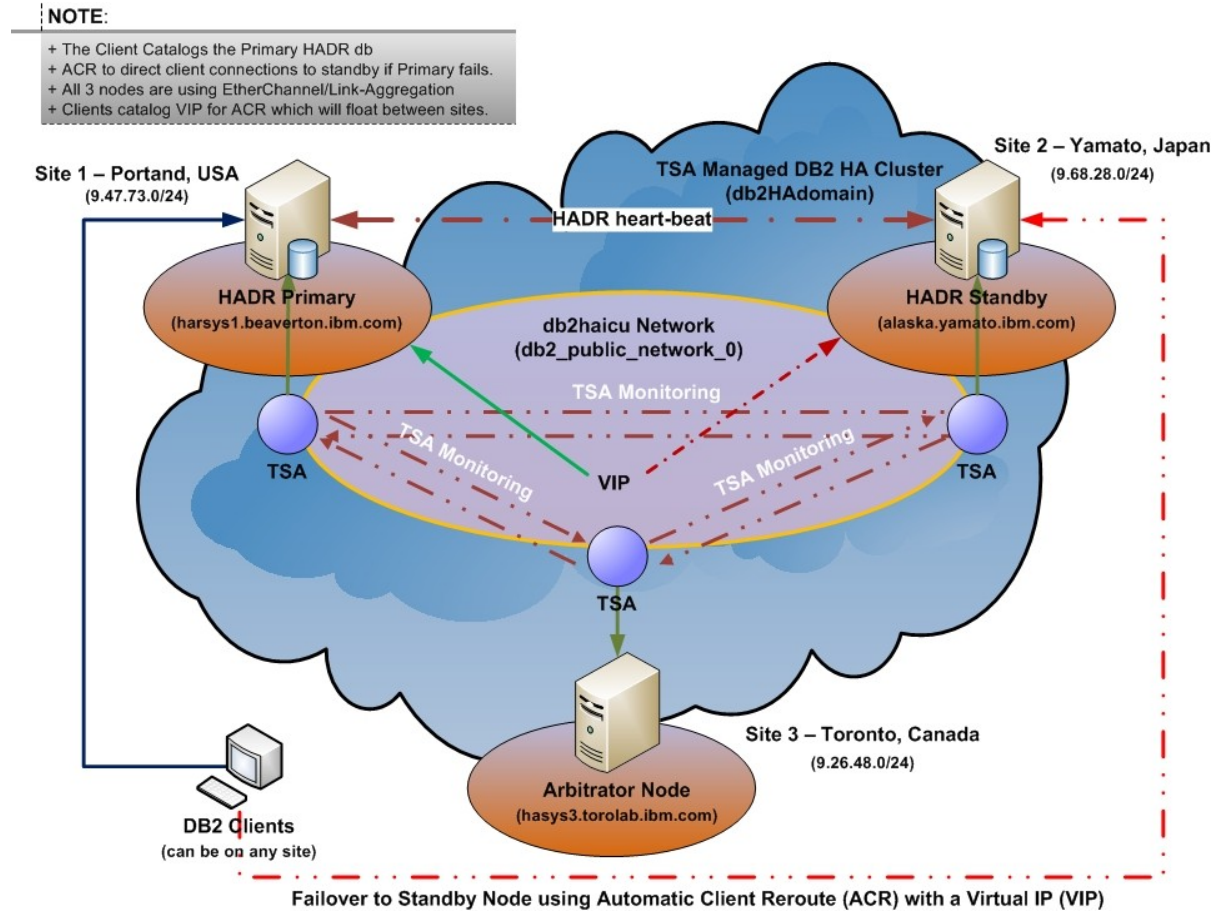
### A cross-site subnet topology using a Virtual IP (VIP)

---

In addition to the public cloud between the sites and clients, this topology has a “private” network trunk between the sites. Typically, this type of configuration is limited to a metropolitan area or within a state/province, and will not span across continents like our topologies.

In this topology, provided that you can create a single subnet which can span across the sites, you can use the Virtual IP (VIP)/Service IP feature available in the DB2 Integrated HADR solution. Such a topology using an arbitrator node is shown in Figure 4.

In this case we have the ability to create one *big* db2haicu network/TSA NIC equivalency group to manage the failover of the VIP.



**Figure 4. Three-Node multi-site DB2 HADR cluster topology with cross-site subnet and VIP**

Cluster nodes connected to this inter-site trunk can have physical NICs or EtherChannel/bonded NICs, but using EtherChannel/bonded NICs is strongly recommended. The big db2haicu network/TSA NIC equivalency group will allow VIP to fail onto any NIC on the same/different nodes until all configured NICs in the cluster are exhausted.

As you can see, the practicability to create such a multi-site subnet when nodes are placed in different continents like our topologies is highly unlikely. Hence, such a topology will mostly be applicable to sites located reasonably close such as within a metropolitan area or a within a state, for example.

## Managing long-distance multi-site DB2 HA clusters

---

When a cluster configuration spans across sites with a public cloud in the middle and private corporate trunks, you need to pay special attention to the behavior of your network topology. In particular, the network must be reliable and any firewalls must continue to allow access to the required ports (as described in this document). A stable network and proper Firewall configuration are critical piece of a successful cluster solution.

## Monitoring RSCT cluster communications

---

Successful cluster operations depend on a stable network environment. Hence it's necessary to continuously monitor the status of the network, especially to record and monitor the latency between each node in the RSCT cluster. In case of higher network latency, by modifying RSCT Communication Group parameters such as, *Sensitivity*, *Heartbeat period*, and *Ping Grace period*, we can improve cluster stability (with respect to network glitches). The following procedure can be used to monitor the “network health” and suggested cluster heartbeat configuration values are given. We suggest that you start with these initial values given below and monitor and tune further based upon the characteristics of your own network topology.

1. Periodically check whether there were missed heartbeats:

- You could run something like `grep Missed /var/ct/<RPD domain name>/log/cthats/nim*`

```
root> grep Missed /var/ct/db2HADomain/log/cthats/nim*
/var/ct/db2HADomain/log/cthats/nim.cthats.en2:08/11 09:22:00.872: Heartbeat was NOT
received. Missed HBs: 1. Limit: 10
/var/ct/db2HADomain/log/cthats/nim.cthats.en2:08/11 09:22:06.882: Heartbeat was NOT
received. Missed HBs: 2. Limit: 10
/var/ct/db2HADomain/log/cthats/nim.cthats.en2:08/11 09:22:12.892: Heartbeat was NOT
received. Missed HBs: 3. Limit: 10
```

- In case any heartbeats (HBs) were missed, you may see messages like: "Heartbeat was NOT received. Missed HBs: 2. Limit: 10".

2. In case you notice a node is offline, check the HATS log file (`/var/ct/<RPD domain name>/log/cthats/cthats.DD.HHMMSS`) to see whether remote nodes were being flagged as down (which should happen if `ConfigRM` responds as though quorum was lost).

- You can run a regular expression like the following to parse out which files/lines contain the message "Sending node [Hb\_Death] notifications.":

```
root> grep 'Sending node \[Hb_Death\] notifications'
/var/ct/db2HADomain/log/cthats/cthats.[0-9]*.*
/var/ct/db2HADomain/log/cthats/cthats.07.153312:08/07 15:48:18.277:hatsd[0]: Sending node
[Hb_Death] notifications.
/var/ct/db2HADomain/log/cthats/cthats.07.154825:08/07 15:53:44.973:hatsd[0]: Sending node
[Hb_Death] notifications.
/var/ct/db2HADomain/log/cthats/cthats.07.155507:08/07 16:05:56.828:hatsd[0]: Sending node
[Hb_Death] notifications.
```

- The logs above indicate that HATS has indeed flagged remote nodes as down. Now look for message "Reachable nodes [...]" prior to the "Sending node [Hb\_Death] notifications." message to see which nodes HATS detected to be alive.

```

root> grep 'Reachable nodes'
/var/ct/db2HADomain/log/cthats/cthats.07.153312
08/07 15:36:51.400:hatsd[0]: Reachable nodes (1 hop)      : 2
08/07 15:48:16.134:hatsd[0]: Reachable nodes (1 hop)      : 1-3
08/07 15:48:18.276:hatsd[0]: Reachable nodes (1 hop)      : 2 <== This means HATS is not
able to reach nodes 1 & 3

```

- Note that the HATS log files are text files, and not binary trace files as other RSCT components.

3. Record the latency between each node in your RSCT cluster and periodically keep track of the latency changes. For example, in our three-node topology:

- We do an ICMP echo test (ping utility) from our arbitrator node to the two HADR nodes:

```

root> ping -c 10 alaska.yamato.ibm.com
PING alaska (9.68.28.248): 56 data bytes
64 bytes from 9.68.28.248: icmp_seq=0 ttl=235 time=191 ms
64 bytes from 9.68.28.248: icmp_seq=1 ttl=235 time=193 ms
64 bytes from 9.68.28.248: icmp_seq=2 ttl=235 time=190 ms

--- alaska ping statistics ---
10 packets transmitted, 9 packets received, 10% packet loss
round-trip min/avg/max = 190/190/193 ms
root> ping -c 10 harsys1.beaverton.ibm.com
PING harsys1 (9.47.73.22): 56 data bytes
64 bytes from 9.47.73.22: icmp_seq=0 ttl=238 time=127 ms
64 bytes from 9.47.73.22: icmp_seq=1 ttl=238 time=137 ms
64 bytes from 9.47.73.22: icmp_seq=2 ttl=238 time=116 ms

--- harsys1 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 113/118/137 ms

```

- Then we do another ICMP echo from one of the HADR nodes to the other HADR node. Sample ICMP results from HADR standby to primary:

```

root> ping -c 10 harsys1.beaverton.ibm.com
PING harsys1: (9.47.73.22): 56 data bytes
64 bytes from 9.47.73.22: icmp_seq=0 ttl=240 time=194 ms
64 bytes from 9.47.73.22: icmp_seq=1 ttl=240 time=175 ms
64 bytes from 9.47.73.22: icmp_seq=2 ttl=240 time=137 ms

----harsys1 PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 129/150/194 ms

```

- Summarize the above results using average round-trip time. In our setup, for example:

```

- HADR Primary <-> HADR Standby      : 150 ms
- Arbitrator <-> HADR Primary         : 118 ms
- Arbitrator <-> HADR Standby        : 190 ms

```

## Tuning RSCT Communication Group parameters

If we notice that there are frequent missed heartbeat events in the cluster, we need to tune the RSCT Communication Group parameters to increase the reliability of the cluster operations.

1. List all the Communication Groups in the cluster:

- List the Communications Groups:

```

root> lscomg
Name Sensitivity Period Priority Broadcast SourceRouting NIMPathName NIMParameters Grace
CG1 4 1 1 Yes Yes
(Default)

```

DB2 system topology and configuration for automated multi-site HA and DR

- Check which nodes belong to which Communication Group:

```
root> lscomg -i CG1
Name NodeName IPAddress Subnet SubnetMask
en2 alaska 9.68.28.248 9.68.28.0 255.255.255.0
en2 hasys3 9.26.48.48 9.26.48.0 255.255.255.0
en2 harsys1 9.47.73.22 9.47.73.0 255.255.255.0
```

**Notes:**

- A NIC/IP can belong to *only* one Communications Group at a time per single node - i.e., if there are multiple NICs on a cluster node, they will be placed in different Communication Groups.
- **Sensitivity:** The number of missed heartbeats that constitute a failure
- **Period:** The number of seconds between heartbeats
- **Ping Grace Period:** Whenever a node loses connection with rest of the cluster nodes, the RSC Topology Services subsystem will issue an ICMP echo to check whether the system is still reachable. If that node responds within the time period set by Ping Grace Period, the cluster will not detect this as a node failure. Note that Ping Grace Period is not really meant for network glitches, but for cases where daemons get blocked because of memory starvation or other factors. We set this value to 30 seconds in both of our cluster topologies.

2. Change the Sensitivity, Heartbeat Period, and Ping Grace Period of the Communication Group CG1

```
root> chcomg -s 10 -p 3000 -g 30000
```

**Notes:**

- Using a sensitivity value of 10 is a reasonable starting point.
- Allowing 3000 ms (3 s) between internal cluster heartbeats is a reasonable starting point.
- Set the Ping Grace Period to 30000 ms (30 s). We see no reason to change this value.

3. Verify that the Communication Group “CG1” is updated with the new parameters:

```
root> lscomg
Name Sensitivity Period Priority Broadcast SourceRouting NIMPathName NIMParameters Grace
CG1 10 3 1 Yes Yes 30
```

## Monitoring Cluster Quorum Status

---

During the normal operational state of the cluster, it is recommended to monitor the *Operational Quorum Status* for the following reasons: 1) to check that there is no loss of quorum device, which can be either the shared disk or the third node, and 2) to make note of which node is the Master node/Group leader.

In the case of Arbitrator node, there will be messages indicating loss of quorum and lost of third node in the SYSLOGs of the two HADR servers. A simple Internet Control Message Protocol (ICMP) echo/ping test is all you need to quickly validate whether the node is down. Note that tiebreaker is never exercised *except* in the case where there is a potential *split brain* (e.g., in a two-node cluster topology when one of the nodes goes down, only then is tiebreaker code exercised).

However, loss of a shared DISK quorum device may not be particularly visible after the cluster has achieved “HAS QUORUM” state.

DB2 system topology and configuration for automated multi-site HA and DR

This is due to the following factors:

- AIX ODM is refreshed only during a boot up or by running configuration manager (`cfgmgr`). Hence, even if the tiebreaker LUN is offline, that `hdisk` will still be visible from AIX host.
- RSCT produces no indication at all that the quorum device access was lost until `ConfigRM` tries to issue a SCSI reservation on the DISK tiebreaker device.
- Hence, when the two nodes are “in quorum” there will be no errors/warnings logged in SYSLOGs if the LUN used as the shared DISK quorum tiebreaker device is offline.

Therefore, the only time we will notice the loss of the DISK tiebreaker is when one of the nodes has failed and the other node tried to acquire a lock on the tiebreaker device. In this case, it will fail, an error will be logged into SYSLOG, and this node will also be rebooted.

## Monitoring the quorum DISK tiebreaker

A fairly rudimentary way to monitor the quorum disk tiebreaker device is to use the `lspath` command in AIX environments to periodically verify (using a cron job) if the `hdisk#` used by the tiebreaker is accessibly on both nodes. A failure on one/or both nodes can be logged into SYSLOG and in addition generate an alert (by e-mail or SNMP trap) to a system administrator.

The following shell script (`checktb.ksh`) will log to SYSLOG if the disk tiebreaker device is not accessible from AIX and also sent an email alert to the system administrator.

### checktb.ksh

```
#!/bin/ksh -p
#
# Set the shared DISK quorum tiebreaker device name
# You can use lsrsrc -s "Name='tb'" IBM.TieBreaker to find out hdisk#
tb=hdisk4
#
# Set the Admin email
admin=admin@mydomain.com
#
# Check if tb is accessible and notify if error
lspath -l ${tb} | grep -i Failed
if [ ${ $? } -ne 0 ]; then
logger -i -p debug -t $0 "Unable to access quorum DISK tiebreaker ${tb}"
mail -s "Unable to access quorum DISK tiebreaker ${tb}" $admin
fi
exit 0
```

The above script can be saved, for example, under `/usr/sbin/rsct/sapolicies/db2` and then scheduled to be run using a cron job every 15 minutes on both nodes. For example, we used the following cron job on our cluster nodes:

```
0,15,30,45 * * * * /usr/sbin/rsct/sapolicies/db2/checktb.ksh
```

**Note:** Make sure the above script is executable by root (i.e., `chmod 755 checktb.ksh`)

The above cron job, for example, will log into SYSLOGs something similar to following:

```
Sep 14 12:15:00 harsys1 user:debug /usr/sbin/rsct/sapolicies/db2/checktb.ksh[663636]: Unable to
access quorum DISK tiebreaker hdisk3
Sep 14 12:45:00 harsys1 user:debug /usr/sbin/rsct/sapolicies/db2/checktb.ksh[663720]: Unable to
access quorum DISK tiebreaker hdisk3
Sep 14 13:00:00 harsys1 user:debug /usr/sbin/rsct/sapolicies/db2/checktb.ksh[688270]: Unable to
access quorum DISK tiebreaker hdisk3
```

## Taking note of the Master node/Group Leader

By checking the status of Recovery and Config Resource managers we can get this information.

```
root> lssrc -ls IBM.RecoveryRM
Subsystem      : IBM.RecoveryRM
PID            : 368854
Cluster Name   : db2HADomain
Node Number    : 1
Daemon start time : 08/31/09 15:44:33
```

```
Daemon State:
My Node Name   : harsysl
Master Node Name : alaska (node number = 2)
Our IVN        : 2.2.0.7
Our AVN        : 2.2.0.7
Our CVN        : 341251747221 (0x224a9c2595)
Total Node Count : 2
Joined Member Count : 2
Config Quorum Count : 2
Startup Quorum Count : 1
Operational Quorum State: HAS_QUORUM
In Config Quorum : TRUE
In Config State  : TRUE
```

```
root> lssrc -ls IBM.ConfigRM
Subsystem      : IBM.ConfigRM
PID            : 463100
Cluster Name   : db2HADomain
Node Number    : 1
Daemon start time : 08/31/09 15:40:28
```

Daemon State: Online in db2HADomain, pinned, security disabled

```
ConfigVersion: 0x84a9c4a21
Group IBM.ConfigRM:
  Providers: 2
  GroupLeader: alaska, 0xff0b5e225a2f931d, 2
```

## Monitoring operational status of RSCT Topology/Group Services

During the normal operational state of the cluster, it is also beneficial to periodically query the status of the RSCT Topology Services and Group Services subsystems.

1. RSCT Topology Services monitors the networks that correspond to the communication groups set up by the configuration resource manager (ConfigRM). To see the status of the networks, issue the following command on a cluster node that is online:  
root> **lssrc -ls cthats**
2. The Group Services provides other RSCT applications and subsystems within an RSCT peer domain with a distributed coordination and synchronization service. You can display the operational status of the Group Services daemon by issuing:  
root> **lssrc -ls cthags**

Note that if you get a message that there has been no heartbeat connection for some time, it could mean that the Topology Services subsystem is not running.

## Disabling High Availability

---

To disable the HA configuration for a particular instance, the `db2haicu -disable` command can be used. After issuing this command, the system will not respond to any failures and all resource groups for the instance will be locked. Any maintenance work can be performed in this state without worrying about cluster manager intervention.

To enable HA, just issue the `db2haicu` command again, and choose the “Yes” option when prompted to continue.

## Manual takeovers

---

There might be situations when a DBA wants to perform a manual takeover to switch HADR database roles. For more information:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0011553.html>

To do this, log on to the standby machine and type in the following command to perform a manual takeover:

```
dba> db2 takeover hadr on db <database name>
```

Once the takeover has been completed successfully, the “`lssam`” or the “`db2pd -ha`” commands will reflect the changes.

## The db2haicu maintenance mode

---

When a system is already configured for High Availability, `db2haicu` runs in maintenance mode. Typing `db2haicu` on the primary or standby node will produce the menu shown below.

```
/home/db2inst1% db2haicu
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu).

You can find detailed diagnostic information in the DB2 server diagnostic log file called
db2diag.log. Also, you can use the utility called db2pd to query the status of the cluster
domains you create.

For more information about configuring your clustered environment using db2haicu, see the
topic called 'DB2 High Availability Instance Configuration Utility (db2haicu)' in the DB2
Information Center.

db2haicu determined the current DB2 database manager instance is db2inst1. The cluster
configuration that follows will apply to this instance.

db2haicu is collecting information on your current setup. This step may take some time as
db2haicu will need to activate all databases for the instance to discover all paths ...
When you use db2haicu to configure your clustered environment, you create cluster domains.
For more information, see the topic 'Creating a cluster domain with db2haicu' in the DB2
Information Center. db2haicu is searching the current machine for an existing active cluster
domain ...
db2haicu found a cluster domain called hadr_domain on this machine. The cluster configuration
that follows will apply to this domain.

Select an administrative task by number from the list below:
 1. Add or remove cluster nodes.
 2. Add or remove a network interface.
 3. Add or remove HADR databases.
 4. Add or remove an IP address.
 5. Move DB2 database partitions and HADR databases for scheduled maintenance.
 6. Create a new quorum device for the domain.
 7. Destroy the domain.
 8. Exit.
Enter your selection:
```

This menu can be used to carry out various maintenance tasks and change any Cluster Manager-specific, DB2-specific or network-specific values configured during the initial setup.

DB2 system topology and configuration for automated multi-site HA and DR

## Stopping and starting the entire RSCT domain

---

During the operational life of the cluster, you may need to stop the entire RSCT domain. This is the procedure if you wish to stop and start the entire cluster domain.

### Stopping the domain

Now we are in the process of stopping the domain.

1. Before stopping the domain check the current status of the cluster resources by issuing:

```
root> lssam
```

```
root> lsrpdomain
```

```
Name OpState RSCTActiveVersion MixedVersions TSPort GSPort
db2HADomain Online 2.4.10.0 No 12347 12348
```

2. We must now take the resources Offline. This is done by issuing the following command (with root authority) on either the standby or the primary nodes:

```
root> chrg -o Offline -s "Name like '%'"
```

3. Wait one or two minutes for all the resources to come Offline, validated by issuing **lssam** command.

4. To bring the domain Offline, run the “stoprpdomain” command, for example:

```
root> stoprpdomain db2HADomain
```

```
root> lsrpdomain
```

```
Name OpState RSCTActiveVersion MixedVersions TSPort GSPort
db2HADomain Offline 2.4.10.0 No 12347 12348
```

```
root> lssam
```

```
lssam: No resource groups defined or cluster is offline!
```

### Starting the domain

Now we are in the process of starting the domain.

1. To start the domain we use the command “starttrpdomain”, for example:

```
root> starttrpdomain db2HADomain
```

2. Now check the current status of the cluster resources by issuing:

```
root> lsrpdomain
```

```
Name OpState RSCTActiveVersion MixedVersions TSPort GSPort
db2HADomain Online 2.4.10.0 No 12347 12348
```

```
root> lssam
```

3. Now, we must bring the resources Online. This is done by issuing the following command (with root authority) on either the standby or the primary nodes:

```
root> chrg -o online -s "Name like '%'"
```

4. Wait one or two minutes for all the resources to come Online, validated by issuing **lssam** command.

Now we have safely stopped and started the domain using the appropriate commands.

## Removing a RSCT peer domain

---

During the initial testing/deployment phase of the HA cluster configuration, you may need to remove the entire RSCT cluster. There are two methods you can use to remove a RSCT cluster.

### Using db2haicu '-delete' option

The `db2haicu` tool can also be run with the “-delete” option. This option removes a system’s entire HA configuration and deletes all resources in the cluster for the instance in question. If no other instance is using the domain at the time, the domain is deleted as well.

As a good practice, it is recommended to run `db2haicu` with the delete option on an instance before it is made highly available. This makes sure that we are starting from scratch and not building on top of leftover resources.

For example, when running `db2haicu` with an XML file, any invalid attribute in the file will cause `db2haicu` to exit with a non-zero error code. However, before `db2haicu` is run again with the corrected XML file, one can run the `-delete` option to make sure that any temporary resources created during the initial run are cleaned up.

Note that the `db2haicu -delete` option will leave the instances and the HADR replication unaffected. That is, it will not stop the db2 instances or HADR replications. However, any IP addresses that were highly available are removed and no longer present after the command completes.

To use "`db2haicu -delete`", the *RSCT peer domain must be Online*. If for some reason is cluster is in an inconsistent state (e.g. nodes are down or the whole peer domain is down), you need to use RSCT commands to remove that peer domain.

### Using RSCT commands

Removing a peer domain involves removing the peer domain definition from each node on the peer domain. You can remove the peer domain definition by issuing the `rmrpdomain` command with the name of the peer domain, from any online node in the peer domain. If all the nodes are reachable, then the command will attempt to remove the peer domain definition from all nodes.

If a node is not reachable from the node where the `rmrpdomain` is run (for example, the network is down or the node is inoperative), you will need to run the `rmrpdomain` command from each node that did not have their peer domain definition removed. Include the `-f` option to force the removal: `rmrpdomain -f db2HADomain`.

You can also use the `-f` flag if an RSCT subsystem (such as Topology Services or Group Services) rejects the `rmrpdomain` command because a peer domain resource is busy. The `-f` flag will force the RSCT subsystems to take the peer domain offline and remove the peer domain definitions regardless of the state of peer domain resources.

## Troubleshooting unsuccessful failovers

---

In the case when a critical failure occurs on the primary cluster node, a failover action is initiated on the HADR resource group, and all HADR resources are moved to the standby machine. If such a failover operation is unsuccessful, it will be reflected by the fact that all HADR resources residing on both the primary and the standby machines are “Failed Offline”. This can be due to the following reasons:

### Insufficient HADR\_PEER\_WINDOW database configuration parameter value

When moving the HADR resources during a failure, the Cluster Manager issues the following command on the standby database:

```
db2 takeover hadr on db <database name> by force peer window only
```

The peer window value is the amount of time that a takeover must be done on the standby database from the time when the primary database failed. If a takeover is not done within this “window”, the above-mentioned takeover command will fail, and the standby database will not be able to assume the primary role. Hence, if a takeover fails on your system, you might have to update this parameter to a larger value, and try the failure scenario again. Also, make sure that the dates and times on both the standby and the primary machines are synchronized. See the section [Considerations for configuring the HADR\\_TIMEOUT and HADR\\_PEER\\_WINDOW](#) for more details on how to determine a good value for the peer window.

If peer window expiration is what caused the takeover to fail, a message indicating this would be logged in the DB2 diagnostic log. At this point, you can issue the following command at the standby machine and force HADR takeover on the standby:

```
db2 takeover hadr on db <database name> by force
```

However, prior to issuing the above command, you are urged to consult the URL:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0011553.html>

### The HADR resource group is in the “Locked” state

A lock on the HADR resource group indicates the fact that no replication is being carried out between the HADR databases and that they are not in “peer” state. In such a state, if a primary node is to fail, a failover will not be initiated. This is because at this point, the standby database cannot be trusted to be the complete copy of the primary, and hence not fit to take over.

### Time of the Day clocks on HADR nodes are out-of Sync

Ensure that the two machines (hosting the primary and standby HADR databases) are synchronized with respect to time. Ensure that the “TZ=GMT date” command, when issued simultaneously at both nodes, will return results (calculated for GMT) that are equal (to within 5 seconds). If possible configure both servers to a NTP server local to the time-zone of each node.

### The “SYSLOG” and the DB2 server diagnostic log file (db2diag.log)

For debugging and troubleshooting unsuccessful failovers or any other DB2 HA Cluster operation failures, you must examine data logged into two files. The SYSLOG for information from the cluster manager, and the DB2 server diagnostic log file for DB2 HA related errors.

## Appendix A - List of references for more information

---

1. **IBM white paper: Automated Cluster Controlled HADR (High Availability Disaster Recovery) Configuration Setup using the IBM DB2 High Availability Instance Configuration Utility (db2haicu)** by Steve Raspudic, Malaravan Ponnuthurai (IBM Canada Ltd./IBM Toronto Software Lab), June 2009
  - This paper deal with creating automated DB2 HA and DR topologies using the DB2 Integrated HA solution in a local area network (LAN) topology.
  - PDF download: <http://www.ibm.com/developerworks/data/library/long/dm-0907hadrd2haicu/>
2. **IBM Redbooks® publication: *High Availability and Disaster Recovery Options for DB2 on Linux, UNIX, and Windows***, by Whei-Jen Chen, Masafumi Otsuki, Paul Descovich, Selvaprabhu Arumuggharaj, Toshihiko Kubo, and Yong Jun Bi (IBM Corporation), February 2009.
  - A reference for a wide range of strategies and solutions involving High Availability and Disaster Recovery for databases running IBM DB2 for Linux, UNIX, and Windows. <http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247363.html>
3. **IBM Tivoli System Automation for Multiplatforms (Version 2 Release 2)** product/technical documentation:
  - **Base Component Administrator's and User's Guide**
  - **Installation and Configuration Guide**
  - **Base Component Reference**
  - <http://publib.boulder.ibm.com/tividd/td/IBMTivoliSystemAutomationforMultiplatforms2.2.html>
4. **Reliable Scalable Cluster Technology (RSCT) Administration Guide**
5. **IBM pSeries and Cluster Information Centers on the Web**
  - pSeries: <http://publib16.boulder.ibm.com/pseries/index.htm>
  - Cluster: <http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp>
6. **IBM DB2 9.5 and DB2 9.7 for Linux, UNIX, and Windows Information Centers on the Web**
  - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>
    - The place for all the latest information about IBM DB2 9.7 for this platform.
  - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp>
    - The place for all the latest information about IBM DB2 9.5 for this platform.

## Appendix B - DBA's checklist to cluster planning

We include the following tables/guidelines as a cluster planning tool for database administrators (DBAs). You may need to consult the Network, Storage, and Systems Administration personnel to collect the following information.

### 1. Cluster node information:

Cluster node hostname	Node IP	Domain	Default Gateway IP	Primary DNS IP	Role
					HADR Primary
					HADR Standby
					Arbitrator

**Note:** The Arbitrator node information is required only if you plan to implement the three-node cluster topology.

### 2. iSCSI filer information: Only required for two-node shared DISK tiebreaker topology.

iSCSI filer IP	IQN	Port

### 3. RSCT/HADR Ports information: This is essential to set up Firewall rules

hostname [source IP address]	hostname [destination IP address]	Application name (/etc/services)	Port number	Remark (usage)
		cthats	12347/udp	RSCT cluster port for CTHATS daemon
		cthags	12348/udp	RSCT cluster port for CTHAGS daemon
		rnc	657/tcp, 657/udp	RSCT cluster port for RMC daemon
				DB2 listener port (SVCENAME)
				HADR heartbeat ports
				HADR heartbeat ports
				Port for iSCSI NetApp filer

### 4. DB2 HADR Configuration Parameters:

DBM cfg parameter	Primary node	Standby node
HADR_LOCAL_HOST		
HADR_LOCAL_SVC		
HADR_REMOTE_HOST		
HADR_REMOTE_SVC		
HADR_REMOTE_INST		
HADR_SYNCMODE	NEARSYNC	NEARSYNC
HADR_TIMEOUT		
HADR_PEER_WINDOW		

### 5. Network Latency between sites: You can use ICMP echo (ping) to obtain this information.

Source <-> Destination	Latency (ms)
HADR Primary <-> HADR Standby	
Arbitrator <-> HADR Primary	
Arbitrator <-> HADR Standby	

**Note:** The Arbitrator node information is required only if you plan to implement the three-node cluster topology.

## Appendix C - Configuring EtherChannel

---

**Note:** you need to be logged in as root using a Hardware Management Console (HMC) to perform these activities. Do not use a terminal session (i.e., `telnet`), since the Ethernet interfaces need to be detached and brought down.

1. Since both NICs on each node are already configured, we need to first detach each interface.

```
root> ifconfig en0 detach
root> ifconfig en1 detach
```

2. Next we need to bring down the above two network interfaces:

```
root> chdev -P -l en0 -a state=down
root> chdev -P -l en1 -a state=down
```

3. Create the EtherChannel adapter `en2`:

```
root> mkdev -c adapter -s pseudo -t ibm_ech -a adapter_names=ent0 -a
backup_adapter=ent1 -a netaddr=9.68.28.1 -a num_retries=3 -a retry_time=3
```

**Note:** If you use "SMIT" to create the EtherChannel using `smitty etherchannel -> Add An EtherChannel / Link Aggregation`, skip Step 4 and execute the command in Step 6 instead.

4. The "Standard Ethernet Network Interface" is not automatically created when EtherChannel is configured. Hence we need to create it manually and associate it with "Internet Network Extension" (`inet0`):

```
root> mkdev -c if -s EN -t en -a netaddr=9.68.28.248 -a
netmask=255.255.255.0 -w en2 -a state=up -a arp=on mkdev -l inet0
```

5. Verify whether the "Standard Ethernet Network Interface" (`en2`) was created when the EtherChannel adapter was created:

```
root> lsdev -C | grep -i en2
en2          Available          Standard Ethernet Network Interface
```

6. Make sure "short name" (i.e., `hostname -s`) is associated with `inet0`

```
root> chdev -l inet0 -a hostname=`uname -a | awk '{print $2}'`
```

7. **[OPTIONAL]** If you used SMIT menus to create the EtherChannel adapter, configure the "Standard Ethernet Network Interface" associated with the EtherChannel adapter with the IP/Net-Mask values and activate:

```
root> chdev -l en2 -a netaddr=9.68.28.248 -a netmask=255.255.255.0 -a
state=up -a arp=on
```

**Note:** You can do the same operation using `smitty chinnet`

8. Verify that Default Gateway and Name Server Information is properly set for `en2` adapter by: `smitty tcpip -> Minimum Configuration & Startup ->` And select the "Standard Ethernet Network Interface" associated with the EtherChannel adapter we created in Step 4.

9. Repeat the same process on the rest of the nodes.

**Reference:** [http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.commadmn/doc/commadmndita/etherchannel\\_config.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.commadmn/doc/commadmndita/etherchannel_config.htm)

DB2 system topology and configuration for automated multi-site HA and DR

## Appendix D - db2haicu XML input file

---

We used the following XML file as the input to the `db2haicu` tool, to create the initial cluster with the two HADR nodes in both topologies.

**Note:** This sample XML file below was used to configure the “Primary” node. The XML file used on the standby node was identical except for the “<HADRDBSet>” stanza. You must switch the values of the `localHost` and `remoteHost` parameters, when using it on the other node.

```
<HADRDBSet>
  <HADRDB databaseName="HADB" localInstance="db2inst1"
    remoteInstance="db2inst1" localHost="harsys1" remoteHost="alaska" />
</HADRDBSet>

<?xml version="1.0" encoding="UTF-8"?>
<!--
*****
** Licensed Materials - Property of IBM
**
** (C) COPYRIGHT International Business Machines Corp. 2007
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
**
** SOURCE FILE NAME: db2ha_sample_HADR.xml
**
** SAMPLE: Initial Setup Configuration of DB2 HA HADR failover automation
** using db2haicu
**
** FUNCTION: This sample showcases the way in which one can write XML
** configuration file for HADR failover automation using db2haicu HA utility
**
** USAGE: db2haicu -f db2ha_sample_HADR.xml
**
** DESCRIPTION: The environment for this sample is described below:
**   1. The physical topology is already setup and available
**       The hardware topology is includes the following
**         - 2 computers
**         - 1 NIC/box
**   2. DB2 has to be installed on both nodes (hasys01 and hasys02)
**   3. Cluster manager has to be installed and running on both nodes
**   4. HADR is configured and initialized
**
** PREREQUISITES:
**   1. Hardware installed and configured (physical networks)
**   2. IP addresses reserved or assigned
**   3. OS is installed, patched and configured
**   4. Users, groups and authentication set-up on both machines
**   5. TSA v2.2 installed and configured on both nodes
**   6. Gathering the information on hardware specifications like IP, NIC
**   7. Root privilege is required while installing the DB2
**
```

DB2 system topology and configuration for automated multi-site HA and DR

```

*****
*****
-->
  <!-- ===== -->
  <!-- = DB2 High Availability configuration schema = -->
  <!-- = Schema describes the elements of DB2 High Availability = -->
  <!-- = that are used in the configuration of a HA cluster = -->
  <!-- ===== -->
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="db2ha.xsd" clusterManagerName="TSA"
version="1.0">
  <!-- ===== -->
  <!-- = ClusterDomain element = -->
  <!-- = This element encapsulates the cluster configuration = -->
  <!-- = specification = -->
  <!-- ===== -->
  <ClusterDomain domainName="db2HADomain">
    <!-- ===== -->
    <!-- = ClusterNodeName element = -->
    <!-- = The set of nodes in the cluster domain = -->
    <!-- ===== -->
      <ClusterNode clusterNodeName="harsys1"/>
      <ClusterNode clusterNodeName="alaska"/>
    </ClusterDomain>
    <!-- ===== -->
    <!-- = Failover policy element = -->
    <!-- = The failover policy specifies the failover order of the = -->
    <!-- = cluster nodes = -->
    <!-- = In the current sample the failover policy is to restart = -->
    <!-- = instance in place (LocalRestart) = -->
    <!-- ===== -->
    <FailoverPolicy>
      <HADRFailover></HADRFailover>
    </FailoverPolicy>
    <!-- ===== -->
    <!-- = DB2 Partition element = -->
    <!-- = The DB2 partition type specifies a DB2 Instance Name, = -->
    <!-- = partition number = -->
    <!-- ===== -->
    <DB2PartitionSet>
      <DB2Partition dbpartitionnum="0" instanceName="db2inst1">
        </DB2Partition>
      </DB2PartitionSet>
      <!-- ===== -->
      <!-- = HADRDBSet = -->
      <!-- = Set of HADR Databases for this instance = -->
      <!-- = Specify the database name, the name of the local instance on = -->
      <!-- = this machine controlling the HADR database, the name of the = -->
      <!-- = remote instance in this HADR pair, the name of the local = -->
      <!-- = hostname and the remote hostname for the remote instance = -->
      <!-- ===== -->
      <HADRDBSet>
        <HADRDB databaseName="HADB" localInstance="db2inst1"
          remoteInstance="db2inst1" localHost="harsys1" remoteHost="alaska" />
      </HADRDBSet>
    </DB2Cluster>

```

## Appendix E - Configuring NetApp iSCSI shared LUN for the two AIX hosts

---

For the two-node shared DISK tiebreaker topology we used a single shared iSCSI LUN off a NetApp filer.

1. Open the NetApp Web Administration Console (*FilerView for OnTap 7.2.3*) and add a new Volume:

- Launch the **Volume Wizard** by selecting **Volume -> Add** from left-pane
- Create a flex volume (e.g. `ha_shared_disk_tb_vol`), set volume name and click next.
- Select an aggregate value (e.g. we chose aggregate `aggr0`) and set "Space Guarantee" option to that volume and click next. Note that you must always create a flex volume inside an aggregate.
- Set the Volume size and, Snapshot Reserve size - we chose 20 MB as the volume size (which is the minimum allowed volume size on the NetApp filer).

The screenshot shows a web browser window titled "svt3070a: Volume Wizard - Microsoft Internet Explorer". The main content area is titled "Volume Wizard - Flexible Volume Size". It contains three sections:

- Volume Size Type:** A radio button selection between "Total Size" (selected) and "Usable Size". Below the text: "Select **Total Size** to enter the total volume size (including snap reserve) and **Usable Size** to enter the usable volume size (excluding snap reserve)." There are help icons next to each option.
- Volume Size:** A text input field containing "20" and a dropdown menu set to "MB". Below the text: "Enter the desired volume size. The containing aggregate, **aggr0** has a maximum of 366 GB space available." The value "366 GB (Max)" is displayed on the right.
- Snapshot Reserve:** A text input field containing "0" and a percentage sign "%". Below the text: "Enter the snapshot reserve for volume **ha\_shared\_disk\_tb\_vol**! The range is between 0% and 50%. The default is 20%." There is a help icon next to the field.

At the bottom of the form are three buttons: "< Back", "Cancel", and "Next >".

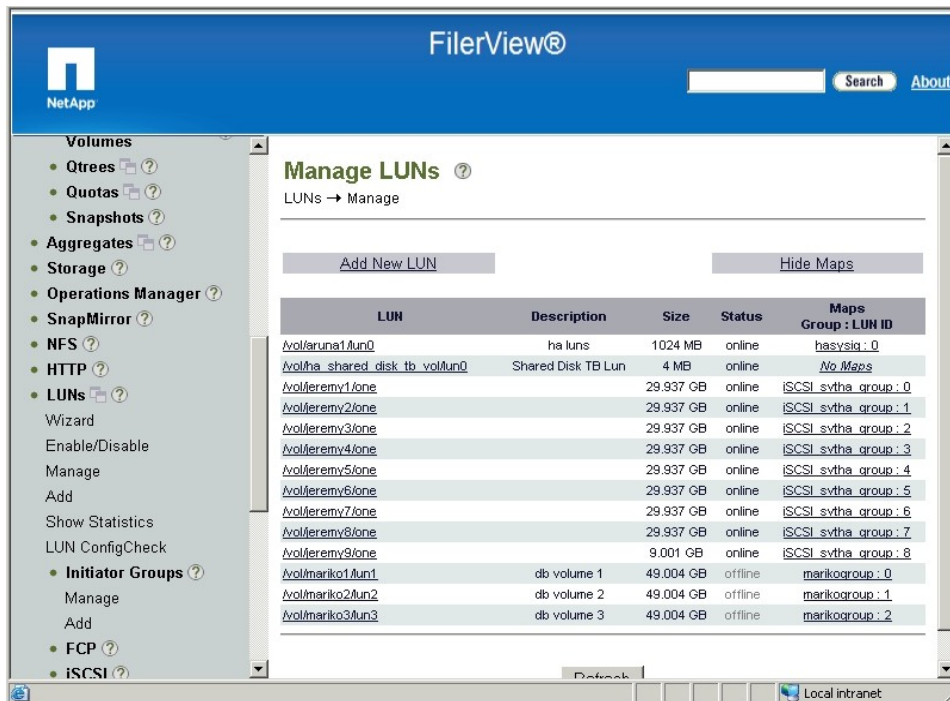
- Commit the values and exit Volume Wizard.

2. Create a LUN on the Volume we defined above using **LUNs -> Add**:

- Select the **LUN Protocol Type** as AIX.
- We used 4 MB as the size of the LUN (which is the minimum allowed LUN size on the NetApp filer) and selected *Space Reserved* option.
- Also named the LUN `/vol/ha_shared_disk_tb_vol/lun0`. Note that LUN names must be in the form of `/vol/<volume name>/<lun name>`.
- Click the Add button to create a LUN on the volume we defined earlier.

DB2 system topology and configuration for automated multi-site HA and DR

- Verify that the LUN was created using **LUNs -> Manage**. **Note** at this time, there will be no mapping beside the LUN `/vol/ha_shared_disk_tb_vol/lun0`.

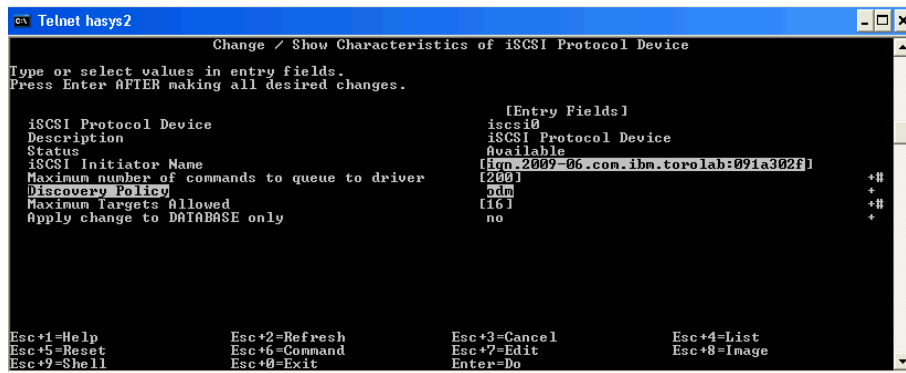


3. Configure the iSCSI node name (IQN) and "Discovery Policy" of the iSCSI protocol device on the two AIX HADR nodes:

- Look up the IQN number on the two AIX hosts:
 

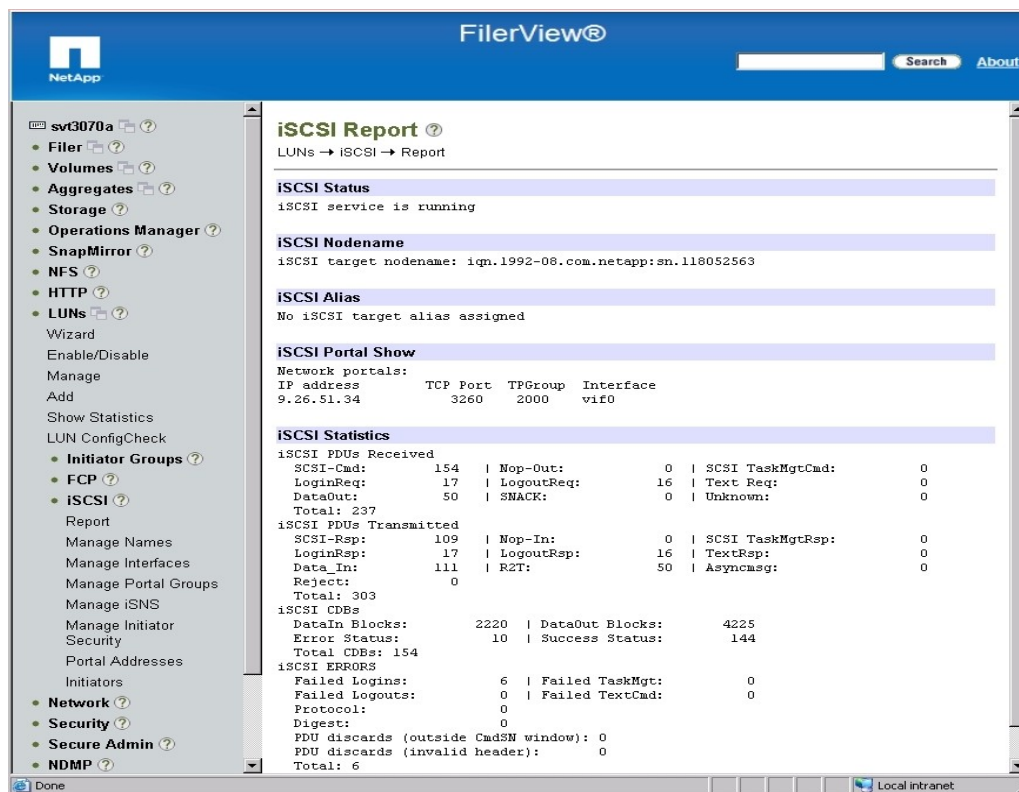
```
root> lsattr -El iscsi0 | grep -i initiator_name
initiator_name iqn.alaska.hostid.09441cf8 iSCSI Initiator Name
True
```
- To add an iSCSI node to an initiator group on the NetApp filer, we need to modify the default IQN value in the AIX ODM to conform to the following format:
 

```
iqn.yyyy-mm.backward_naming_authority:unique__device_name
yyyymm: Month and year in which the naming authority acquired the domain name
backward_naming_authority: Reverse domain name, e.g. com.ibm
unique__device_name: Free-format unique name assigned to this device by naming authority
```
- Hence we used a modified IQN value - e.g., `iqn.2009-08.com.ibm.yamato:09441cf8` when creating the initiator group. In our HADR configuration, we updated iSCSI protocol devices with the following values :
  - On HADR primary: `root> chdev -l 'iscsi0' -a initiator_name='iqn.2009-08.com.ibm.beaverton:092f4916' -a disc_policy='odm'`
  - On HADR standby: `root> chdev -l 'iscsi0' -a initiator_name='iqn.2009-08.com.ibm.yamato:09441cf8' -a disc_policy='odm'`
- **Note:** You can do the same using `smitty iscsi -> iSCSI Protocol Device -> Change / Show Characteristics of an iSCSI Protocol Device -> select iSCSI device -> change the Initiator name (IQN) and discovery policy`



4. Configure the iSCSI target device on the AIX hosts.

- We need the IQN, port number, and IP address of the NetApp filer for this.
- Open the NetApp Web Administration Console (FilerView for OnTap 7.2.3)
- Look at the NetApp filer's IQN number and port using **LUNs->iSCSI->Report**, under the sections **iSCSI Nodename** and **iSCSI Portal Show**.

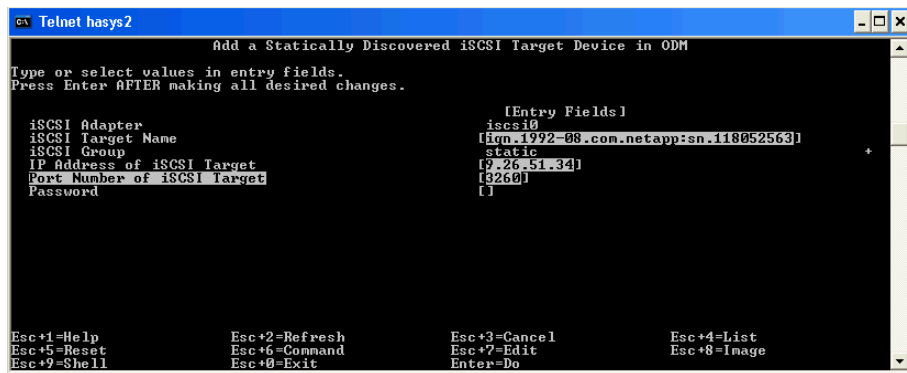


- Using the values you got from NetApp filer's report for IQN, port number, and IP address, add the NetApp filer as an iSCSI Target Device in AIX ODM on the two HADR nodes:

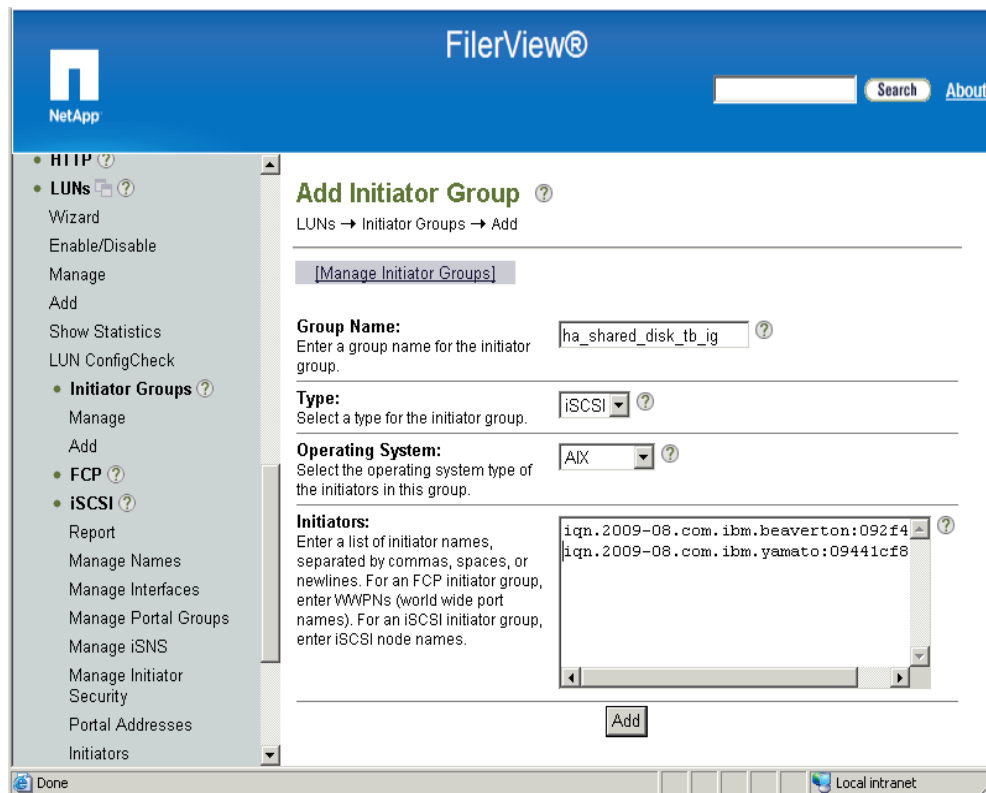
```

root> mkiscsi -l 'iscsi0' -t 'iqn.1992-08.com.netapp:sn.118052563' -g
'static' -i '9.26.51.34' -n '3260'
  
```

- **Note:** You can do the same using `smitty iscsi` -> **iSCSI Target Device Parameters in ODM** -> **Add an iSCSI Target Device in ODM** -> **Add a Statically Discovered iSCSI Target Device in ODM**

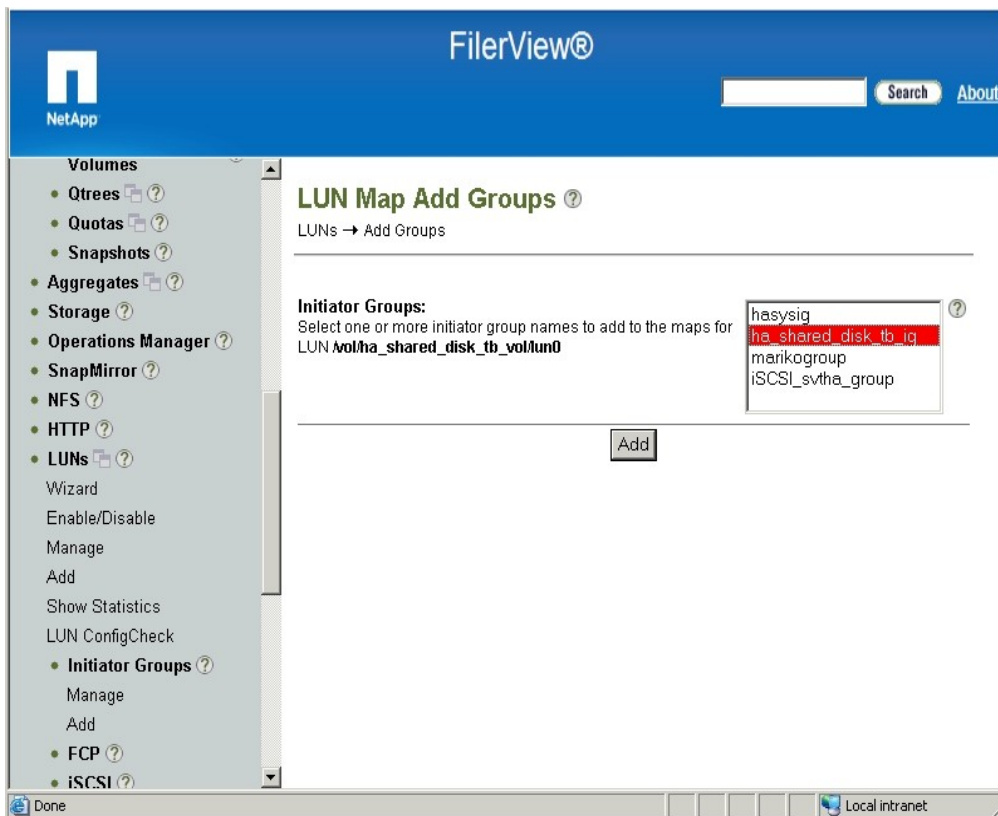


5. From the NetApp Web Administration Console, create a new LUN initiator group:
  - Add the LUN to the initiator group using **LUNs-> Initiator Groups->Add**:
  - Specify a name for the initiator group, for example, `ha_shared_disk_tb_ig`
  - Select the type of the initiator group as “iSCSI” and operating system type of the initiators in this group as “AIX”.
  - Add the names (iqn) of the two initiators created in step 4.

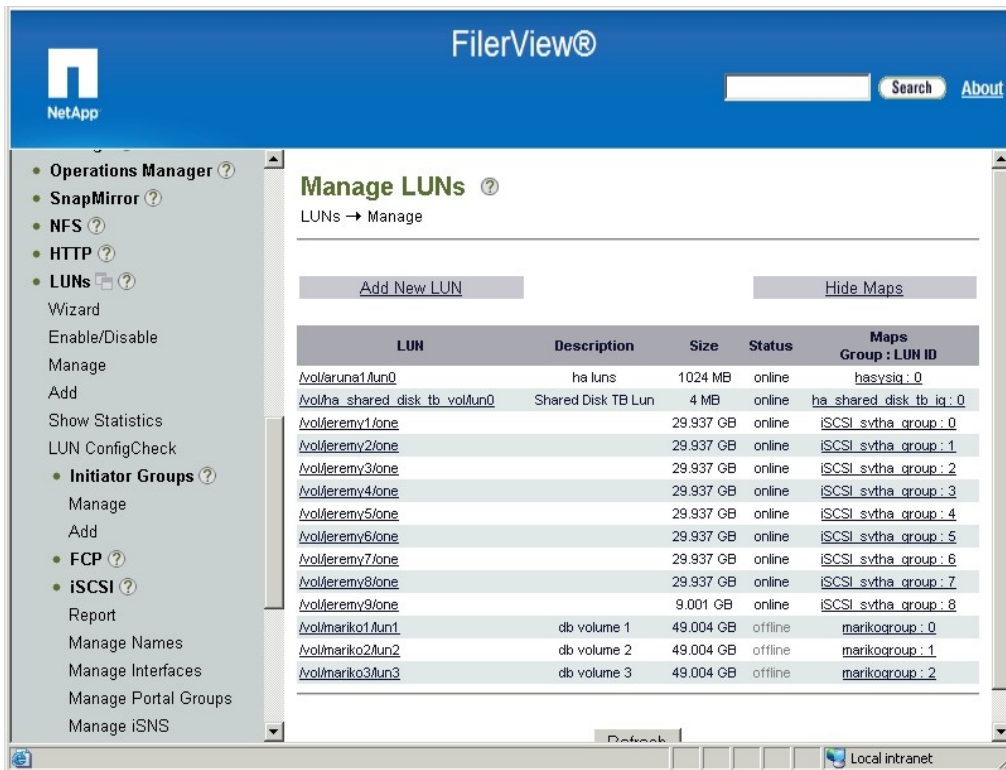


- Click **Add** to create the initiator group.

6. Map the Volumes configured in Steps 1-2 to the iSCSI LUN initiator group we configured in the above step.
  - Go Back to **LUNs** -> **Manage**, and click the link that says "No Maps" beside your LUN's name (e.g., /vol/ha\_shared\_disk\_tb\_vol/lun0) to launch the screen where you can add maps.
  - On the next screen, click the “**Add Groups to Map**” link.
  - Select the initiator group you want to add. For example, we previously created a LUN initiator group `ha_shared_disk_tb_ig`, so select that. Click **Add**.



- Do not add any LUN IDs to the mapping. Click **Apply** to create the mapping.
- Go Back to **LUNs** -> **Manage**. You should now see the mapping created above.



## 7. Expose the new shared iSCSI LUN to each AIX host:

- Run Configuration Manager to discover the new LUN:

```
root> cfmgr -v
```

- Verify that you can see the new device from AIX:

```
root> lsdev -Ccdisk
```

```
hdisk0 Available 01-08-00-3,0 16 Bit LVD SCSI Disk Drive
hdisk1 Available 01-08-00-4,0 16 Bit LVD SCSI Disk Drive
hdisk2 Available 01-08-00-5,0 16 Bit LVD SCSI Disk Drive
hdisk3 Available MPIO Other iSCSI Disk Drive
```

```
root> lsattr -E1 hdisk3
```

```
PCM          PCM/friend/iscsiother          Path Control Module          False
algorithm    fail_over                          Algorithm                     True
clr_q        no                                  Device CLEARS its Queue on error
True
dist_err_pcmt 0                                  Distributed Error Percentage  True
dist_tw_width 50                                  Distributed Error Sample Time True
hcheck_cmd    test_unit_rdy                      Health Check Command         True
hcheck_interval 60                                  Health Check Interval        True
hcheck_mode   nonactive                          Health Check Mode            True
host_addr     9.26.51.34                         Hostname or IP Address       False
location      Location Label                      Location Label                True
LUN_id        0x0                                  Logical Unit Number ID       False
max_transfer  0x40000                             Maximum TRANSFER Size        True
port_num      0xcbc                               PORT Number                  False
pvid          none                                 Physical volume identifier    False
q_err        yes                                  Use QERR bit                 True
q_type        simple                               Queuing TYPE                 True
queue_depth   8                                    Queue DEPTH                  True
reassign_to   120                                 REASSIGN time out value      True
reserve_policy single_path                          Reserve Policy                True
rw_timeout    30                                  READ/WRITE time out value    True
start_timeout 60                                  START unit time out value    True
target_name   iqn.1992-08.com.netapp:sn.118052563 Target NAME                   False
unique_id     260CHnWNZorKdb9T07FAS307006NETAPPiscsi Unique device identifier      False
```

- Whenever the iSCSI LUN is accessed from the host (when Configuration Manager is running on the host, for example), the host initiators will make a connection to the NetApp filer. You can check for new host initiator connections, by issuing the following command on the NetApp filer:  
root> **iscsi initiator show**

## Configuring NetApp iSCSI LUNs on Linux

We have included the following steps as a guideline to configure iSCSI LUNs on Linux.

1. Get the iSCSI initiator name, or `iscsi-iname`, from both Linux hosts, for example, on the primary server:  
root> `iscsi-iname`  
`iqn.1996-07.com.cisco:01.54c47d5690d3`
2. On the NetApp filer, configure the LUNs using the above initiator names (using the steps similar to what we have done with the AIX hosts). Make sure that you select “Linux” as the LUN protocol when adding the shared LUN.
3. Install iSCSI initiator (`iscsi-initiator-utils rpm`) on the two Linux servers. This will create the necessary binaries and will create `/etc/iscsi.conf` and `/etc/initiatorname.iscsi` configuration files.
4. Add `iscsi-iname` (from Step 1 above) to `/etc/initiatorname.iscsi` configuration file.
5. Update the `/etc/iscsi.conf` file with the following information about NetApp filer.  
`Continuous=no`  
`HeaderDigest=never`  
`DataDigest=never`  
`ImmediateData=yes`  
`DiscoveryAddress=9.26.51.34`  
Note: The `DiscoveryAddress` is the IP address of the NetApp filer.
6. Start the iSCSI initiator:  
root> `/etc/init.d/iscsi start`  
Checking iscsi config: [ OK ]  
Loading iscsi driver: [ OK ]  
Starting iscsid: [ OK ]
7. Set iSCSI initiator to start automatically after reboot:  
root> `chkconfig iscsi on`
8. Check whether the iSCSI LUN shows up on the Linux host:  
root> `iscsi-ls`
9. Now you should have a new device in your Linux host (e.g., `/dev/sdc`) - that is your iSCSI tiebreaker device.
10. Repeat the same process on the secondary node.

## Appendix F - Testing three-node DB2 HA cluster topology

---

We conducted eight test scenarios covering everything from planned outages to application failures (DB2 crash) to unplanned/unexpected failures.

### Prerequisites:

- Configure EtherChannel on all three nodes/AIX hosts.
- Install and configure TSA 2.20.7.
- Create a TSA/R SCT two-node cluster with HADR nodes and then add the third node (Arbitrator) to the same cluster.

### Test scenarios

#### Notes:

- Node failures were simulated using both graceful reboot (`shutdown -Fr`), quick reboot (`reboot -q`) and LPAR shutdown from the HMC.
- For each scenario we ran our stress client from another server located in Toronto Lab. The stress client basically ran transactions against the primary node. Since we had configured ACR, we just observed whether the application had switched over to the standby automatically.
- Two monitor scripts, one to monitor the health of the instance (`db2V95_monitor.ksh`) and second to monitor HADR database/resources group status (`hadrV95_monitor.ksh`) are continuously running every 10 seconds on each node. An example of a monitor script run from the SYSLOGs on one of the nodes follows:

```
Aug 13 16:58:39 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/db2V95_monitor.ksh[217314]: Returning 1 (db2inst1, 0)
Aug 13 16:58:39 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/hadrV95_monitor.ksh[122940]: Returning 2 : db2inst1
db2inst1 HADB
```

#### 1. Graceful “role-switch” or planned outage:

- Start with both nodes in peer mode.
- Issue `takeover hadr on db <db name> on the standby`
  - Note that TSA will issue the following HA Policy scripts:
    - **Primary Node:** `/usr/sbin/rsct/sapolicies/db2/hadrV95_stop.ksh`
    - **Standby Node:** `/usr/sbin/rsct/sapolicies/db2/hadrV95_start.ksh`
- Standby becomes new primary, and the old primary becomes new standby
- Since there was no node failure, the Arbitrator node is not considered by cluster manager for this scenario.

#### 2. Forced "role-switch":

- Nodes are not in peer mode. For example, issue `db2_kill` or `kill -9 <PID of db2sysc>` on primary and make sure they are in "DisconnectedPeer" state.
- Issue `takeover hadr on db <db name> by force on the standby`.

- TSA reintegrates the old primary as the new standby the next time the Monitor script runs.
  - Note that TSA will issue the following HA Policy scripts:
    - **Old Primary/New Standby:** /usr/sbin/rsct/sapolicies/db2/db2V95\_start.ksh
    - **Old Standby/New Primary:** /usr/sbin/rsct/sapolicies/db2/hadrV95\_start.ksh
- Since there is no node failure, the Arbitrator node is not considered by cluster manager for this scenario.
- Note:** since TSA first does a local resource recovery, it is possible to have a scenario where the old primary instance is activated before "take over by force" is completed. In this case, to avoid a split-brain scenario, TSA will kill the DB2 instance on the old primary - i.e., issue the policy script: /usr/sbin/rsct/sapolicies/db2/hadrV95\_stop.ksh

### 3. Forced "role-switch" using PEER WINDOW ONLY:

- Start with both nodes in peer mode.
- Always ensure that the clocks on the HADR nodes are synchronized!
- Issue takeover hadr on db <db name> by force PEER WINDOW ONLY on the standby.
  - Note that TSA will issue the following HA Policy scripts:
    - **Primary Node:** /usr/sbin/rsct/sapolicies/db2/hadrV95\_stop.ksh
    - **Standby Node:** /usr/sbin/rsct/sapolicies/db2/hadrV95\_start.ksh
- Standby becomes new primary, and the old primary becomes new standby
- Since there was no node failure, the Arbitrator node is not considered by cluster manager for this scenario.

### 4. Primary node failure:

- Start with both nodes in peer mode.
- Always ensure that the clocks on the HADR nodes are synchronized!
- Primary is rebooted. At this point the primary node will lose the Master node Status in the cluster (if it was indeed the Master at this point). On standby you will see the following as TSA resource status (**lssam**) :

```
alaska:db2inst1:/home/db2inst1>lssam
Online IBM.ResourceGroup:db2_db2inst1_alaska_0-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_alaska_0-rs
  '- Online IBM.Application:db2_db2inst1_alaska_0-rs:alaska
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADB-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs
  '- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs:alaska
'- Failed_offline IBM.Application:db2_db2inst1_db2inst1_HADB-rs:harsys1 Node=Offline
Failed_offline IBM.ResourceGroup:db2_db2inst1_harsys1_0-rg Nominal=Online
'- Failed_offline IBM.Application:db2_db2inst1_harsys1_0-rs
  '- Failed_offline IBM.Application:db2_db2inst1_harsys1_0-rs:harsys1 Node=Offline
```

- SYSLOGs on the standby and Arbitrator nodes indicates that a member has left the cluster and master node has changed, i.e., The master node now is the standby node:

```
Aug 7 11:30:48 hasys3 daemon:info RecoveryRM[335982]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6n41170cW2T8/a/00k.X47.....:Reference ID: :::Template
ID: 890f11b3:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,2571
:::RECOVERYRM_INFO_4_ST A member has left. Node number = 1
Aug 7 11:30:48 hasys3 daemon:info RecoveryRM[335982]: (Recorded using libct_ffdc.a cv
2)::Error ID: 61cy9F.cW2T8/pT00k.X47.....:Reference ID: :::Template
```

```
ID: 112fea03::Details File:  ::Location: RSCT,Protocol.C,1.54.1.22,2605
::RECOVERYRM_INFO_6_ST Master has left, a new master has taken over. Node number of the
new master = 2
```

•Note that TSA will issue the following HA Policy scripts:

- **Standby node:** /usr/sbin/rsct/sapolicies/db2/hadrV95\_start.ksh
- **Primary node:** /usr/sbin/rsct/sapolicies/db2/db2V95\_start.ksh

•Standby becomes new primary, and the old primary becomes new standby after coming back online and reintegrated into the cluster. Sample cluster status output from standby node:

```
alaska:db2inst1:/home/db2inst1>lssam
Online IBM.ResourceGroup:db2_db2inst1_alaska_0-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_alaska_0-rs
   '- Online IBM.Application:db2_db2inst1_alaska_0-rs:alaska
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADB-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs
   '- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs:alaska
   '- Offline IBM.Application:db2_db2inst1_db2inst1_HADB-rs:harsys1
Online IBM.ResourceGroup:db2_db2inst1_harsys1_0-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_harsys1_0-rs
   '- Online IBM.Application:db2_db2inst1_harsys1_0-rs:harsys1
```

•The new standby (old primary) is now part of the cluster and quorum state is achieved. SYSLOG output on the new primary after coming back online:

```
Aug 7 08:36:33 harsys1 daemon:notice ConfigRM[389322]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 4bddfbcc::Details File:  ::Location:
RSCT,PeerDomain.C,1.99.1.311,16907      ::CONFIGRM_HASQUORUM_ST The operational
quorum state of the active peer domain has changed to HAS_QUORUM. In this state, cluster
resources may be recovered and controlled as needed by management applications.
Aug 7 08:36:33 harsys1 daemon:notice ConfigRM[389322]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 3b16518d::Details File:  ::Location:
RSCT,PeerDomain.C,1.99.1.311,12054      ::CONFIGRM_ONLINE_ST The node is online in
the domain indicated in the detail data. Peer Domain Name db2HADomain
Aug 7 08:36:34 harsys1 daemon:notice StorageRM[381096]: (Recorded using libct_ffdc.a cv
2)::ErrorID: ::Reference ID:  ::Template ID: edff8e9b::Details File:  ::Location:
RSCT,IBM.StorageRmD.C,1.41,142        ::STORAGERM_STARTED_ST IBM.StorageRM
daemon has started.
Aug 7 08:36:35 harsys1 daemon:notice RecoveryRM[368828]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6cqj1q01c2T8/1Ze0KY097.....::Reference ID:  ::Template
ID: b60efda8::Details File:  ::Location: RSCT,IBM.RecoveryRmD.C,1.21.2.1,136
::RECOVERYRM_INFO_0_ST IBM.RecoveryRM daemon has started.
Aug 7 08:36:39 harsys1 daemon:notice RecoveryRM[368828]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6bGpGm/5c2T8/WdC.KY097.....::Reference ID:  ::Template
ID: 724b54a7::Details File:  ::Location: RSCT,Protocol.C,1.54.1.22,380
::RECOVERYRM_INFO_7_ST This node has joined the IBM.RecoveryRM group. My node number =
1 ; Master node number = 2
```

## 5. Standby node failure:

- Start with both nodes in peer mode.
- Standby is rebooted and primary is now in "DisconnectedPeer" state. And the cluster status output (lssam) is similar to the following:

```
$ lssam
Offline IBM.ResourceGroup:db2_db2inst1_alaska_0-rg Nominal=Online
'- Failed offline IBM.Application:db2_db2inst1_alaska_0-rs
   '- Failed offline IBM.Application:db2_db2inst1_alaska_0-rs:alaska
Offline IBM.ResourceGroup:db2_db2inst1_db2inst1_HADB-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs
   '- Failed offline IBM.Application:db2_db2inst1_db2inst1_HADB-rs:alaska
   '- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs:harsys1
Offline IBM.ResourceGroup:db2_db2inst1_harsys1_0-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_harsys1_0-rs
   '- Online IBM.Application:db2_db2inst1_harsys1_0-rs:harsys1
```

•Note that TSA will issue the following HA Policy scripts:

- **Standby node:** /usr/sbin/rsct/sapolicies/db2/db2V95\_start.ksh

•Standby will re-join the cluster after reboot and no role switching is required.

- Arbitrator node detects the standby node failure and subsequent re-join into the cluster.

Sample SYSLOG output from Arbitrator node:

```
Aug 7 15:45:31 hasys3 daemon:info RecoveryRM[335982]: (Recorded using libct_ffdc.a cv
2):::Error ID: 6n41170PF6T8/AXT/k.X47.....:::Reference ID: :::Template
ID: 890f11b3:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,2571
:::RECOVERYRM_INFO_4_ST A member has left. Node number = 2
Aug 7 15:45:31 hasys3 daemon:info RecoveryRM[335982]: (Recorded using libct_ffdc.a cv
2):::Error ID: 64LGh0/PF6T8//U/k.X47.....:::Reference ID: :::Template
ID: 42b525c6:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,2599
:::RECOVERYRM_INFO_5_ST Master has left, this node is now the master.
Aug 7 15:45:40 hasys3 daemon:notice RecoveryRM[335984]: (Recorded using libct_ffdc.a cv
2):::ErrorID: 6cqj1q0YF6T8/H.Llk.X47.....:::Reference ID: :::Template ID:
b60efda8:::Details File: :::Location: RSCT,IBM.RecoveryRMd.C,1.21.2.1,136
:::RECOVERYRM_INFO_0_ST IBM.RecoveryRM daemon has started.
Aug 7 15:45:41 hasys3 daemon:notice RecoveryRM[335984]: (Recorded using libct_ffdc.a cv
2):::ErrorID: 6bGpGm/ZF6T8//eb.k.X47.....:::Reference ID: :::Template ID:
724b54a7:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,380
:::RECOVERYRM_INFO_7_ST This node has joined the IBM.RecoveryRM group. My node number =
3 ; Master node number = 1
```

## 6. Unexpected primary node failure - hardware/power/single-site failure

- Start with both nodes in peer mode.
- Using HMC turn off the power to the primary node. From the HMC shutdown menu we used the "Immediate" option, which will shut down the logical partition (LPAR) as quickly as possible, without notifying the logical partitions.
- Always ensure that the clocks on the HADR nodes are synchronized!
- From Arbitrator SYSLOGS we see that primary node has left the cluster and that Master node has switched-over:

```
Aug 13 17:24:20 hasys3 daemon:info RecoveryRM[327888]: (Recorded using libct_ffdc.a cv
2):::Error ID: 6n411702G6V8/Zq11k.X47.....:::Reference ID: :::Template
ID: 890f11b3:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,2571
:::RECOVERYRM_INFO_4_ST A member has left. Node number = 2
Aug 13 17:24:20 hasys3 daemon:info RecoveryRM[327888]: (Recorded using libct_ffdc.a cv
2):::Error ID: 61cy9F.2G6V8/RHmlk.X47.....:::Reference ID: :::Template
ID: 112fea03:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,2605
:::RECOVERYRM_INFO_6_ST Master has left, a new master has taken over. Node number of the
new master = 1
```

- At this point the old standby will do a TAKEOVER BY FORCE PEER WINDOW ONLY and become new primary. However, since the old primary is still down, the new primary will be in "Disconnected" state.

```
$ db2pd -hadr -db hadb
Database Partition 0 -- Database HADB -- Active -- Up 1 days 03:04:44
HADR Information:
Role      State           SyncMode HeartBeatsMissed  LogGapRunAvg (bytes)
Primary  Disconnected   Nearsync  0                  0
ConnectStatus ConnectTime      Timeout
Disconnected Thu Aug 13 14:24:05 2009 (1250198645) 180
PeerWindowEnd PeerWindow
Null (0)      180
LocalHost     LocalService
harsys1.beaverton.ibm.com 50001
RemoteHost    RemoteService    RemoteInstance
alaska.yamato.ibm.com    50002            db2inst1
PrimaryFile   PrimaryPg   PrimaryLSN
S0000021.LOG 463        0x00000000006D57925
StandByFile   StandByPg   StandByLSN
S0000000.LOG 0          0x0000000000000000
```

- On new primary (old standby) you will see the following as TSA resource status (`lssam`):

```
# lssam
FailedOffline IBM.ResourceGroup:db2_db2inst1_alaska_0-rg Control=StartInhibited Nominal=Online
- FailedOffline IBM.Application:db2_db2inst1_alaska_0-rs
- FailedOffline IBM.Application:db2_db2inst1_alaska_0-rs:alaska Node=Offline
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADB-rg Request=NOQ Nominal=Online
- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs
- FailedOffline IBM.Application:db2_db2inst1_db2inst1_HADB-rs:alaska Node=Offline
- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs:harsys1
Online IBM.ResourceGroup:db2_db2inst1_harsys1_0-rg Nominal=Online
- Online IBM.Application:db2_db2inst1_harsys1_0-rs
- Online IBM.Application:db2_db2inst1_harsys1_0-rs:harsys1
```

**Note:** Since HADR pair is not in peer mode, the cluster resource for HADR database "db2\_db2inst1\_db2inst1\_HADB-rg" will be locked.

- The old primary is brought online after ~30 minutes delay and will be reintegrated into the cluster as standby. SYSLOGs from the new standby:

```
Aug 13 16:52:26 alaska daemon:notice ConfigRM[204994]: (Recorded using libct_ffdc.a cv
2):::Error ID: ::Reference ID: ::Template ID: 4bddfbcc:::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,16907 ::CONFIGRM_HASQUORUM_ST The operational
quorum state of the active peer domain has changed to HAS_QUORUM. In this state, cluster
resources may be recovered and controlled as needed by management applications.
Aug 13 16:52:26 alaska daemon:notice ConfigRM[204994]: (Recorded using libct_ffdc.a cv
2):::Error ID: ::Reference ID: ::Template ID: 3b16518d:::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,12054 ::CONFIGRM_ONLINE_ST The node is online in
the domain indicated in the detaildata. Peer Domain Name db2HADomain
Aug 13 16:52:28 alaska daemon:notice RecoveryRM[155954]: (Recorded using libct_ffdc.a cv
2):::ErrorID: 6c9j1q0Qg6V8/Id0sn/F7.....:Reference ID: ::Template ID:
b60efda8:::Details File: ::Location: RSCT,IBM.RecoveryRmD.C,1.21.2.1,136
::RECOVERYRM_INFO_0_ST IBM.RecoveryRM daemon has started.
Aug 13 16:52:29 alaska daemon:notice StorageRM[184508]: (Recorded using libct_ffdc.a cv
2):::Error ID: ::Reference ID: ::Template ID: edff8e9b:::Details File: ::Location:
RSCT,IBM.StorageRmD.C,1.41,142 ::STORAGERM_STARTED_ST IBM.StorageRM
daemon has started.
Aug 13 16:52:34 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/db2V95_monitor.ksh[172284]: Returning 2 (db2inst1, 0)
Aug 13 16:52:34 alaska daemon:notice RecoveryRM[155954]: (Recorded using libct_ffdc.a cv
2):::ErrorID: 6bGpGm/Wg6V8/rqa/sn/F7.....:Reference ID: ::Template ID:
724b54a7:::Details File: ::Location: RSCT,Protocol.C,1.54.1.22,380
::RECOVERYRM_INFO_7_ST This node has joined the IBM.RecoveryRM group. My node number =
2 ; Master node number = 1
Aug 13 16:52:34 alaska user:notice
/usr/sbin/rsct/sapolicies/db2/hadrV95_monitor.ksh[192664]: Begin forcing applications
connected to HADB
Aug 13 16:52:34 alaska auth|security:notice su: from root to db2inst1 at /dev/tty??
Aug 13 16:52:35 alaska user:notice db2V95_start.ksh[180698]: Entered
/usr/sbin/rsct/sapolicies/db2/db2V95_start.ksh, db2inst1, 0
Aug 13 16:52:35 alaska user:debug db2V95_start.ksh[176372]: Able to cd to
/home/db2inst1/sqllib : /usr/sbin/rsct/sapolicies/db2/db2V95_start.ksh, db2inst1, 0
Aug 13 16:52:35 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/db2V95_monitor.ksh[176380]: Returning 2 (db2inst1, 0)
Aug 13 16:52:35 alaska user:debug db2V95_start.ksh[208922]: 1 partitions total:
/usr/sbin/rsct/sapolicies/db2/db2V95_start.ksh, db2inst1, 0
Aug 13 16:52:36 alaska daemon:warn|warning rshd[172056]: Failed rsh authentication from
alaska for local user db2inst1 via remote user db2inst1
Aug 13 16:52:36 alaska auth|security:notice su: from root to db2inst1 at /dev/tty??
Aug 13 16:52:36 alaska user:notice
/usr/sbin/rsct/sapolicies/db2/hadrV95_monitor.ksh[229610]: Done forcing applications
connected to HADB
Aug 13 16:52:36 alaska auth|security:notice su: from root to db2inst1 at /dev/tty??
Aug 13 16:52:36 alaska auth|security:notice su: from root to db2inst1 at /dev/tty??
Aug 13 16:52:40 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/hadrV95_monitor.ksh[192794]: Returning 2 : db2inst1
db2inst1 HADB
Aug 13 16:52:44 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/db2V95_monitor.ksh[241906]: Returning 1 (db2inst1, 0)
```

- Note that TSA will issue the following HA Policy scripts:
  - **New Primary Node:** /usr/sbin/rsct/sapolicies/db2/hadrV95\_start.ksh
  - **New Standby Node (coming online after ~30 minutes):**  
/usr/sbin/rsct/sapolicies/db2/db2V95\_start.ksh

## 7. Network interface card (NIC) failures to test EtherChannel:

- Once EtherChannel is configured, you will not be able to use `ifconfig` or any other command to simulate failure of individual bonded adapters.
- Note that as far as cluster manager is concerned the EtherChannel adapter is like any other standard Ethernet adapter. Hence, individual members of the EtherChannel /Link-Aggregation are not visible to cluster manager or TSA, and thus are not controllable or monitored. Therefore, SYSLOGs/RSCT logs are not useful to determine whether EtherChannel has failed-over to one of the backup adapters.
- First figure out which Ethernet adapter is defined as the "Primary Channel" and which one is the backup:

```
root> lsattr -El ent2 | awk '$1 ~/adapter/'
adapter_names    ent0             EtherChannel Adapters                True
backup_adapter   ent1             Adapter used when whole channel fails  True
```

- Next find out which adapter is currently used by EtherChannel as the "Active Adapter":

```
root> netstat -v | awk '/Active channel/'
Active channel: primary channel
```

- Now we are ready to do failover testing. Two methods to do EtherChannel failover testing:
  - Use AIX EtherChannel Management commands to "force a failover". Issue a command similar to the following to do this (or using `smitty etherchannel -> Force A Failover In An EtherChannel / Link Aggregation`)

```
root> /usr/lib/methods/ethchan_config -f 'ent2'
```

- OR disconnect the Ethernet cable from "Currently Active Adapter"

- After executing the failover, check which adapter is now acting as the "Active channel":

```
root> netstat -v | awk '/Active channel/'
Active channel: backup adapter          <== this indicates we have failed-over successfully
```

## 8. Unexpected failure of arbitrator node /failure of third site:

- All nodes online in the cluster, and two HADR nodes in peer mode at the start
- Using HMC turn off the power to the Arbitrator node. From the HMC shutdown menu we use the "Immediate" option, which will shut down the logical partition (LPAR) as quickly as possible, without notifying the logical partitions.
- SYSLOGs on HADR nodes will indicate the failure of the Arbitrator node. Sample SYSLOGs from the primary node in our cluster:

```
Aug 13 13:42:06 harsysl daemon:info RecoveryRM[446686]: (Recorded using libct_ffdc.a cv
2):::Error ID: 6n41170Se5V8/PSx/KYo97.....:::Reference ID: :::Template
ID: 890f11b3:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,2571
:::RECOVERYRM_INFO_4_ST A member has left. Node number= 3
Aug 13 13:42:06 harsysl daemon:info RecoveryRM[446686]: (Recorded using libct_ffdc.a cv
2):::Error ID: 64LGh0/Se5V8/Nhx/KYo97.....:::Reference ID: :::Template
ID: 42b525c6:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,2599
:::RECOVERYRM_INFO_5_ST Master has left, this node is now the master.
```

- We do a "role-switch" while the Arbitrator is down - e.g., on standby issue `db2 takeover hadr on db hadb`

- The Arbitrator node is brought back online. The SYSLOGs on HADR nodes will indicate that the Arbitrator node has re-joined the cluster. Sample SYSLOGs from the primary node in our cluster indicate:

```
Aug 13 14:06:33 harsysl daemon:info RecoveryRM[446688]: (Recorded using libct_ffdc.a cv
2):::Error ID: 6GNPKt/N/6V8/x0D0KYo97.....:::Reference ID: :::Template
```

```
ID: 7959b652:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,2559  
:::RECOVERYRM_INFO_3_ST A new member has joined. Node number = 3
```

- SYSLOGs from the Arbitrator node indicate that "quorum" state of the cluster has been established and the node is now part of the domain "db2HADomain":

```
Aug 13 17:06:55 hasys3 daemon:notice ConfigRM[225514]: (Recorded using libct_ffdc.a cv  
2):::Error ID: :::Reference ID: :::Template ID: 4bddfbcc:::Details File: :::Location:  
RSCT,PeerDomain.C,1.99.1.311,16907          :::CONFIGRM_HASQUORUM_ST The operational  
quorum state of the active peer domain has changed to HAS_QUORUM. In this state, cluster  
resources may be recovered and controlled as needed by management applications.  
Aug 13 17:06:55 hasys3 daemon:notice ConfigRM[225514]: (Recorded using libct_ffdc.a cv  
2):::Error ID: :::Reference ID: :::Template ID: 3b16518d:::Details File: :::Location:  
RSCT,PeerDomain.C,1.99.1.311,12054          :::CONFIGRM_ONLINE_ST The node is online in  
the domain indicated in the detail data. Peer Domain Name db2HADomain  
Aug 13 17:06:58 hasys3 daemon:notice RecoveryRM[327888]: (Recorded using libct_ffdc.a cv  
2):::ErrorID: 6c9j1q0m/6V8/jP0.k.X47.....:::Reference ID: :::Template ID:  
b60efda8:::Details File: :::Location: RSCT,IBM.RecoveryRMd.C,1.21.2.1,136  
:::RECOVERYRM_INFO_0_ST IBM.RecoveryRM daemon has started.  
Aug 13 17:06:58 hasys3 daemon:notice StorageRM[356442]: (Recorded using libct_ffdc.a cv  
2):::Error ID: :::Reference ID: :::Template ID: edff8e9b:::Details File: :::Location:  
RSCT,IBM.StorageRMd.C,1.41,142          :::STORAGE_RM_STARTED_ST IBM.StorageRM  
daemon has started.  
Aug 13 17:07:02 hasys3 daemon:notice RecoveryRM[327888]: (Recorded using libct_ffdc.a cv  
2):::ErrorID: 6bGpGm/q/6V8/3cG/k.X47.....:::Reference ID: :::Template ID:  
724b54a7:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,380  
:::RECOVERYRM_INFO_7_ST This node has joined the IBM.RecoveryRM group. My node number =  
3; Master node number = 2
```

- Note that if there is any node failure at this time, the cluster will be brought offline, since at this point none of the nodes will be able to exercise the tiebreaker mechanism and will not be able to establish "HAS QUORUM" state.

## Appendix G - Testing two-node DB2 HA cluster with shared DISK tiebreaker

---

We conducted eight test scenarios covering everything from planned outages to application failures (DB2 crash) to unplanned/unexpected failures.

### Prerequisites:

- Configure EtherChannel on both HADR nodes/AIX hosts.
- Install and configure TSA 2.2.0.7.
- Create a TSA/RSCT two-node cluster with the two HADR nodes located on two physically separated sites.

### Test scenarios

#### Notes:

- Node failures were simulated using both graceful reboot (`shutdown -Fr`), quick reboot (`reboot -q`), and LPAR shutdown from HMC.
- For each scenario we ran our stress client from another server located in Toronto Lab. The stress client basically ran transactions against the primary node. Since we had configured ACR, we just observed whether the application had switched over to the standby automatically.
- Two monitor scripts, one to monitor the health of the instance (`db2V95_monitor.ksh`) and second to monitor HADR database/resources group status (`hadrV95_monitor.ksh`) are continuously running every 10 seconds on each node. Sample monitor script run from SYSLOGs on one of the nodes:

```
Aug 13 16:58:39 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/db2V95_monitor.ksh[217314]: Returning 1 (db2inst1, 0)
Aug 13 16:58:39 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/hadrV95_monitor.ksh[122940]: Returning 2 : db2inst1
db2inst1 HADB
```

#### 1. Graceful “role-switch” or planned outage:

- Start with both nodes in peer mode.
- Issue `takeover hadr on db <db name>` on the standby.
  - Note that TSA will issue the following HA Policy scripts:
    - **Primary Node:** `/usr/sbin/rsct/sapolicies/db2/hadrV95_stop.ksh`
    - **Standby Node:** `/usr/sbin/rsct/sapolicies/db2/hadrV95_start.ksh`
- Standby becomes new primary, and the old primary becomes new standby.
- Since there was no node failure, the tiebreaker (TB) is not exercised. As long as both nodes remain alive, RSCT will *not* put a lock on the disk using the TB – i.e., the TB is only called into action when RSCT sees the other node as unreachable.

#### 2. Forced "role-switch":

- Nodes are not in peer mode. For example, issue `db2_kill` or `kill -9 <PID of db2sysc>` on the primary and make sure they are in "DisconnectedPeer" state.

DB2 system topology and configuration for automated multi-site HA and DR

- Issue `takeover hadr on db <db name> by force on the standby.`
- TSA reintegrates the old primary as the new standby the next time the Monitor script runs.
  - Note that TSA will issue the following HA Policy scripts:
    - **Old Primary/New Standby:** `/usr/sbin/rsct/sapolicies/db2/db2V95_start.ksh`
    - **Old Standby/New Primary:** `/usr/sbin/rsct/sapolicies/db2/hadrV95_start.ksh`
- Since there was no node failure, the tiebreaker is not exercised.
- **Note:** since TSA first does a local resource recovery, it is possible to have a scenario where the old primary instance is activated before "take over by force" is completed. In this case, to avoid a split-brain scenario, TSA will kill the DB2 instance on the old primary - i.e., issue the policy script: `/usr/sbin/rsct/sapolicies/db2/hadrV95_stop.ksh`

### 3. Forced "role-switch" using PEER WINDOW ONLY:

- Start with both nodes in peer mode.
- Always ensure that the clocks on the HADR nodes are synchronized!
- Issue `takeover hadr on db <db name> by force PEER WINDOW ONLY on the standby.`
  - Note that TSA will issue the following HA Policy scripts:
    - **Primary Node:** `/usr/sbin/rsct/sapolicies/db2/hadrV95_stop.ksh`
    - **Standby Node:** `/usr/sbin/rsct/sapolicies/db2/hadrV95_start.ksh`
- Standby becomes new primary, and the old primary becomes new standby
- Since there was no node failure, the tiebreaker is not exercised.

### 4. Primary node failure:

- Start with both nodes in peer mode.
- Always ensure that the clocks on the HADR nodes are synchronized!
- Primary is rebooted.
  - Note that TSA will issue the following HA Policy scripts:
    - **Standby node:** `/usr/sbin/rsct/sapolicies/db2/hadrV95_start.ksh`
    - **Primary node:** `/usr/sbin/rsct/sapolicies/db2/db2V95_start.ksh`
- On the standby, SYSLOGs will indicate loss of quorum when the standby sees the primary as down, and then standby will reserve the disk TB in order for the node to be in quorum again. Hence the subsequent SYSLOG entries show that the "operational quorum" is established and the standby node is promoted to the master node status:

```
Aug 27 15:56:24 alaska daemon:err|error ConfigRM[204994]: (Recorded using libct_ffdc.a cv
2):::Error ID: :::Reference ID: :::Template ID: a098bf90:::Details File: :::Location:
RSCT,PeerDomain.C,1.99.1.311,16911      :::CONFIGRM_PENDINGQUORUM_ER The
operational quorum state of the active peer domain has changed to PENDING_QUORUM. This
state usually indicates that exactly half of the nodes that are defined in the peer
domain are online. In this state cluster resources cannot be recovered although none
will be stopped explicitly.
Aug 27 15:56:24 alaska daemon:info RecoveryRM[192556]: (Recorded using libct_ffdc.a cv
2):::Error ID: 6n41170s9jz8/5iw/sn/F7.....:::Reference ID: :::Template
ID: 890f11b3:::Details File: :::Location: RSCT,Protocol.C,1.54.1.22,2571
:::RECOVERYRM_INFO_4_ST A member has left. Node number = 1
```

```
Aug 27 15:56:24 alaska daemon:info RecoveryRM[192556]: (Recorded using libct_ffdc.a cv
2)::Error ID: 64LGh0/s9jz8/tw/sn/F7.....:Reference ID: ::Template
ID: 42b525c6::Details File: ::Location: RSCT,Protocol.C,1.54.1.22,2599
::RECOVERYRM_INFO_5_ST Master has left, this node is now the master.
Aug 27 15:56:27 alaska daemon:notice ConfigRM[204994]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 4bddfbcc::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,16907      ::CONFIGRM_HASQUORUM_ST The operational
quorum state of the active peer domain has changed to HAS_QUORUM. In this state, cluster
resources may be recovered and controlled as needed by management applications.
```

- Standby becomes new primary, and the old primary becomes new standby after coming back online and getting reintegrated into the cluster. Sample SYSLOG output from primary /old standby:

```
Aug 27 16:02:31 alaska daemon:info RecoveryRM[192556]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6GNPKt/bFjz8/Lpflsn/F7.....:Reference ID: ::Template
ID: 7959b652::Details File: ::Location: RSCT,Protocol.C,1.54.1.22,2559
::RECOVERYRM_INFO_3_ST A new member has joined. Node number = 1
```

- Sample SYSLOG output from standby /old primary indicates the reintegration phase after coming back online:

```
Aug 27 08:46:10 harsysl daemon:notice cthats[438360]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6UpNEL00deZ8//CY/KYo97.....:Reference ID: ::Template
ID: 97419d60::Details File: ::Location: rsct,bootstrp.C,1.215,4477
::TS_START_ST Topology Services daemon started Topology Services daemon started by: SRC
Topology Services daemon log file location
/var/ct/db2HADomain/log/cthats/cthats.27.0846/var/ct/db2HADomain/run/cthats/ Topology
Services daemon run directory /var/ct/db2HADomain/run/cthats/
Aug 27 08:46:11 harsysl daemon:notice cthags[331980]: (Recorded using libct_ffdc.a cv
2)::Error ID: 63Y7ej0ldeZ8/WPm.KYo97.....:Reference ID: ::Template
ID: afa89905::Details File: ::Location: RSCT,pgsd.C,1.62.1.9,612
::GS_START_ST Group Services daemon started DIAGNOSTIC EXPLANATION HAGS daemon started
by SRC. Log file is /var/ct/db2HADomain/log/cthags/trace_1_0.
Aug 27 08:46:22 harsysl daemon:notice RMCdaemon[253966]: (Recorded using libct_ffdc.a cv
2)::ErrorID: 64rCpW0CdeZ8/9pO/KYo97.....:Reference ID: ::Template ID:
a2d4edc6::Details File: ::Location: RSCT,rmcd.c,1.62,1003
::RMCD_INFO_1_ST The daemon is stopped. Number of command that stopped the daemon 1
Aug 27 08:46:22 harsysl daemon:notice RMCdaemon[253968]: (Recorded using libct_ffdc.a cv
2)::ErrorID: 6eKora0CdeZ8/F/2lKY097.....:Reference ID: ::Template ID:
a6df45aa::Details File: ::Location: RSCT,rmcd.c,1.62,213
::RMCD_INFO_0_ST The daemon is started.
Aug 27 08:46:23 harsysl daemon:notice StorageRM[700568]: (Recorded using libct_ffdc.a cv
2)::ErrorID: ::Reference ID: ::Template ID: edff8e9b::Details File: ::Location:
RSCT,IBM.StorageRMd.C,1.41,142      ::STORAGE_RM_STARTED_ST IBM.StorageRM
daemon has started.
Aug 27 08:46:24 harsysl daemon:notice RecoveryRM[442490]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6cqj1q0EdeZ8/XMA.KYo97.....:Reference ID: ::Template
ID: b60efda8::DetailsFile: ::Location: RSCT,IBM.RecoveryRMd.C,1.21.2.1,136
::RECOVERYRM_INFO_0_ST IBM.RecoveryRM daemon has started.
Aug 27 08:46:25 harsysl daemon:notice ConfigRM[356536]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 4bddfbcc::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,16907      ::CONFIGRM_HASQUORUM_ST The operational
quorum state of the active peer domain has changed to HAS_QUORUM. In this state, cluster
resources may be recovered and controlled as needed by management applications.
Aug 27 08:46:25 harsysl daemon:notice RecoveryRM[442490]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6bGpGm/FdeZ8/K2M.KYo97.....:Reference ID: ::Template
ID: 724b54a7::Details File: ::Location: RSCT,Protocol.C,1.54.1.22,380
::RECOVERYRM_INFO_7_ST This node has joined the IBM.RecoveryRM group. My node number =
1 ; Master node number = 1
Aug 27 08:46:25 harsysl daemon:notice ConfigRM[356536]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 3b16518d::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,12054      ::CONFIGRM_ONLINE_ST The node is online in
the domain indicated in the detail data. Peer Domain Name db2HADomain
Aug 27 08:46:29 harsysl daemon:info RecoveryRM[442490]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6GNPKt/JdeZ8/.94lKY097.....:Reference ID: ::Template
ID: 7959b652::Details File: ::Location: RSCT,Protocol.C,1.54.1.22,2559
::RECOVERYRM_INFO_3_ST A new member has joined. Node number = 2
```

## 5. Standby node failure:

DB2 system topology and configuration for automated multi-site HA and DR

- Start with both nodes in peer mode.
- Standby is rebooted and primary is now in "DisconnectedPeer" state. When the primary sees the standby as down, the primary will reserve the disk TB in order for the node to be in quorum again. And you will see the loss and subsequent reservation of quorum device on primary SYSLOGs:

```
Aug 31 17:39:22 alaska daemon:err|error ConfigRM[139738]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: a098bf90::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,16911      ::CONFIGRM_PENDINGQUORUM_ER The
operational quorum state of the active peer domain has changed to PENDING_QUORUM. This
state usually indicates that exactly half of the nodes that are defined in the peer
domain are online. In this state cluster resources cannot be recovered although none
will be stopped explicitly.
Aug 31 17:39:22 alaska daemon:info RecoveryRM[61680]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6n4l170023b8/o.l.sn/F7.....::Reference ID: ::Template
ID: 890f11b3::Details File: ::Location: RSCT,Protocol.C,1.54.1.22,2571
::RECOVERYRM_INFO_4_ST A member has left. Node number = 1
Aug 31 17:39:22 alaska daemon:info RecoveryRM[61680]: (Recorded using libct_ffdc.a cv
2)::Error ID: 64LGh0/O23b8/REl.sn/F7.....::Reference ID: ::Template
ID: 42b525c6::Details File: ::Location: RSCT,Protocol.C,1.54.1.22,2599
::RECOVERYRM_INFO_5_ST Master has left, this node is now the master.
Aug 31 17:39:25 alaska daemon:notice ConfigRM[139738]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 4bddfbcc::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,16907      ::CONFIGRM_HASQUORUM_ST The operational
quorum state of the active peer domain has changed to HAS_QUORUM. In this state, cluster
resources may be recovered and controlled as needed by management applications.
Aug 31 17:39:27 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/db2V95_monitor.ksh[258260]: Returning 1 (db2inst1, 0)
```

- Note that TSA will issue the following HA Policy scripts:

- **Standby node:** /usr/sbin/rsct/sapolicies/db2/db2V95\_start.ksh

- Standby will re-join the cluster after reboot and no role switching is required. When the node is reintegrated, you will see messages similar to the following in the SYSLOGs (on the standby):

```
Aug 31 15:44:31 harsys1 daemon:notice StorageRM[426216]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: edff8e9b::Details File: ::Location:
RSCT,IBM.StorageRmD.C,1.41,142      ::STORAGERM_STARTED_ST IBM.StorageRM
daemon has started.
Aug 31 15:44:33 harsys1 daemon:notice RecoveryRM[368854]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6cqj1q0F73b8/n5A/KYo97.....::Reference ID: ::Template
ID: b60efda8::Details File: ::Location: RSCT,IBM.RecoveryRmD.C,1.21.2.1,136
::RECOVERYRM_INFO_0_ST IBM.RecoveryRM daemon has started.
Aug 31 15:44:34 harsys1 daemon:err|error automountd[155882]: Duplicate request
Aug 31 15:44:35 harsys1 daemon:notice ConfigRM[463100]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 4bddfbcc::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,16907      ::CONFIGRM_HASQUORUM_ST The operational
quorum state of the active peer domain has changed to HAS_QUORUM. In this state, cluster
resources may be recovered and controlled as needed by management applications.
Aug 31 15:44:39 harsys1 daemon:notice ConfigRM[463100]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 3b16518d::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,12054      ::CONFIGRM_ONLINE_ST The node is online in
the domain indicated in the detail data. Peer Domain Name db2HADomain
Aug 31 15:44:43 harsys1 daemon:notice RecoveryRM[368854]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6bGpGm/P73b8/S4e/KYo97.....::Reference ID: ::Template
ID: 724b54a7::Details File: ::Location: RSCT,Protocol.C,1.54.1.22,380
::RECOVERYRM_INFO_7_ST This node has joined the IBM.RecoveryRM group. My node number =
1 ; Master node number = 2
Aug 31 15:44:44 harsys1 user:debug
/usr/sbin/rsct/sapolicies/db2/db2V95_monitor.ksh[417968]: Returning 2 (db2inst1, 0)
Aug 31 15:44:44 harsys1 auth|security:notice su: from root to db2inst1 at /dev/tty??
Aug 31 15:44:44 harsys1 daemon:err|error automountd[155882]: Duplicate request
Aug 31 15:44:46 harsys1 user:notice db2V95_start.ksh[417978]: Entered
/usr/sbin/rsct/sapolicies/db2/db2V95_start.ksh, db2inst1, 0
```

- You can also observe that standby node is online by examining primary SYSLOGs:

```
Aug 31 17:45:32 alaska daemon:info RecoveryRM[61680]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6GNPKt/A83b8/KwT1sn/F7.....::Reference ID:   ::Template
ID: 7959b652::Details File:   ::Location: RSCT,Protocol.C,1.54.1.22,2559
::RECOVERYRM_INFO_3_ST A new member has joined. Node number = 1
```

## 6. Unexpected primary node failure - hardware/power/single-site failure

- Start with both nodes in peer mode.

- Using HMC turn off the power to the primary node. From the HMC shutdown menu we use the "Immediate" option, which will shutdown the logical partition (LPARs) as quickly as possible, without notifying the logical partitions.

- SYSLOGs (on the standby) indicate loss of quorum when the standby sees the primary as down. Then standby will reserve the disk TB in order for the node to be in quorum again. Hence the subsequent SYSLOG entries show that the “operational quorum” is established and the standby node is promoted to the master node status:

```
Aug 31 11:31:09 harsys1 daemon:err|error ConfigRM[360634]: (Recorded using libct_ffdc.a
cv 2)::Error ID:   ::Reference ID:   ::Template ID: a098bf90::Details File:
::Location: RSCT,PeerDomain.C,1.99.1.311,16911   ::CONFIGRM_PENDINGQUORUM_ER
The operational quorum state of the active peer domain has changed to PENDING_QUORUM.
This state usually indicates that exactly half of the nodes that are defined in the peer
domain are online. In this state cluster resources cannot be recovered although none
will be stopped explicitly.
Aug 31 11:31:09 harsys1 daemon:info RecoveryRM[413732]: (Recorded using libct_ffdc.a cv
2)::Error ID: 6n41170hP/b8/X5V0KY097.....::Reference ID:   ::Template
ID: 890f11b3::Details File:   ::Location: RSCT,Protocol.C,1.54.1.22,2571
::RECOVERYRM_INFO_4_ST A member has left. Node number = 2
Aug 31 11:31:09 harsys1 daemon:info RecoveryRM[413732]: (Recorded using libct_ffdc.a cv
2)::Error ID: 64LGh0/hP/b8/BKV0KY097.....::Reference ID:   ::Template
ID: 42b525c6::Details File:   ::Location: RSCT,Protocol.C,1.54.1.22,2599
::RECOVERYRM_INFO_5_ST Master has left, this node is now the master.
Aug 31 11:31:10 harsys1 auth|security:notice su: from root to db2inst1 at /dev/tty??
Aug 31 11:31:11 harsys1 daemon:notice ConfigRM[360634]: (Recorded using libct_ffdc.a cv
2)::Error ID:   ::Reference ID:   ::Template ID: 4bddfbcc::Details File:   ::Location:
RSCT,PeerDomain.C,1.99.1.311,16907   ::CONFIGRM_HASQUORUM_ST The operational
quorum state of the active peer domain has changed to HAS_QUORUM. In this state, cluster
resources may be recovered and controlled as needed by management applications.
Aug 31 11:31:12 harsys1 user:debug
/usr/sbin/rsct/sapolicies/db2/hadrV95_monitor.ksh[663628]: Returning 2 : db2inst1
db2inst1 HADB
Aug 31 11:31:18 harsys1 user:notice
/usr/sbin/rsct/sapolicies/db2/hadrV95_start.ksh[729298]: Entering : db2inst1 db2inst1 HADB
```

- Note that TSA will issue the following HA Policy scripts:

- **New Primary Node:** /usr/sbin/rsct/sapolicies/db2/hadrV95\_start.ksh

- **New Standby Node (coming online after ~30 minutes):**

- /usr/sbin/rsct/sapolicies/db2/db2V95\_start.ksh

- At this point the old standby will do a TAKEOVER BY FORCE PEER WINDOW ONLY and become the new primary. However, since the old primary is still down, the new primary will be in "Disconnected" state.

```

$ db2pd -hadr -db hadb
Database Partition 0 -- Database HADB -- Active -- Up 3 days 21:34:42
HADR Information:
Role      State              SyncMode HeartBeatsMissed   LogGapRunAvg (bytes)
Primary  Disconnected        Nearsync  0                    0
ConnectStatus ConnectTime              Timeout
Disconnected Mon Aug 31 11:31:23 2009 <1251743483> 180
PeerWindowEnd PeerWindow
Null (<0>)    180
LocalHost      LocalService
harsys1.beaverton.ibm.com 50001
RemoteHost     RemoteService RemoteInstance
alaska.yamato.ibm.com    50002 db2inst1
PrimaryFile PrimaryPg PrimaryLSN
S0000207.LOG 772 0x000000003568C48F
StandByFile StandByPg StandByLSN
S0000000.LOG 0 0x0000000000000000

```

- On new primary (old standby) you will see the following as TSA resource status (**lssam**):

```

# lssam
Failed offline IBM.ResourceGroup:db2_db2inst1_alaska_0-rg Control=StartInhibited Nominal=Online
'- Failed offline IBM.Application:db2_db2inst1_alaska_0-rs
'- Failed offline IBM.Application:db2_db2inst1_alaska_0-rs:alaska Node=Offline
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADB-rg Request=lock Nominal=Online
'- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs
'- Failed offline IBM.Application:db2_db2inst1_db2inst1_HADB-rs:alaska Node=Offline
'- Online IBM.Application:db2_db2inst1_db2inst1_HADB-rs:harsys1
Online IBM.ResourceGroup:db2_db2inst1_harsys1_0-rg Nominal=Online
'- Online IBM.Application:db2_db2inst1_harsys1_0-rs
'- Online IBM.Application:db2_db2inst1_harsys1_0-rs:harsys1

```

**Note:** Since HADR pair is not in peer mode, the cluster resource for HADR database "db2\_db2inst1\_db2inst1\_HADB-rg" will be locked.

- The old primary is brought online after ~30 minutes delay and will be reintegrated into the cluster as standby. Sample SYSLOG from the new standby:

```

Aug 31 14:16:22 alaska daemon:notice RMCdaemon[65946]: (Recorded using libct_ffdc.a cv
2):::Error ID: 6eKora0440b8/LZN0sn/F7.....:::Reference ID: :::Template
ID: a6df45aa:::Details File: :::Location: RSCT,rmcd.c,1.62,213
:::RMCINFO_0_ST The daemon is started.
Aug 31 14:16:28 alaska daemon:notice ConfigRM[160226]: (Recorded using libct_ffdc.a cv
2):::Error ID: :::Reference ID: :::Template ID: de84c4db:::Details File: :::Location:
RSCT,IBM.ConfigRMD.C,1.44,288
:::CONFIGRM_STARTED_ST IBM.ConfigRM daemon
has started.
Aug 31 14:16:31 alaska daemon:notice ctcasd[139556]: (Recorded using libct_ffdc.a cv
2):::Error ID: 6YzeY.1D40b8/wGmlsn/F7.....:::Reference ID: :::Template ID:
c092afe4:::Details File: :::Location: rsct.core.sec,ctcas_main.c,1.27,325
:::ctcasd Daemon Started
Aug 31 14:16:34 alaska user:info no[172498]: Network option nonlocsrcroute was set to the
value 1
Aug 31 14:16:34 alaska user:info no[172498]: Network option ipsrcroutesend was set to the
value 1
Aug 31 14:16:34 alaska user:info no[172498]: Network option ipsrcrouterrecv was set to the
value 1
Aug 31 14:16:34 alaska user:info no[172498]: Network option ipsrcrouteforward was set to
the value 1
Aug 31 14:16:35 alaska daemon:notice cthats[176620]: (Recorded using libct_ffdc.a cv
2):::Error ID: 6UpNEL0H40b8/upk0sn/F7.....:::Reference ID: :::Template ID:
97419d60:::Details File: :::Location: rsct,bootstrp.C,1.215,4477
:::TS_START_ST Topology Services daemon started Topology Services daemon started by: SRC
Topology Services daemon log file location
/var/ct/db2HADomain/log/cthats/cthats.31.1416/var/ct/db2HADomain/run/cthats/ Topology
Services daemon run directory /var/ct/db2HADomain/run/cthats/
Aug 31 14:16:36 alaska daemon:notice cthags[164338]: (Recorded using libct_ffdc.a cv
2):::Error ID: 63Y7ej0I40b8/30Z1sn/F7.....:::Reference ID: :::Template ID:
afa89905:::Details File: :::Location: RSCT,pgsd.C,1.62.1.9,612
:::GS_START_ST Group Services daemon started DIAGNOSTIC EXPLANATION HAGS daemon started
by SRC. Log file is /var/ct/db2HADomain/log/cthags/trace_2_1.
Aug 31 14:16:41 alaska user:info syslog: ifconfig -a

```

```

Aug 31 14:20:16 alaska daemon:notice ConfigRM[160226]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 4bddfbcc::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,16907      ::CONFIGRM_HASQUORUM_ST The operational
quorum state of the active peer domain has changed to HAS_QUORUM. In this state, cluster
resources may be recovered and controlled as needed by management applications.
Aug 31 14:20:16 alaska daemon:notice ConfigRM[160226]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: 3b16518d::Details File: ::Location:
RSCT,PeerDomain.C,1.99.1.311,12054      ::CONFIGRM_ONLINE_ST The node is online in
the domain indicated in the detail data. Peer Domain Name db2HADomain
Aug 31 14:20:19 alaska daemon:notice StorageRM[127062]: (Recorded using libct_ffdc.a cv
2)::Error ID: ::Reference ID: ::Template ID: edff8e9b::Details File: ::Location:
RSCT,IBM.StorageRmD.C,1.41,142         ::STORAGE_RM_STARTED_ST IBM.StorageRM
daemon has started.
Aug 31 14:20:21 alaska daemon:notice RecoveryRM[192560]: (Recorded using libct_ffdc.a cv
2)::ErrorID: 6cqjlq0p70b8/xia0sn/F7.....:Reference ID: ::Template ID:
b60efd88::Details File: ::Location: RSCT,IBM.RecoveryRmD.C,1.21.2.1,136
::RECOVERYRM_INFO_0_ST IBM.RecoveryRM daemon has started.
Aug 31 14:20:24 alaska daemon:notice RecoveryRM[192560]: (Recorded using libct_ffdc.a cv
2)::ErrorID: 6bGpGm/s70b8/g0x/sn/F7.....:Reference ID: ::Template ID:
724b54a7::Details File: ::Location: RSCT,Protocol.C,1.54.1.22,380
::RECOVERYRM_INFO_7_ST This node has joined the IBM.RecoveryRM group. My node number =
2 ; Master node number = 1
Aug 31 14:20:24 alaska user:debug
/usr/sbin/rsct/sapolicies/db2/db2V95_monitor.ksh[188884]: Returning 2 (db2inst1, 0)
Aug 31 14:20:25 alaska auth|security:notice su: from root to db2inst1 at /dev/tty??
Aug 31 14:20:27 alaska user:notice db2V95_start.ksh[184338]: Entered
/usr/sbin/rsct/sapolicies/db2/db2V95_start.ksh, db2inst1, 0

```

## 7. Network interface card (NIC) failures to test EtherChannel:

- Once EtherChannel is configured, you will not be able to use `ifconfig` or any other command to simulate failure of individual bonded adapters.
- Note that as far as cluster manager is concerned the EtherChannel adapter is like any other standard Ethernet adapter. Hence, individual members of the EtherChannel /Link-Aggregation are not visible to cluster manager or TSA, and thus are not controllable or monitored. Therefore, SYSLOGS/RSCT logs are not useful to determine whether EtherChannel has failed over to one of the backup adapters.
- First figure out which Ethernet adapter is defined as the Primary Channel and which one is the backup:

```

root> lsattr -El ent2 | awk '$1 ~/adapter/'
adapter_names  ent0          EtherChannel Adapters          True
backup_adapter ent1          Adapter used when whole channel fails True

```

- Next find out which adapter is currently used by EtherChannel as the "Active Adapter":

```

root> netstat -v | awk '/Active channel/'
Active channel: primary channel

```

- Now we are ready to do failover testing. Two methods to do EtherChannel failover testing:
  - Use AIX EtherChannel Management commands to "force a failover". Issue a command similar to the following to do this (or using `smitty etherchannel -> Force A Failover In An EtherChannel / Link Aggregation`)

```

root> /usr/lib/methods/ethchan_config -f 'ent2'

```

- OR disconnect the Ethernet cable from "Currently Active Adapter"

- After executing the failover, check which adapter is now acting as the "Active channel":

```

root> netstat -v | awk '/Active channel/'
Active channel: backup adapter      <== this indicates we have failed-over successfully

```

## 8. Loss of quorum DISK tiebreaker:

DB2 system topology and configuration for automated multi-site HA and DR

- Start with the two HADR nodes in peer mode, and ensure that the quorum disk tiebreaker can be accessed from both nodes.
- We take note of the Master node/Group Leader of the cluster. The command `lssrc -ls IBM.RecoveryRM` can be used for this.
- From the NetApp filer we disable (offline) the shared LUN used by the two HA nodes as the DISK tiebreaker. We can verify this LUN is now not accessible by host, for example:

```
root> lspath -l hdisk3
Failed hdisk3 iscsi0
```
- The cluster manager/RSCT will not detect loss of a tiebreaker device until ConfigRM issues a SCSI reservation. Hence during the normal operation when the cluster is in “HAS QUORUM” state the loss of the shared disk tiebreaker will go unnoticed (i.e., no warning /error messages will be logged in SYSLOG).
- We do a "role-switch" while the shared DISK tiebreaker is offline to show that no SCSI reservations are exercised -e.g., on standby issue "db2 takeover hadr on db hadb".
- Note that if there is any node failure at this time, the cluster will be brought offline, since at this point none of the nodes will be able to exercise the tiebreaker mechanism and will not be able to establish “HAS QUORUM” state.

## Appendix H - Tivoli SA policies and scripts used by DB2 Integrated HA solution

---

We include a brief overview of the Tivoli SA (TSA) policies, control, and monitoring scripts used by DB2 Integrated HA solution.

### Control and Monitoring scripts

The DB2 Integrated HA solution uses the following control and monitor scripts to manage the cluster resources. These scripts are an integral part of the solution and modification of them in any way is not supported. However, some knowledge of the scripts and what tasks they perform can be helpful in understanding cluster activity.

The Integrated HA solution in DB2 9.5 and DB2 9.7 can automate the most common DB2 HA scenarios, including:

- Single partition shared disk automation
- Multi partition shared disk automation
- HADR automation

It is the HADR automation that we will discuss in more detail here. The HADR automation creates cluster objects for both of the HADR instances and creates a cluster object for each HADR database. (These cluster objects are units of control and are called resources.) For a typical DB2 HADR automation scenario, there are three resources: a resource for the instance hosting the HADR primary database, a resource for the instance hosting the HADR standby database, and a resource for the HADR database automation itself.

The DB2 instance resources are automated (at least partially) by means of the following TSA policies:

- `db2V95_start.ksh` (`db2V97_start.ksh` for DB2 9.7)
- `db2V95_stop.ksh` (`db2V97_stop.ksh` for DB2 9.7)
- `db2V95_monitor.ksh` (`db2V97_monitor.ksh` for DB2 9.7)

Each of the above scripts resides in the `/usr/sbin/rsct/sapolicies/db2/` directory on each node, and the permissions are root executable/modifiable only. So again, no modification of these scripts is supported, but understanding them can be helpful (especially when analyzing the behavior of the cluster using the SYSLOG on each machine).

The `db2V95_start.ksh` script is called whenever the instance needs to be started, and only exits (with a zero return code) once the instance is successfully started at “this” node. The `db2V95_stop.ksh` is called whenever the instance at this node needs to be stopped at “this” node. This script will stop the instance at this node appropriately. The `db2V95_monitor.ksh` script returns the state of the instance at “this” node; the state of the instance is either “1” (the instance is ONLINE at this node) or “2” (the instance is OFFLINE at this node).

The HADR database resource is automated (at least partially) using the following TSA policies:

- `hadrV95_start.ksh` (`hadrV97_start.ksh` for DB2 9.7)
- `hadrV95_stop.ksh` (`hadrV97_stop.ksh` for DB2 9.7)

DB2 system topology and configuration for automated multi-site HA and DR

- `hadrV95_monitor.ksh` (`hadrV97_monitor.ksh` for DB2 9.7)

The `hadrV95_start.ksh` script is called whenever the database needs to be started at “this” node. Starting the database involves either a database activation (for databases whose role is already a HADR primary) or a `HADR TAKEOVER BY FORCE PEER WINDOW ONLY` call (for databases that are in HADR standby role). The `hadrV95_stop.ksh` is called whenever the cluster manager decides that the resource must be taken offline at “this” node.

The `hadrV95_monitor.ksh` script returns the state of the HADR database at “this” node; the state of the HADR database is either “1” (the database is primary at this node) or “2” (the database is not primary at this node).

## **Failover policies supported by DB2 Integrated HA Solution**

A failover policy specifies how a cluster manager should respond when a cluster element such as a network interface card or a database server fails.

### ***Round robin failover policy***

If there is a failure associated with one cluster domain node then the database manager will restart the work from the failed cluster domain node on any other node that is in the cluster domain.

### ***Mutual failover policy***

To configure a mutual failover policy, you associate a pair of nodes in the cluster domain as a system pair. If there is a failure on one of the nodes in this pair, then the database partitions on the failed node will fail over to the other node in the pair. Mutual failover is only available in a DPF configuration.

### ***N Plus M failover policy***

If there is a failure associated with one cluster node then the database partitions on the failed node will fail over to any other node that is in the cluster domain. N Plus M failover is only available in a DPF configuration.

### ***Local restart failover policy***

If there is a failure on one of the cluster nodes, then the database manager will restart the database in place (or locally) on the same node that failed.

### ***HADR failover policy***

When you configure a HADR failover policy, you are enabling the *DB2 High Availability Disaster Recovery (HADR)* feature to manage failover, such as what we have done in this paper. If a HADR primary database fails, the database manager will move the workload from the failed database to a HADR standby database.

### ***Custom failover policy***

When you configure a custom failover policy, you create a list of nodes in the cluster domain onto which the database manager can fail over. If a node in the cluster domain fails, the database manager will move the workload from the failed node to one of the nodes in the list that you specified.

## Notices

---

© Copyright IBM Corporation 2010.

**IBM Canada**  
**8200 Warden Avenue**  
**Markham, ON**  
**L6G 1C7**  
**Canada**

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*  
*IBM Corporation*  
*North Castle Drive*  
*Armonk, NY 10504-1785*  
*U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

DB2 system topology and configuration for automated multi-site HA and DR

**COPYRIGHT LICENSE:**

This information contains sample programs and code in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## **Trademark acknowledgments**

---

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)® are trademarks or registered trademarks of International Business Machines Corporation 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](http://www.ibm.com/legal/copytrade.shtml).

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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