Using DB2 High Availability Disaster Recovery with Tivoli Systems Automation and Reliable Scalable Cluster Technology

Troubleshooting your DB2 high availability integrated solution

Dipali Kapadia
Michelle Chiu

September 30, 2010

The IBM® DB2® High Availability (HA) feature, introduced in IBM® DB2 9.5 for Linux®, UNIX®, and Windows®, enables a new level of integration between the data server and cluster management software, providing a unified High Availability Disaster Recovery (HADR) automation framework. In this tutorial, get an introduction to this integrated solution, and learn about useful diagnostic tools for working with DB2 and Tivoli Systems Automation, a key piece of the solution. Achieve the highest possible level of performance and reliability for your data, understanding how to solve problems and address issues.

This tutorial is meant for readers who have some knowledge of HADR, but may be new to the integrated solution—DB2 with IBM Tivoli® Systems Automation (TSA) and IBM Reliable Scalable Cluster Technology (RSCT).

Introduction to the DB2 High Availability feature

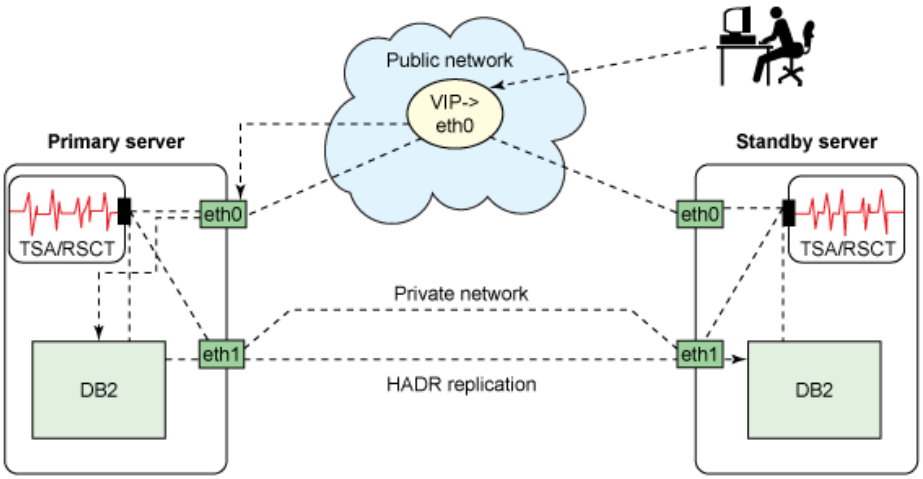
In today's fast-paced world, time is money. More importantly, downtime equals money lost. This is why high availability is so important for all businesses, large and small. A highly available database solution ensures that, should any integral part of your database solution go down, your system would seamlessly fail over to a backup. However, without a cluster manager, this "failover" is not automatic. A database administrator must be paged, and must physically go to the database server and manually issue a takeover command. This is where the power of the integrated HA solution comes into play.

The integrated HA solution, called the DB2 High Availability feature, was introduced in DB2 9.5. In this solution, a cluster manager, Tivoli Systems Automation / Reliable Scalable Cluster Technology (TSA/RSCT), comes bundled with DB2 for Linux, UNIX, and Windows Workgroup Edition and Enterprise Edition. It is responsible for monitoring integral database resources and, in the event of a failure, taking the appropriate action. The main advantages of the integrated solution are:

- It is simple: The DBA does not need to learn a new set of cluster manager commands to manage resources.
- It is integrated: Using the db2haicu tool, DB2 seamlessly interacts with TSA, triggering the correct actions. DB2 comes with TSA/RSCT bundled. As you apply DB2 fix packs, the fix packs automatically upgrade your TSA/RSCT level to pick up critical TSA/RSCT fixes, if necessary.

In a typical integrated HA solution configuration, TSA/RSCT is installed on both hosts. It is responsible for monitoring entities such as network interfaces, DB2 instances, and HADR databases. The client connects to the primary database using the public network using a virtual IP address. In the event of a failure on the primary host, this virtual IP address fails over to the standby. From the perspective of the client, there is no down time, as the transition is automatic. If there are two network interfaces on each host, a private network can be set up exclusively for HADR replication.

Figure 1. Integrated HA solution



Prerequisites

The integrated HA solution using TSA/RSCT as the cluster manager is available for use on Linux and AIX in DB2 9.5 for Linux, UNIX, and Windows and later, and on Solaris in V9.7 and later.

Diagnostic tools in DB2 and TSA/RSCT

This section identifies the tools available from both DB2 and TSA/RSCT that you can use to identify and address troubleshooting issues that may arise in an integrated HA environment. Seasoned DBAs or experienced TSA/RSCT users may already be familiar with many of these tools. However, only through using these utilities together can you narrow down and identify underlying problems. This section describes the role each tool plays in the HA architecture.

DB2 tools

Database configuration parameters

To obtain HADR-related parameters for database pairs (the primary server and the standby server), run the following command on each host:

```
db2 get db cfg for <dbname> | grep HADR
```

Listing 1 shows an example of the output.

Listing 1. Output for HADR-related configuration parameters

```
(db2inst1@host1) /home/db2inst1 $ db2 get db cfg for HADRDB | grep HADR
 HADR database role                                      = STANDARD
 HADR local host name                  (HADR_LOCAL_HOST) = host1
 HADR local service name                (HADR_LOCAL_SVC) = 50001
 HADR remote host name                (HADR_REMOTE_HOST) = host2
 HADR remote service name              (HADR_REMOTE_SVC) = 50002
 HADR instance name of remote server  (HADR_REMOTE_INST) = db2inst1
 HADR timeout value                       (HADR_TIMEOUT) = 120
 HADR log write synchronization mode     (HADR_SYNCMODE) = SYNC
 HADR peer window duration (seconds)  (HADR_PEER_WINDOW) = 300
```

The *HADR database role* is the true role of the database. Its possible values are PRIMARY, STANDBY, and STANDARD. It is always a good idea to compare this with the output of `lssam` (a TSA/RSCT command) to verify that this data is consistent. Table 1 shows how the output of `lssam` will map to the HADR database role.

Table 1. HADR role to OpState mapping

HADR database role	HADR resource OpState from lssam
Primary	Online
Standby	Offline
Standard	Offline

The db2haicu utility uses these parameter values to generate the clustered resource name. Similarly, it may use these values when checking whether a resource already exists in the cluster. Pay special attention to these configurations when troubleshooting setup issues.

The db2haicu tool poses certain naming requirements on top of the HADR parameters. Table 2 summarizes these naming requirements.

Table 2. Naming limitations

DB parameter	HADR	db2haicu
`HADR_LOCAL_HOST`	Any name that can be resolved to the host name (for example, short name, fully qualified domain name, IP address, alias)	Short name; fully qualified domain name (LI74662 in V95FP5, IC62233 in V97FP1)
`HADR_REMOTE_HOST`	Any name that can be resolved to the host name (for example, short name, fully qualified domain name, IP address, alias)	Short name; fully qualified domain name (LI74662 in V95FP5, IC62233 in V97FP1)
`HADR_REMOTE_INST`	Not case-sensitive	Case-sensitive
`HADR_PEER_WINDOW`	0 or greater	Greater than 0

db2pd utility

This utility can be run as follows:

```
db2pd -hadr -db <name>
```

Listing 2 shows an example of the output.

Listing 2. db2pd output

```
(db2inst1@host2) /home/db2inst1$ db2pd -hadr -db hadrdb

Database Partition 0 -- Database HADRDB -- Standby -- Up 0 days 00:16:44 --
Date 08/04/2010 17:08:05

HADR Information:
Role    State                SyncMode HeartBeatsMissed   LogGapRunAvg (bytes)
Standby RemoteCatchup        Sync     0                  365349536

ConnectStatus ConnectTime                           Timeout
Connected     Wed Aug  4 16:51:22 2010 (1280955082) 120

PeerWindowEnd                          PeerWindow
Null (0)                               300

LocalHost                                 LocalService
host2                                     50002

RemoteHost                                RemoteService      RemoteInstance
host1                                  50001              db2inst1

PrimaryFile  PrimaryPg  PrimaryLSN
S0029674.LOG 0          0x000000001F060B7B

StandByFile  StandByPg  StandByLSN         StandByRcvBufUsed
S0008608.LOG 0          0x000000000A8D3EE1 2%
```

The db2pd utility directly accesses the DB2 system memory state. This is the correct role of the database. In certain error cases or time-sensitive scenarios, there could be a discrepancy between how the cluster manager sees the database and how the database sees itself. It is good practice to verify the true state with this tool.

db2diag.log

This log is typically found under ~/sqllib/db2dump/db2diag.log. The location can be configured using the `DIAGPATH` database manager configuration parameter.

This log is the main diagnostic log for the DB2 instance and database. It records diagnostic messages at different notification levels (info/warning/error/severe). The default level for this log is 3 (warning). The log level is configurable by the `DIAGLEVEL` setting. When troubleshooting a reproducible problem, it is good practice to change the `DIAGLEVEL` to 4 for maximum data capture.

```
db2 update dbm cfg using DIAGLEVEL 4
```

By examining the db2diag.log, you can see a history of HADR roles, db2haicu interaction calls with the cluster manager, and some cluster manager error messages. Run `db2diag h` for options to filter and format this log.

Messages related to the db2haicu and integrated HA code are prefixed by 'sqlha'. Since the integrated solution implicitly invokes TSA/RSCT calls underneath the covers, some errors are directly passed down from the cluster manager. These errors are dumped into the db2diag.log

with special syntax (### --- ####-###). The ####-### is a RSCT error code. You can search these errors in the RSCT information center (see Related topics).

If you are able to distinguish where the error originates, this will help IBM Support focus on the right component.

Listing 3 shows an example of the db2diag.log output.

Listing 3. db2diag.log output

```
2010-07-28-09.54.04.342244-240 E1753286A844       LEVEL: Error
PID     : 589910               TID  : 1           PROC : db2haicu
INSTANCE: db2inst1             NODE : 000
EDUID   : 1
FUNCTION: DB2 Common, SQLHA APIs for DB2 HA Infrastructure, sqlhaDeleteResourceGroup, pro
be:300
MESSAGE : ECF=0x90000548=-1879046840=ECF_SQLHA_DELETE_GROUP_FAILED
          Delete group failed
DATA #1 : String, 35 bytes
Error during vendor call invocation
DATA #2 : unsigned integer, 4 bytes
30
DATA #3 : String, 29 bytes
db2_db2inst1_db2inst1_HADRDB-rg
DATA #4 : unsigned integer, 8 bytes
1
DATA #5 : signed integer, 4 bytes
8
DATA #6 : String, 101 bytes
Line # : 7650---2621-008 Failed to update resource because of configuration
data replication errors.
DATA #7 : Object [PD_TYPE_STRING] not dumped
Address: 0x00000001100A90FD Size: 0 Reason: Zero-length data
```

For a quick look at the sequence of HADR role changes, issue the following command:

```
cat db2diag.log | grep -i "hadr state"
```

Listings 4 and 5 show the output from the two servers.

Listing 4. Output of HADR state in db2diag.log on the primary server

```
CHANGE  : HADR state set to None (was None)
CHANGE  : HADR state set to P-Boot (was None)
CHANGE  : HADR state set to P-RemoteCatchupPending (was P-Boot)
CHANGE  : HADR state set to P-RemoteCatchup (was P-RemoteCatchupPending)
CHANGE  : HADR state set to P-NearlyPeer (was P-RemoteCatchup)
CHANGE  : HADR state set to P-Peer (was P-NearlyPeer)
CHANGE  : HADR state set to P-RemoteCatchupPending (was P-Peer)
CHANGE  : HADR state set to P-RemoteCatchup (was P-RemoteCatchupPending)
CHANGE  : HADR state set to P-NearlyPeer (was P-RemoteCatchup)
CHANGE  : HADR state set to P-Peer (was P-NearlyPeer)
```

Listing 5. Output on HADR state in db2diag.log on the standby server

```
CHANGE  : HADR state set to None (was None)
CHANGE  : HADR state set to S-Boot (was None)
CHANGE  : HADR state set to S-LocalCatchup (was S-Boot)
CHANGE  : HADR state set to S-RemoteCatchupPending (was S-LocalCatchup)
CHANGE  : HADR state set to S-RemoteCatchupPending (was S-RemoteCatchupPending)
CHANGE  : HADR state set to S-RemoteCatchup (was S-RemoteCatchupPending)
CHANGE  : HADR state set to S-NearlyPeer (was S-RemoteCatchup)
CHANGE  : HADR state set to S-Peer (was S-NearlyPeer)
```

db2support utility

This collects detailed information from the DB2 instance environment and from the operating system. For HADR issues, you should typically run db2support with the `-d` tag to gather database-related information. IBM support personnel will guide you through its usage when necessary.

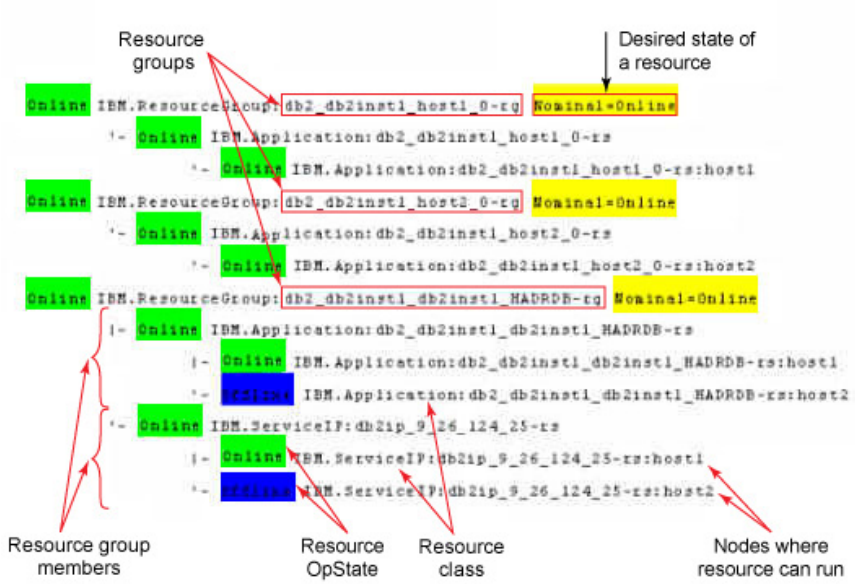
TSA/RSCT tools

Most data collected by the user will be DB2-related, while necessary TSA/RSCT data will be missing. This section describes how to obtain relevant TSA/RSCT information to help give insight into why the cluster manager behaves in a particular way.

`lssam` command

The `lssam` command is used to obtain a snapshot of the clustered resource states. Figure 2 (`lssam` explained) dissects and explains an example typical output from this command. Each resource group has a nominal state—the desired state. A resource group can contain one or many resource group members, and each resource group contains one or many resources. In Figure 2, there are three resource groups shown—one for the local instance, one for the remote instance, and one for the HADR database. The resource OpState is the actual state of the resource, and the resource class is the TSA class that the resource belongs to. The name of the resource also contains a host name, which is the node where the resource may run.

Figure 2. `lssam` explained



TSA/RSCT monitors the DB2 instance and database behaviour by running various DB2 scripts on a specified interval. The scripts are located under /usr/sbin/rsct/sapolicies/db2 on both hosts in an HADR pair. For this solution, there are six main scripts. The scripts use a DB2 utility called db2gcf to interact with the instance and HADR database. See the DB2 information center for further information on db2gcf (see Related topics for a link).

The name of each script contains the current DB2 version. For the purpose of this tutorial, we will use V9.7.

Instance monitor script (db2V97_monitor.ksh)

The instance monitor script is responsible for monitoring the instance status. By default, TSA/RSCT calls this on a specified interval on both nodes in the HADR pair. Based on this, the OpState of the instance resources are updated. In Figure 2, the instance resources are db2_db2inst1_host1_0-rs and db2_db2inst1_host2_0-rs.

Table 3. Instance monitor script return codes

OpState	Script return code	Meaning (DB2 perspective)
Online	1	Instance is started.
Offline	2	Instance is not started, or a problem occurred while getting status.

Instance start script (db2V97_start.ksh)

The instance start script is responsible for starting up the DB2 instance and activating its databases. If necessary, the start script can also perform reintegration.

Instance stop script (db2V97_stop.ksh)

The instance stop script is responsible for stopping the instance. It is used to deliberately stop the instance in the case where something has gone wrong on the respective host.

HADR monitor script (hadrV97_monitor.ksh)

The HADR monitor script is responsible for monitoring HADR database status. By default, TSA/RSCT calls this script every 29 seconds on both hosts in the HADR pair. Based on this, the OpState of the HADR resources are updated. In Figure 2, the HADR resource is db2_db2inst1_db2inst1_HADRDB-rs.

Table 4. HADR monitor script return codes

OpState	Script return code	Meaning (DB2 perspective)
Online	1	HADR is primary.
Offline	2	HADR is standby, or a problem occurred while getting status.

HADR start script (hadrV97_start.ksh)

The HADR start script is responsible for starting HADR on an existing standby database, to have it assume a primary role.

HADR stop script (hadrV97_stop.ksh)

The HADR stop script is responsible for stopping HADR. It is used to deliberately stop HADR in the case where something has gone wrong on the respective host.

syslog

syslog shows a history of resource management by TSA/RSCT. It also shows certain types of TSA/RSCT errors, if they are present. A combined analysis of the syslog with db2diag.log can show the sequence of events that took place during a failure scenario.

On Linux systems, syslog can be found under /var/log/messages/syslog.out.

On AIX systems, syslog can be found under /tmp/syslog.out.

Depending on how your system is customized, the syslog may be located elsewhere. To modify this, look in syslog.conf. This file can be found under /etc.

When used in combination with the db2diag.log, the syslog can be a powerful debugging tool. Cross-comparing key timestamps between the two logs can greatly help in reconstructing the sequence of events in a failover scenario.

For example, Listing 6 shows an instance killed around 22:05:44.

Listing 6. Killed instance

```
db2inst1@host1:~> date
Wed Aug  4 22:05:44 EDT 2010
db2inst1@host1:~> db2_kill
ipclean: Removing DB2 engine and client's IPC resources for db2inst1.
```

TSA/RSCT should invoke the start script. Look in the syslog around 22:05:44 to confirm.

Listing 7. Start script

```
Aug  4 22:05:53 host1 db2V97_start.ksh[9540]: Entered /usr/sbin/rsct/sapolicies/db2
                      /db2V97_start.ksh, db2inst1, 0
Aug  4 22:05:53 host1 db2V97_start.ksh[9541]: Able to cd to /home/db2inst1/sqllib :
                      /usr/sbin/rsct/sapolicies/db2/db2V97_start.ksh, db2inst1, 0
Aug  4 22:05:53 host1 db2V97_start.ksh[9551]: 1 partitions total: /usr/sbin/rsct
                      /sapolicies/db2/db2V97_start.ksh, db2inst1, 0
Aug  4 22:06:00 host1 db2V97_start.ksh[9774]: Returning 0 from /usr/sbin/rsct
                      /sapolicies/db2/db2V97_start.ksh ( db2inst1, 0)
```

The instance is successfully started. Look in the db2diag.log around 22:06:00 to confirm.

Listing 8. Instance started

```
2010-08-04-22.05.57.453549-240 E10236E305          LEVEL: Event
PID     : 9620                 TID  : 47284539421760PROC : db2star2
INSTANCE: db2inst1             NODE : 000
FUNCTION: DB2 UDB, base sys utilities, DB2StartMain, probe:911
MESSAGE : ADM7513W  Database manager has started.
START   : DB2 DBM
```

TSA/RSCT should invoke the HADR start script. Look in the syslog around 22:06:00 to confirm.

Listing 9. HADR start script

```
Aug  4 22:06:14 host1 /usr/sbin/rsct/sapolicies/db2/hadrV97_start.ksh[9948]: ...returns 0
Aug  4 22:06:14 host1 /usr/sbin/rsct/sapolicies/db2/hadrV97_start.ksh[9949]: Returning 0
                      : db2inst1 db2inst1 HADRDB
```

Once HADR has started, it will attempt to reach PEER state. Look in the db2diag.log around 22:06:00, searching for keyword "HADR state."

Listing 10. Verify HADR PEER state

```
2010-08-04-22.06.01.254052-240 E11446E359          LEVEL: Event
PID     : 9634                 TID  : 47103946516800PROC : db2sysc
INSTANCE: db2inst1             NODE : 000
EDUID   : 27                   EDUNAME: db2hadrp (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSetHdrState, probe:10000
CHANGE  : HADR state set to None (was None)

2010-08-04-22.06.01.256495-240 E12922E361          LEVEL: Event
PID     : 9634                 TID  : 47103946516800PROC : db2sysc
INSTANCE: db2inst1             NODE : 000
EDUID   : 27                   EDUNAME: db2hadrp (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSetHdrState, probe:10000
CHANGE  : HADR state set to P-Boot (was None)

2010-08-04-22.06.01.964755-240 E14817E379          LEVEL: Event
PID     : 9634                 TID  : 47103946516800PROC : db2sysc
INSTANCE: db2inst1             NODE : 000
EDUID   : 27                   EDUNAME: db2hadrp (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSetHdrState, probe:10000
CHANGE  : HADR state set to P-RemoteCatchupPending (was P-Boot)

2010-08-04-22.06.03.184782-240 E22305E388          LEVEL: Event
PID     : 9634                 TID  : 47103946516800PROC : db2sysc
INSTANCE: db2inst1             NODE : 000
EDUID   : 27                   EDUNAME: db2hadrp (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSetHdrState, probe:10000
CHANGE  : HADR state set to P-RemoteCatchup (was P-RemoteCatchupPending)

2010-08-04-22.06.03.304770-240 E23429E378          LEVEL: Event
PID     : 9634                 TID  : 47103946516800PROC : db2sysc
INSTANCE: db2inst1             NODE : 000
EDUID   : 27                   EDUNAME: db2hadrp (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSetHdrState, probe:10000
CHANGE  : HADR state set to P-NearlyPeer (was P-RemoteCatchup)

2010-08-04-22.06.03.317330-240 E23808E369          LEVEL: Event
PID     : 9634                 TID  : 47103946516800PROC : db2sysc
INSTANCE: db2inst1             NODE : 000
EDUID   : 27                   EDUNAME: db2hadrp (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSetHdrState, probe:10000
CHANGE  : HADR state set to P-Peer (was P-NearlyPeer)
```

`lsrsrc`

There are other TSA/RSCT commands used to display specific attributes of a resource. `lsrsc` is a useful command to understand the inner workings of various resources belonging to a particular resource class. The most common classes you would work with are IBM.Application, IBM.ServiceIP, and IBM.Test. These classes are created by TSA/RSCT and act as a doorway for other products to interact with the cluster manager.

Cluster resource

Most commonly, cluster resources belong to the class IBM.Application. Cluster resources are created to represent entities such as DB2 instance or HADR database. When db2haicu is run, it uses TSA/RSCT commands to create these resources. For each cluster resource, there are many attributes. The `start`, `stop`, and `monitor` command attributes are tied to the DB2 scripts discussed in "`lssam` command" section. This is how TSA/RSCT knows what to call to administer a resource.

This utility can be run as follows:

```
lsrsrc -s "Name like 'db2_db2inst1%" IBM.Application
```

Listing 11. Running the utility

```
db2inst1@host1:~> lsrsrc -s "Name like 'db2_db2inst1_db2inst1_HADRDB-rs' AND
                  NodeNameList={'host1'}" IBM.Application
Resource Persistent Attributes for IBM.Application
resource 1:
Name                  = "db2_db2inst1_db2inst1_HADRDB-rs"

ResourceType           = 0
AggregateResource     = "0x2028 0xffff 0xa5530134 0xe14e5051 0x91c6ee58
                         0xa6a94e38"
StartCommand          = "/usr/sbin/rsct/sapolicies/db2/hadrV97_start.ksh
                         db2inst1 db2inst1 HADRDB"
StopCommand           = "/usr/sbin/rsct/sapolicies/db2/hadrV97_stop.ksh
                         db2inst1 db2inst1 HADRDB"
MonitorCommand        = "/usr/sbin/rsct/sapolicies/db2/hadrV97_monitor.ksh
                         db2inst1 db2inst1 HADRDB"
MonitorCommandPeriod  = 21
MonitorCommandTimeout = 29
StartCommandTimeout   = 330
StopCommandTimeout    = 140
UserName              = "root"
RunCommandsSync       = 1
ProtectionMode        = 1
HealthCommand         = ""
HealthCommandPeriod   = 10
HealthCommandTimeout  = 5
InstanceName          = ""
InstanceLocation      = ""
SetHealthState        = 0
MovePrepareCommand    = ""
MoveCompleteCommand   = ""
MoveCancelCommand     = ""
CleanupList           = {}
CleanupCommand        = ""
CleanupCommandTimeout = 10
ProcessCommandString  = ""
ResetState            = 0
ReRegistrationPeriod  = 0
```

```
CleanupNodeList       = {}
MonitorUserName       = ""
ActivePeerDomain      = "hadr_domain"
NodeNameList          = {"host1"}
```

Flag resource

Flag resources belong to the class IBM.Test. Flag resources are transient in nature. They are created by DB2 code and used to communicate ongoing action to TSA. At the end of the action they should be cleaned up automatically. If a flag is left behind, this is an indicator that something went wrong. This flag is created by the new primary database as an indicator to the old primary database that it should "reintegrate" itself, meaning it should switch its role to become standby. Checking flags is a good method for debugging; however, in normal scenarios you do not need to monitor this.

This utility can be run as follows:

```
lsrsrc IBM.Test
```

Listing 12. Running the utility

```
db2inst1@host2:~> lsrsrc IBM.Test
Resource Persistent Attributes for IBM.Test
resource 1:
        Name              = "db2_HADRDB_host1_Reintegrate_db2inst1_db2inst1"
        ResourceType      = 0
        AggregateResource = "0x3fff 0xffff 0x00000000 0x00000000 0x00000000
                             0x00000000"
        ForceOpState      = 0
        TimeToStart       = 0
        TimeToStop        = 0
        WriteToSyslog     = 0
        MoveTime          = 0
        MoveFail          = 0
        ForceMoveState    = 0
        ActivePeerDomain  = "hadr_domain"
        NodeNameList      = {"host2"}
```

getsadata

This is the equivalent of db2support from theTSA/RSCT perspective. To gather maximum data, run getsadata with the all flag. This utility must be run as root. IBM Support can guide you on this.

Troubleshooting

Oftentimes, a customer will encounter a problem, but by the time they open a PMR they have lost the opportunity to gather time- and scenario-specific data. The purpose of this section is to help you identify what problem category you may be experiencing so that you can gather the appropriate diagnostic information. This will help IBM Support address the issue with greater efficiency. In some cases, it may even help you resolve the problem on your own.

After studying customer issues encountered over the past couple years, we have come up with four major categories of problems:

- Unexpected failover
- No failover
- Reintegration failure
- Configuration issues with db2haicu

The following sections examine each of these areas.

Unexpected failover

In an unexpected failover, an event occurs on the primary that should not result in the standby taking over. However, failover does occur, resulting in potential data loss and outage.

Considerations:

- Is the solution behaving as designed?
  - Scenarios where a failover is expected:
    - Primary machine goes down (for example, power off)
    - Primary machine public network failure
    - Primary instance failure
- Scenarios where failover should not occur:
  - Killing the instance (for example, db2_kill)
  - Loss of the private network
- Why was there a failover?
  - Original state of the HADR database resource is not primary
  - Instance and HADR start script are not invoked
  - Instance and HADR start script are unsuccessful

Consider the original role of the HADR database resource

TSA/RSCT guarantees that the HADR database resource will be online (primary) on one host and offline (standby) on the other. If this is not the case, what seems like an unexpected failover may actually be TSA's attempt to ensure the resource is online on an available host. There are several ways to check the HADR database role:

- Check the db2diag.log on the original primary. (See the "db2diag.log" section for details on this.) Is the HADR state primary to begin with? Does it remain this way leading up to the failover event?
- Check the database configuration parameters on the original primary. (See the "Database configuration parameters" section.). What is the database role and does this match `lssam` database resource OpState?
- Check db2pd output on the original primary. (See the "db2pd utility" section for details on this.) What is the database role, and does this match `lssam` database resource OpState?

The original role was primary, now what?

The first and most obvious question you must consider is: were any start scripts invoked? Keep in mind that if the instance died, its HADR database would also be unavailable as a result. Thus, TSA/RSCT would invoke both the instance and HADR start scripts. If only the HADR database became unavailable, then just the HADR start script would be invoked.

- Open up the syslog on the original primary.
- Do you see any script invocations?
  - Yes - Continue debugging.
  - No - This is very suspicious. As mentioned in the "syslog" section, the syslog contains a history of resource management by TSA. You should see the monitor script being called at specific intervals. Ask yourself, "Is the cluster configured correctly? Is logging enabled and at the correct level?"
- Do you see a start script invocation? Look for the instance monitor script "Returning 2." Shortly after this, you should see the instance start script invocation.
  - Yes - Continue debugging.
  - No - This is likely a TSA/RSCT problem. Contact IBM Support.
- Did the start script return successfully (in other words, "Returning 0")?
  - Yes - This means that the instance started correctly. However, something occurred shortly after this point to cause a failover. Look in the db2diag.log around this time frame to see if there are any further errors.

db2 activate failure

Once the instance has started, the database must be activated in order for HADR to continue replication. Activation of an inconsistent database would fail. A database would become inconsistent if the instance crashed (for example, killing the instance with db2_kill). If the database has AUTORESTART enabled, a restart would be triggered and crash recovery performed, returning the database to a consistent state.

- Check that AUTORESTART is enabled and try activating manually.

```
db2 get db cfg for <database> | grep AUTORESTART
db2 activate db <database>
```

  - No - Let's take a look at some of the possible root causes.
    - db2gcf timeout
    - db2start failure

db2gcf timeout

The instance start script uses the utility db2gcf to start the instance. If db2gcf timed out, you would see db2diag.log message indicating this.

Listing 13. db2diag.log message

```
2010-06-05-02.05.33.824563-240 I18731842A257      LEVEL: Error
PID     : 1831554              TID : 1
FUNCTION: DB2 Common, Generic Control Facility, GcfCaller::start, probe:40
MESSAGE : ECF=0x9000028C Timeout occured while calling a GCF interface function
```

Try starting the instance manually, and see how long this takes (db2start).

Try starting the instance using db2gcf without specifying a timeout, and see how long this takes. The start time varies, depending on the number of partitions in your environment; however, it should not be more than several seconds per partition.

```
db2gcf -u -i <instance>
```

db2start failure

If the instance started successfully, there would definitely be messages in the db2diag.log indicating this.

Listing 14. db2diag.log message

```
2010-08-04-22.05.57.453549-240 E10236E305          LEVEL: Event
PID     : 9620                 TID  : 47284539421760PROC : db2star2
INSTANCE: db2inst1             NODE : 000
FUNCTION: DB2 UDB, base sys utilities, DB2StartMain, probe:911
MESSAGE : ADM7513W  Database manager has started.
START   : DB2 DBM
```

Try starting the instance manually to see if this is successful.

TSA/RSCT errors

DB2 interacts with the cluster manager by issuing TSA/RSCT commands. If these commands are unsuccessful, there would be a log entry with a specific identifier in db2diag.log. (See the "db2diag.log" section for details on this.)

The role was not primary, now what?

The main question here is why was the original role not primary? Is there a bigger problem at play?

When you cluster an instance using db2haicu, TSA/RSCT resources are created as well as dependencies between them. To see dependencies in your cluster you can use the `lsrel` command.

The utility can be run as follows:

```
lsrel -Ab
```

Listing 15. `lsrel` command

```
db2inst1@host1:~> lsrel -Ab
Displaying Managed Relationship Information:
All Attributes

Managed Relationship 1:
        Class:Resource:Node[Source] = IBM.Application:db2_db2inst1_host1_0-rs
        Class:Resource:Node[Target] = {IBM.Equivalency:db2_public_network_0}
        Relationship                = DependsOn
        Conditional                 = NoCondition
        Name                        = db2_db2inst1_host1_0-
                                      rs_DependsOn_db2_public_network_0-rel
        ActivePeerDomain            = hadr_domain
        ConfigValidity              =
```

An example of a dependency is the one created between the instance resource and the public network. If the public network goes down, clients can no longer connect to the database so a failover is necessary. Check if the public network is up.

Next step

If after going through this section your problem has not been solved, this would be a good point to open a PMR. Include in your description all your findings based on the guidance given above. In addition to this, please provide the following:

- Output of any commands run manually
- db2support
- getsadata

No failover

This section describes how to troubleshoot scenarios where a failover is expected but did not occur. There are a few questions that you should ask yourself before diving deep into further diagnostic steps:

- Should a failover occur?
  - Scenarios where we expect a failover are:
    - Primary machine goes down (for example,. power off)
    - Primary machine public network failure
    - Primary instance failure
- Scenarios where failover should not occur:
  - Killing the instance (for example,. db2_kill)
  - Loss of the private network
- If failover is expected but it did not occur, what actually happened?
  - Failure was not detected, hence no action
  - Failure was detected, but failover was not initiated
  - Failover was attempted but failed

To pick the correct category, let's review the action TSA/RSCT would take in an event of a failure.

There is a common misconception that a failover should always take place when there is a failure on primary. This is not always true. In scenarios where the primary can be brought up quickly, a failover may not be required. First, TSA/RSCT would try to restart the instance, the HADR database, or both on the same host. If that fails, then it would start up the HADR database on the standby host.

For example, when the db2sysc process is killed, no failover would take place if the HADR primary database is able to restart in the allotted time. However, in a prolonged failure scenario (for example, the instance cannot come back up or takes too long to restart), a failover is expected.

Was the failure detected?

A clustered resource is monitored by invoking its monitor script and evaluating its return code. Around the time of failure, you should see a change in the return code from the resource monitor script.

- Open up the syslog on the original primary host.
- Do you see any HADR monitor script entries?
  - Yes - Continue debugging.
  - No - In a correct setup, the monitor script should be run periodically to monitor the state of the HADR database resource. Is logging enabled and at the correct level? Verify that the cluster domain is configured correctly.

> **Tip**
>
> You can search for the instance monitor script "Returning 2" to identify the point when failure of the DB2 instance is detected.

- Scan through the HADR monitor script entries and look for a time frame when the return code changes from "Returning 1" to "Returning 2." Do you see this change?
  - Yes - This is the time when TSA/RSCT first detected that the primary database is down. The failure was detected correctly. Go to the question "Was a failover attempt initiated?"
  - No - Continue debugging.
- Do you see any timeout errors?
  - Yes - The monitor script is taking too long to obtain the status of the resource for the cluster manager. Run the script in verbose mode to see where it is spending the most time or if the script hangs.

Listing 16. Running the script in the verbose mode

```
root@host2:/usr/sbin/rsct/sapolicies/db2#
./hadrV97_monitor.ksh db2inst1 db2inst1
HADRDB verbose
+ CT_MANAGEMENT_SCOPE=2
+ export CT_MANAGEMENT_SCOPE
+ CLUSTER_APP_LABEL=db2_db2inst1_db2inst1_HADRDB
+ [[ verbose == verbose ]]
+ typeset +f
+ typeset -ft main monitorhadr set_candidate_P_instance
+ main db2inst1 db2inst1 HADRDB verbose
+ set_candidate_P_instance
+ candidate_P_instance=db2inst1
+ candidate_S_instance=db2inst1
+ + hostname
+ tr .
+ awk {print $1}
localShorthost=host2
+ [[ db2inst1 != db2inst1 ]]
+ return 0
+ + lsrsrc -s Name = 'db2_HADRDB_host2_Reintegrate_db2inst1_db2inst1'
IBM.Test Name
+ grep -c Name
nn=0
+ [[ 0 -eq 0 ]]
+ [[ db2inst1 != db2inst1 ]]
+ [[ 0 -ne 0 ]]
+ monitorhadr
```

  - No - If the monitor script is always "Returning 1," then TSA/RSCT thinks the resource is still online and no action will be taken.
    - Does this match the status from DB2 perspective?
    - Check db2diag.log, and use db2pd for clues on the database instance health and the last HADR role changes.

Was a failover attempt initiated?

In a scenario where failover is expected (for example, the resource cannot be restarted on the original host), once the failure is detected, a failover should take place. For DB2 HADR resource, this means an invocation of the HADR start script on the old Standby host.

- Open up the syslog on the original standby host.
- Focus in on the relevant time frame.
- Do you see any script invocations?
  - Yes - Continue debugging.
  - No - This is very suspicious. You should see the monitor script being called at a certain time interval in the syslog. Ask yourself, "Is the cluster configured correctly? Is logging to syslog enabled?"
- Do you see any HADR start script invocations?
  - Yes - This means TSA/RSCT attempted to start the resource on the standby host. This is the correct behavior so far. Go to the question "Was the failover successful?"
  - No - This is likely a TSA/RSCT problem. Contact IBM Support.

Was the failover successful?

> **Tip**
>
> You can verify HADR database state by looking in the db2diag.log. Search for the keyword "HADR state" to see the transition into primary.

- Did the HADR start script return successfully (in other words, "Returning 0")?
  - Yes - This means the database was started successfully as Primary on this originally Standby host.
    At this point in time, if you look at the `lssam` output, you would see a role switch, where the HADR database resource is online on the old standby host and failed offline on the old primary host.

Listing 17. `lssam` output

```
root@host1:/# lssam
Online IBM.ResourceGroup:db2_db2inst1_host1_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_host1_0-rs
                '- Online IBM.Application:db2_db2inst1_host1_0-rs:host1
Failed offline IBM.ResourceGroup:db2_db2inst1_host2_0-rg Nominal=Online
        '- Failed offline IBM.Application:db2_db2inst1_host2_0-rs
                '- Failed offline IBM.Application:db2_db2inst1_host2_0-rs:host2
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg
Control=MemberInProblemState
Nominal=Online
        '- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
           Control=MemberInProblemState
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:host1
                '- Failed offline
                IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:host2
Online IBM.Equivalency:db2_db2inst1_host1_0-rg_group-equ
        '- Online IBM.PeerNode:host1:host1
Online IBM.Equivalency:db2_db2inst1_host2_0-rg_group-equ
        '- Online IBM.PeerNode:host2:host2
Online IBM.Equivalency:db2_db2inst1_db2inst1_HADRDB-rg_group-equ
```

```
        |- Online IBM.PeerNode:host2:host2
        '- Online IBM.PeerNode:host1:host1
```

Continue troubleshooting. A takeover did occur successfully, so something must have happened later on, causing an apparent failure to failover. Follow through syslog entries to piece together the next sequence of events.

Note: Look at the HADR monitor script return codes. Do they initially return 1 and then later on return 2 or time out? If the HADR monitor script returns 2 later on, then a second failure occurred. Verify this with the db2diag.log.

- No - Using HADR start script, TSA/RSCT is not able to start up the database as primary on the standby host. Examine the db2diag.log to diagnose further. Always match the timestamps in the syslog with the timestamps in the db2diag.log file. Are there any db2diag.log error messages around the same time frame that the HADR start script fails? Let's take a look at some possible root causes.

Peer window expires

If the `HADR_PEER_WINDOW` configuration parameter is too small, it may expire by the time the cluster manager tries to issue a takeover from the standby. In Listing 18 below, `HADR_PEER_WINDOW` is set to 5.

Listing 18. Standby db2diag.log

```
2010-08-05-10.30.21.185557-240 I609665A427        LEVEL: Severe
PID     : 741466               TID  : 4502        PROC : db2sysc
INSTANCE: db2inst1              NODE : 000
EDUID   : 4502                 EDUNAME: db2hadrs (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrEduAcceptEvent, probe:20215
RETCODE : ZRC=0x8280001B=-2105540581=HDR_ZRC_COMM_CLOSED
          "Communication with HADR partner was lost"

2010-08-05-10.30.21.185758-240 E610093A373        LEVEL: Event
PID     : 741466               TID  : 4502        PROC : db2sysc
INSTANCE: db2inst1              NODE : 000
EDUID   : 4502                 EDUNAME: db2hadrs (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSetHdrState, probe:10000
CHANGE  : HADR state set to S-DisconnectedPeer (was S-Peer)

2010-08-05-10.30.21.186057-240 I610467A356        LEVEL: Warning
PID     : 741466               TID  : 4502        PROC : db2sysc
INSTANCE: db2inst1              NODE : 000
EDUID   : 4502                 EDUNAME: db2hadrs (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrCloseConn, probe:30595
MESSAGE : Peer window end time :1281018623



2010-08-05-10.30.24.198297-240 I612083A367        LEVEL: Warning
PID     : 741466               TID  : 4502        PROC : db2sysc
INSTANCE: db2inst1              NODE : 000
EDUID   : 4502                 EDUNAME: db2hadrs (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrEduAcceptEvent, probe:20202
MESSAGE : Peer window ends. Peer window expired.

2010-08-05-10.30.24.198729-240 E612451A389        LEVEL: Event
PID     : 741466               TID  : 4502        PROC : db2sysc
INSTANCE: db2inst1              NODE : 000
EDUID   : 4502                 EDUNAME: db2hadrs (HADRDB)
```

```
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSetHdrState, probe:10000
CHANGE  : HADR state set to S-RemoteCatchupPending (was S-DisconnectedPeer)
```

By the time the failure on the primary database has been detected by TSA/RSCT and a failover is deemed necessary, the peer window has already expired. This would result in the HADR start script failing.

Listing 19. Standby syslog

```
Aug  5 10:31:15 host1 user:notice /usr/sbin/rsct/sapolicies/db2/hadrV97_monitor.ksh
                                  [663766]: Returning 2 : db2inst1 db2inst1 HADRDB
Aug  5 10:31:17 host1 user:debug /usr/sbin/rsct/sapolicies/db2/db2V97_monitor.ksh
                                  [692326]: Returning 1 (db2inst1, 0)
Aug  5 10:31:27 host1 user:debug /usr/sbin/rsct/sapolicies/db2/db2V97_monitor.ksh
                                  [471160]: Returning 1 (db2inst1, 0)
Aug  5 10:31:34 host1 user:notice /usr/sbin/rsct/sapolicies/db2/hadrV97_start.ksh
                                  [471162]: Entering : db2inst1 db2inst1 HADRDB
Aug  5 10:31:36 host1 user:notice /usr/sbin/rsct/sapolicies/db2/hadrV97_monitor.ksh
                                  [598066]: Entering : db2inst1 db2inst1 HADRDB
Aug  5 10:31:37 host1 user:debug /usr/sbin/rsct/sapolicies/db2/db2V97_monitor.ksh
                                  [700494]: Returning 1 (db2inst1, 0)
Aug  5 10:31:39 host1 user:notice /usr/sbin/rsct/sapolicies/db2/hadrV97_monitor.ksh
                                  [598080]: Returning 2 : db2inst1 db2inst1 HADRDB
Aug  5 10:31:44 host1 user:notice /usr/sbin/rsct/sapolicies/db2/hadrV97_start.ksh
                                  [471192]: Returning 3 : db2inst1 db2inst1 HADRDB
```

Listing 20. Standby db2diag.log around the HADR start script invocation

```
2010-08-05-10.31.36.489581-240 I728601A485        LEVEL: Error
PID     : 741466               TID  : 4502        PROC : db2sysc
INSTANCE: db2inst1              NODE : 000
EDUID   : 4502                 EDUNAME: db2hadrs (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSDoTakeover, probe:47004
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"

2010-08-05-10.31.36.489773-240 I729087A363        LEVEL: Warning
PID     : 741466               TID  : 4502        PROC : db2sysc
INSTANCE: db2inst1              NODE : 000
EDUID   : 4502                 EDUNAME: db2hadrs (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSDoTakeover, probe:47008
MESSAGE : Info: Standby has failed to takeover.
```

Try increasing the `HADR_PEER_WINDOW` value.

Cluster manager errors

In the db2diag.log error message, do you find errors with the identifier "###---####-###"? This may indicate an error directly returned by TSA/RSCT or TSA/RSCT being non responsive at the time. Refer to the RSCT information center for further details (see Related topics for a link).

Takeover failed

As you may have gathered, the main step in the old standby assuming a primary role is a "db2 takeover." Look through the db2diag.log to see if there are any messages having to do with this. Listing 21 illustrates what a successful takeover message would look:

Listing 21. Successful takeover message

```
2010-08-06-10.31.20.530184-240 I8024A377          LEVEL: Warning
PID     : 622806               TID  : 4405        PROC : db2sysc
INSTANCE: db2inst1             NODE : 000
EDUID   : 4405                 EDUNAME: db2hadrp (HADRDB)
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrSDoTakeover, probe:47003
MESSAGE : Info: Standby has completed takeover (now primary).
```

Do you see this?

Check the current HADR state of your database (by db2pd). If it is not primary, try to manually issue a db2 takeover.

Listing 22. Issue a db2 takeover

```
db2inst1@host2) /vbs/engn/ha/salinux $ db2 takeover hadr on db HADRDB
DB20000I  The TAKEOVER HADR ON DATABASE command completed successfully.
```

Next step

If after going through this section your problem has not been solved, this would be a good point to open a PMR. Include in your description all your findings based on the guidance given above. In addition to this, provide:

- Output of any commands run manually
- db2support
- getsadata

Reintegration failure

After a failover, the old primary must come back and function as the new standby, which is known as "reintegration." In some cases reintegration may fail, sending the cluster domain into an undesirable state. This section explores the possible causes for an integration failure and expands on the steps to further debug.

Questions to ask yourself:

- Did HADR start successfully on the new primary?
- Did the instance start successfully on the old primary?

Did HADR start successfully on the new primary?

As previously mentioned, reintegration is when the old primary restarts as standby. What that means is TSA/RSCT called the HADR start script on the old standby, which in turn resulted in a "db2 takeover" command being issued. If the takeover is not successful, you are in a "No failover" situation. Refer to "No failover" section for assistance on this.

Did the instance start successfully on the old primary?

The first step in reintegration is to start the instance on the old primary, if it is not already started.

- Open up the syslog on the old primary.
- Do you see any script invocations?
  - Yes - Continue debugging.
  - No - This is very suspicious. You should see the monitor script being called at a certain time interval in the syslog. Ask yourself, "Is the cluster configured correctly? Is logging to syslog enabled?"
- Do you see an instance start script invocation? (Look for the instance monitor script "Returning 2." Shortly after this you should see the instance start script invocation.)
  - Yes - Continue debugging.
  - No - This is likely a TSA/RSCT problem. Contact IBM Support.
- Did the instance start script return successfully ("Returning 0")?
  - Yes - This means the instance started successfully and should have reintegrated. Check the database configuration parameters or db2pd to confirm HADR database role.
  - No - Look at the following possible root causes.

db2gcf timeout

The instance start script uses the utility db2gcf to start the instance. If db2gcf timed out, you would see db2diag.log message indicating this.

Listing 23. db2diag.log message - db2gcf timed out

```
2010-06-05-02.05.33.824563-240 I18731842A257      LEVEL: Error
PID     : 1831554              TID : 1
FUNCTION: DB2 Common, Generic Control Facility, GcfCaller::start, probe:40
MESSAGE : ECF=0x9000028C Timeout occurred while calling a GCF interface function
```

Try starting the instance manually and see how long this takes (db2start). You could also try starting the instance using db2gcf without specifying a timeout and see how long this takes:

```
db2gcf -u -i db2inst1
```

The start time varies, depending on the number of partitions in your environment. However, it should not be more than several seconds per partition.

db2start failure

If the instance started successfully, there would definitely be a message in the db2diag.log indicating this.

Listing 24. db2diag.log message - successful start message

```
2010-08-04-22.05.57.453549-240 E10236E305          LEVEL: Event
PID     : 9620                 TID  : 47284539421760PROC : db2star2
INSTANCE: db2inst1             NODE : 000
FUNCTION: DB2 UDB, base sys utilities, DB2StartMain, probe:911
MESSAGE : ADM7513W  Database manager has started.
START   : DB2 DBM
```

Try starting the instance manually to see if this is successful.

Reintegration failure

In this case, the instance has started successfully but the old primary HADR database has failed to start as standby. As mentioned in the "`lsrsrc`" section, flags are a way for DB2 to communicate with TSA/RSCT. In the case of reintegration, the new primary would create a reintegration flag. This is a signal for the old primary to start itself as standby. Once the reintegration is complete, the flag is deleted. Check if the reintegration flag still exists. If the flag still exists, this is a strong indication that failure to reintegrate had something to do with the flag detection.

Table 5 lists some APARS relating to reintegration failure. Read through each one and see if it applies to your scenario.

Table 5. Reintegration APARS

APAR	Fixed in
IC65836 / IC65837	V95FP6 / V97FP2
IC64142 / IC64666	V95FP6 / V97FP2
IC67393	V97FP2

Next step

If after going through this section your problem has not been solved, this would be a good point to open a PMR. Include in your description all your findings based on the guidance given. In addition to this, provide the following:

- Output of any commands run manually
- db2support
- getsadata

Configuration issues with db2haicu

In order to cluster an HADR setup, the db2haicu tool is used. This tool interacts with TSA/RSCT under the covers to create resources to manage various DB2 entities. When run in interactive mode, the user would be prompted with a series of questions. In most cases, if the user enters an incorrect/improper answer, he would be prompted as such.

In this section, find guidance on how to identify those cases where incorrect input is not corrected immediately (in certain versions of db2haicu), resulting in improper functioning of the cluster.

There are two ways db2haicu can fail during setup:

- Invalid inputs
- TSA/RSCT command failure

To distinguish between issues due to invalid inputs and TSA/RSCT errors, look through the db2diag.log. Focus on logs for the functions in the family "sqlha." Do you see the ####-###

identifier in the data field? If so, this is an error directly from the cluster manager. Otherwise, this may be a configuration error. The data section should indicate the resource it is trying to action on (for example, the name of the resource or resource group that it is attempting to create or find).

Note: The configuration parameter requirements for DB2 HADR are different from those of the db2haicu tool. Due to this difference, if the HADR database pair is working perfectly, the db2haicu tool may still fail when creating resources in the cluster domain. Refer to Table 2 for details.

Common invalid inputs

In most cases, an obvious invalid user input will be rejected from the db2haicu interactive interface (for example, using a host name as a VIP). However, some configuration errors cannot be immediately detected, so the setup process may fail at the end with a generic message. You will need to examine the db2diag.log for further debugging.

Let's take a look at a few common areas that are prone to invalid inputs:

- Host name
- `HADR_REMOTE_INST`
- General naming format

Host name

The integrated HA solution relies on various sources for host name. As a result, you should verify that the formatting of all user inputs and system settings are consistent with one another.

To find out how TSA/RSCT views the host name, issue `lsrpnode`.

To find out how DB2 views the host name, issue `uname`. (**Note:** It is good practice to keep uname host name consistent with the output of host name.)

Listing 25 shows a few places where the host name is requested from users.

Listing 25. Host name requests

```
Create a unique name for the new domain:
ha
Nodes must now be added to the new domain.
How many cluster nodes will the domain ha contain?
2
Enter the host name of a machine to add to the domain:
host1
Enter the host name of a machine to add to the domain:
host2
db2haicu can now create a new domain containing the 2 machines that you specified.
If you choose not to create a domain now, db2haicu will exit.

Create the domain now? [1]
1. Yes
2. No
1
Creating domain ha in the cluster...
Creating domain ha in the cluster was successful.
```

For the HADR database, the host names are specified in `HADR_LOCAL_HOST` and `HADR_REMOTE_HOST` parameters. If an IP address is specified here, the db2haicu interactive tool will then prompt for the actual host name.

Listing 26. Prompt for host name

```
Setting a high availability configuration parameter for instance db2inst1 to TSA.
Adding DB2 database partition 0 to the cluster...
Adding DB2 database partition 0 to the cluster was successful.
Do you want to validate and automate HADR failover for the HADR database HADRDB? [1]
1. Yes
2. No
1
Adding HADR database HADRDB to the domain...
The cluster node 9.26.53.5 was not found in the domain. Please re-enter the host name.
host1
The cluster node 9.26.53.50 was not found in the domain. Please re-enter the host name.
host2
Adding HADR database HADRDB to the domain...
The HADR database HADRDB has been determined to be valid for high availability.
However, the database cannot be added to the cluster from this node because db2haicu
detected this node is the standby for the HADR database HADRDB.
Run db2haicu on the primary for the HADR database HADRDB to configure the database for
automated failover.
All cluster configurations have been completed successfully. db2haicu exiting...
```

The value specified in parameters `HADR_LOCAL_HOST` and `HADR_REMOTE_HOST` will be used as-is to generate the clustered resource names. If there is inconsistency in formatting between these values for host name sources, this will result in failure. Failure to keep this consistency will lead to db2haicu configuration failure. Best practice is to use canonical host names in all areas.

Listings 27 and 28 show error messages that may be seen in a situation where there is host name inconsistency.

Listing 27. Error message from db2haicu

```
The HADR database HADRDB cannot be added to the cluster because the standby instance
is not configured in the domain. Run db2haicu on the standby instance to configure it
into the cluster.
All cluster configurations have been completed successfully. db2haicu exiting ...
```

Listing 28. Error message from db2diag.log

```
2010-08-04-13.45.17.792165+720 E258375A757        LEVEL: Error
PID     : 23901                TID  : 2199089142032PROC : db2haicu
INSTANCE: db2inst3             NODE : 000
FUNCTION: DB2 Common, SQLHA APIs for DB2 HA Infrastructure, sqlhaAddResourceGroup,
          probe:300
MESSAGE : ECF=0x90000557=-1879046825=ECF_SQLHA_CLUSTER_ERROR
          Error reported from Cluster
DATA #1 : String, 35 bytes
Error during vendor call invocation
DATA #2 : unsigned integer, 4 bytes
46
DATA #3 : String, 45 bytes
db2_db2inst3_lildb202.nz.thenational.com_0-rg
DATA #4 : unsigned integer, 4 bytes
1
DATA #5 : unsigned integer, 8 bytes
```

```
1
DATA #6 : signed integer, 4 bytes
0
DATA #7 : String, 0 bytes
Object not dumped: Address: 0x00000000800D59FC Size: 0 Reason: Zero-length data
```

HADR_REMOTE_INST

Another source of error for HADR configuration is `HADR_REMOTE_INST`, which is case-sensitive for db2haicu.

Listing 29. Error message from db2diag.log

```
2010-08-05-09.16.57.317614-240 E49090A665         LEVEL: Error
PID     : 717032               TID  : 1           PROC : db2haicu
INSTANCE: db2inst1             NODE : 000
EDUID   : 1
FUNCTION: DB2 Common, SQLHA APIs for DB2 HA Infrastructure, sqlhaUICreateHADR, probe:895
RETCODE : ECF=0x9000056F=-1879046801=ECF_SQLHA_HADR_VALIDATION_FAILED
          The HADR DB failed validation before being added to the cluster
MESSAGE : Please verify that HADR_REMOTE_INST and HADR_REMOTE_HOST are correct
          and in the exact format and case as the Standby instance name and
          host name.
DATA #1 : String, 7 bytes
Db2inst1
DATA #2 : String, 10 bytes
host2

2010-08-05-09.16.57.317983-240 E49756A594         LEVEL: Error
PID     : 717032                TID  : 1          PROC : db2haicu
INSTANCE: db2inst1              NODE : 000
EDUID   : 1
FUNCTION: DB2 Common, SQLHA APIs for DB2 HA Infrastructure, sqlhaUICreateHADR, probe:900
RETCODE : ECF=0x9000056F=-1879046801=ECF_SQLHA_HADR_VALIDATION_FAILED
          The HADR DB failed validation before being added to the cluster
DATA #1 : String, 7 bytes
db2inst1
DATA #2 : String, 7 bytes
Db2inst1
DATA #3 : String, 10 bytes
host1
DATA #4 : String, 10 bytes
host2
DATA #5 : String, 7 bytes
HADRDB
```

General naming format

The instance name cannot contain an underscore (_) because it is used as a delimiter for the TSA/RSCT resource. It may lead to incorrect parsing. Improvements have been made in V95FP6 (IC62705) and V97FP2 (IC64856) to allow instance names containing underscores. However, it is advisable to steer away from such naming.

TSA/RSCT command failure

It is possible for db2haicu configuration to fail even if all inputs are valid. During the configuration process, db2haicu would make TSA/RSCT calls to create and register various resources. If the vendor call invocation returns with an error, this would lead to db2haicu exiting unsuccessfully.

To verify this, examine the db2diag.log for any error messages containing the "### --- ####-###" indicator. Listings 30 - 34 show typical error messages from db2haicu and db2diag.log to illustrate common failures.

- Common failure #1
- Common failure #2
- Common failure #3

Common failure #1

Listing 30. Typical error message from db2haicu

```
Create the domain now? [1]
1. Yes
2. No
1
Creating domain HA in the cluster ...
Creating domain failed. Refer to db2diag.log and the DB2 Information Center for details.
```

Listing 31. Typical error message from db2haicu

```
Adding DB2 database partition 0 to the cluster ...
There was an error with one of the issued cluster manager commands. Refer to db2diag.log
and the DB2 Information Center for details.
```

Listing 32. Typical error message in db2diag.log

```
2010-01-06-14.31.50.656997-420 E3426E903           LEVEL: Error
PID     : 3485                 TID  : 48008924417584PROC : db2haicu
INSTANCE: db2inst1             NODE : 000
FUNCTION: DB2 Common, SQLHA APIs for DB2 HA Infrastructure, sqlhaCreateCluster, probe:900
MESSAGE : ECF=0x90000544=-1879046844=ECF_SQLHA_CREATE_CLUSTER_FAILED
          Create cluster failed
DATA #1 : String, 24 bytes
Error creating HA Domain
DATA #2 : String, 6 bytes
UC4GHA
DATA #3 : unsigned integer, 4 bytes
2
DATA #4 : signed integer, 4 bytes
21
DATA #5 : String, 350 bytes
Line # : 8839---2632-044 The domain cannot be created due to the following errors that
         were detected while harvesting information from the target nodes:
         db200: 2610-441 Permission is denied to access the resource class specified
         in this command. Network Identity UNAUTHENT requires 's' permission for the
         resource class IBM.PeerDomain on node db200.
```

Solution: `preprpnode` was not run on each node. As root, issue TSA/RSCT command `preprpnode <host1> <host2>` on each node.

Common failure #2

Listing 33. Typical error message in db2diag.log

```
   2010-07-20-14.52.28.537943-240 E15731289A765      LEVEL: Error
   PID     : 974914               TID  : 1           PROC : db2haicu
```

```
   INSTANCE: db2inst1            NODE : 000
   EDUID   : 1
   FUNCTION: DB2 Common, SQLHA APIs for DB2 HA Infrastructure,
   sqlhaAddResource, probe:1600
   MESSAGE : ECF=0x90000542=-1879046846=ECF_SQLHA_CREATE_GROUP_FAILED
             Create group failed
   DATA #1 : String, 35 bytes
   Error during vendor call invocation
   DATA #2 : unsigned integer, 4 bytes
   0
   DATA #3 : String, 0 bytes
   Object not dumped: Address: 0x0000000110165890 Size: 0 Reason: Zero-
   length data
   DATA #4 : unsigned integer, 8 bytes
   1
   DATA #5 : signed integer, 4 bytes
   309
   DATA #6 : String, 86 bytes
   Line # : 7227---2621-309 Command not allowed as daemon does not have a
   valid license.
```

Solution: This is a TSA/RSCT licensing issue. Obtain the license from Passport Advantage and apply it using the `samlic` command.

Common failure #3

Listing 34. Typical error message in db2diag.log

```
2010-07-20-14.52.28.538365-240 E15732055A369      LEVEL: Error
PID     : 974914              TID  : 1           PROC : db2haicu
INSTANCE: db2inst1            NODE : 000
EDUID   : 1
FUNCTION: DB2 Common, SQLHA APIs for DB2 HA Infrastructure,
sqlhaCreateNetwork, probe:50
RETCODE : ECF=0x90000542=- RETCODE :
          ECF=0x90000542=-1879046846=ECF_SQLHA_CREATE_GROUP_FAILED
          Create group failed.
Line # : 6531---host1: 2661-011 The command specified for attribute MonitorCommand is
         NULL, not a absolute path, does not exist or has insufficient permissions to
         be run.
```

Solution: This indicates the sapolicies scripts are not found in the /usr/sbin/rsct/sapolicies/db2 location. See the technote "Error 'The command specified for attribute MonitorCommand is NULL' reported by db2haicu" (IBM, March 2010) for a resolution. (See Related topics for a link.)

For other cluster manager errors, refer to the RSCT information center (see Related topics for a link) or contact IBM Support.

Conclusion

After reading this tutorial, you should have a deeper understanding of how different pieces of the HA architecture relate to one another and how you can use the tools to debug issues with each component. This information is not only valuable in initial diagnostics, it also demystifies the components of the HA architecture, sets the proper expectations of how these components work together, and what each piece of diagnostic information really means from the DB2 and TSA/RSCT perspectives.

Related topics

- "Automated cluster controlled HADR configuration setup using the IBM DB2 high availability instance configuration utility" (developerWorks, August 2009): Learn how to configure a failover solution for IBM DB2 LUW, using the DB2 high availability disaster recovery (HADR) feature and the DB2 high availability instance configuration utility (db2haicu).
- Tivoli software information center: Find product documentation on IBM Tivoli System Automation for Multiplatforms.
- Reliable Scalable Cluster Technology (RSCT) information center: Find help and product documentation for Reliable Scalable Cluster Technology.
- IBM DB2 Database for Linux, UNIX, and Windows information center: Find information describing how to use the DB2 family of products and features.
- "Error 'The command specified for attribute MonitorCommand is NULL' reported by db2haicu" (IBM, March 2010): In this technote, learn how to resolve this potential error reported during automation setup.
- DB2 for Linux, UNIX, and Windows area on developerWorks: Get the resources you need to advance your skills in DB2.
- DB2 Express-C: Download DB2 Express-C, a no-charge version of DB2 Express database server for the community.
- DB2 for Linux, UNIX, and Windows: Download a free trial version of DB2 for Linux, UNIX, and Windows.