High Availability and Disaster Recovery Configuration Guide for IBM Security Manager 6.x/7.x

Table of Contents

Introduction

In order to ensure critical data is readily available or to reduce system downtime resulting from temporary or large-scale failures, many customers will have requirements to set up IBM Security Identity Manger (ISIM) in either a High Availability (HA) environment, a Disaster Recover (DR) environment or in both configurations.

This document discusses how to configure either ISIM 6.x or the ISIM 7.x Virtual Appliance (VA) for HA, DR or HA/DR environments. The deployment topology and configuration are consistent with current documentation for IBM Security Identity Manager (ISIM) 6.x /7.x. Both high availability and data replication features provided by the associated middleware are leveraged to provide a complete solution. This article covers the high availability configuration steps that were tested in Security Performance labs and describes fail-over scenarios for stand-by DB2 databases and stand-by Websphere Application Server (WAS) cluster systems. This document also details subject matter expert recommendations on configuring ISIM and its accompanying middleware in separate Data Centers for Disaster Recovery.

Disclaimer

The configuration and steps here are what the ISIM Development and Test team have currently tested based recommendations from DB2/WAS subject matter experts on how HA and DR should be implemented. Certain aspects of the HA/DR implementations have not yet been completely tested (log shipping, disc mirroring, replication consistency groups, HACMP, Solaris Clusters) .

Do not consider the following document to be the absolute authority on HA and DR for ISIM until completely tested!

Defining High Availability and Disaster Recovery

For the purposes of this document, the following definitions are used:

- **High Availability** (HA) is configuring ISIM and its middleware to reduce downtime within a data center.

- **Disaster Recovery** (DR) is configuring ISIM and middleware to reduce downtime across data centers. The two can be set up together or separately

Evaluating the Need for HA or DR

It is important to evaluate the actual need for HA and/or DR for ISIM vs the complexity of setting up and maintaining such a configuration.

If ISIM is critical path for a business process that cannot tolerate downtime, then it makes sense to configure ISIM for HA and likely DR. For example, you would set up HA and/or DR if you are using ISIM as the front-end for external users to reset their password to access billing information.

If ISIM is being used in a scenario where downtime can be tolerated, then setting up for HA or DR may not be worth the investment in configuration time and system complexity. For example, you might not choose to implement HA or DR if ISIM is only managing accounts for a small set of internal users. In this scenario perhaps the downtime to repair a damaged ISIM installation from backups costs less than the hardware and administrative overhead required for an HA environment.

Important: Neither HA or DR configurations remove the need for regular system backups. Due to the need for HA and DR environments to have consistent data, they are vulnerable to data corruption issues just like non-HA/DR environments

ISIM 6.x HA/DR Features and Support

ISIM 6.0 FP 4 / ISIM7.0.1.4 and higher allow a cluster node to function as a hot-standby node. In a hot-standby server cluster, only the primary nodes perform useful work. The standby node is powered on but cannot run scheduled jobs, reconciliations or participate in load-balancing work.

WAS 8.5 includes a messaging engine that has been enhanced to be able to stop gracefully in the event of a database failure. Previously the failure would stop the entire JVM. Once stopped, it can then be automatically re-enabled. The messaging engine utilizes multiple cores to provide quicker messaging engine start-up when large numbers of messages and destinations are present.

DB2 11.1 HADR supports up to three standby databases. You can have two auxiliary standby databases in a second data center while keeping the primary and principle standby database in the primary data center.

DB2 11.1 log spooling allows log data sent from the primary database to be written to disk on the standby database if the log buffer becomes full.

Configuring for High Availability (HA)

The goal of high availability is to minimize the impact of a single point of failure. The figure below shows a high level view of a ISIM system set up for high availability.

Multiple WAS nodes in a single cluster run the ISIM application. A load balancer routes the incoming traffic to the back-end nodes on a round-robin basis. The load balancer is usually the IBM HTTP Server that is included with WAS. For HA, the load balancer needs to detect when a node is not available and stop routing traffic to that node. The WAS HTTP plug-in for IBM HTTP Server serves this function. To ensure that the load balancer is not a single point of failure, configure at least two of them.

In an HA environment all of the ISIM nodes in the cluster should be active and accepting traffic. There is no restriction to prevent all nodes from being active. There is no benefit to having a cold-standby or warm-standby node.

Tip: There are performance advantages when you run the IBM HTTP Server as the load balancer. Consult the ISIM Performance Tuning Guide for more information.

Figure 1 shows a typical topology.

- An HTTP server accepts incoming traffic and is supported by a standby server,
- In the WAS cluster, each cluster node and the deployment manager (DM) are installed on a separate system.
- ISDI can be co-located on the system with the WAS node.
- A load balancer server with a standby server manages traffic to the ISDS peer masters.
- Two identical DB2 servers are installed on separate systems. One acts as the primary database server and the other one as the standby.

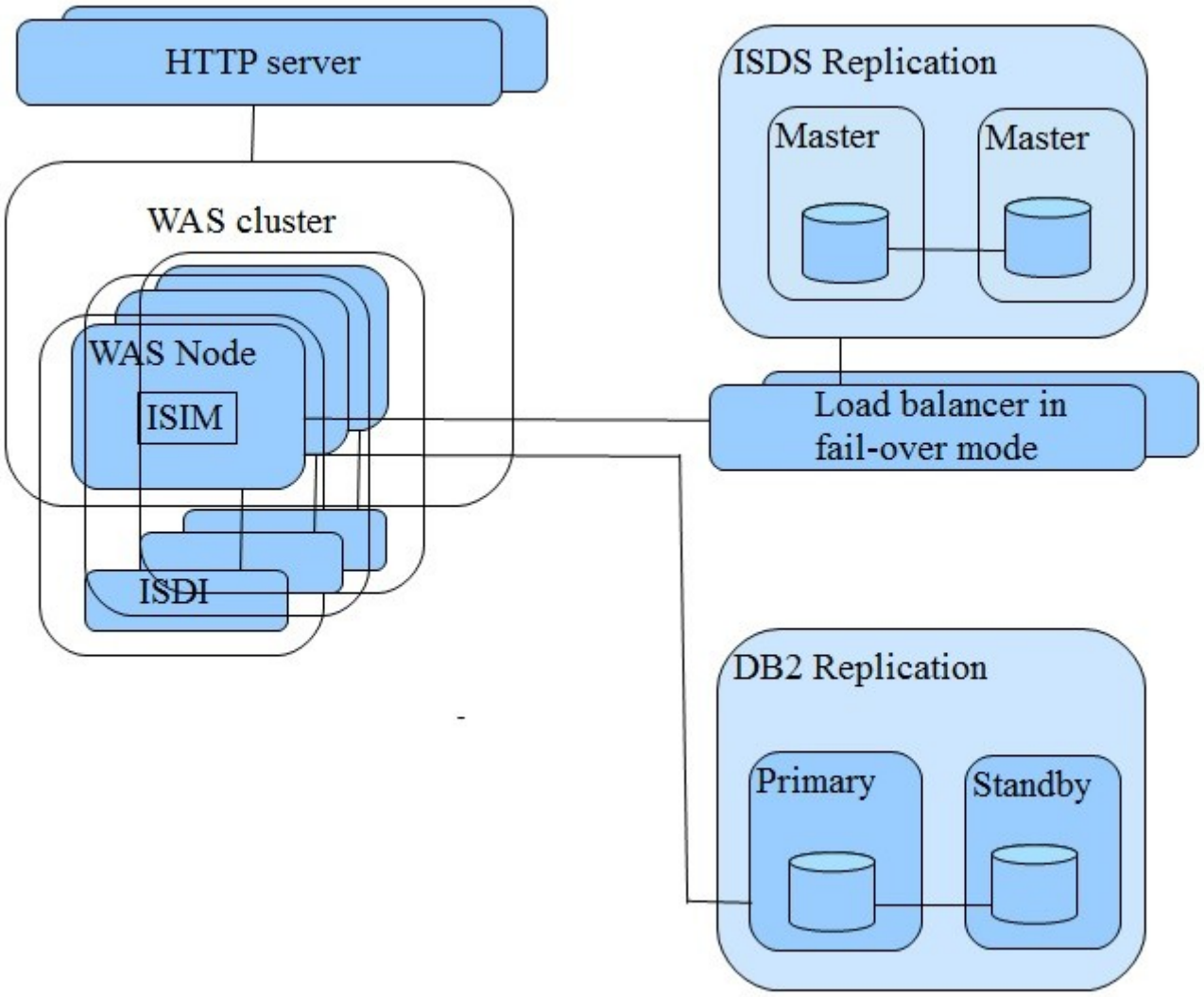


Figure 1

Summary of H/A Configuration Steps

High Availability for ISIM version 6/7 Configuration

- Configure DB2 to support active standby
- Configure IBM Security Directory Server (ISDS) replication
- Configure WAS nodes
- Configure the WAS messaging server
- Configure the data source to support automatic client rerouting
- Configure a hot-standby node (optional)
- Configure DB2 to support active standby

ISIM Supports Two HA DB2 Configurations

DB2 HA Configuration 1

Use two DB2 instances, one active and one standby, each with a separate data store. Have the data replicated between them using the DB2 HADR feature. **This configuration has been tested by the ISIM Development Team.**

DB2 HA Configuration 2

Use two DB2 instances, one active and one passive. Have them share a common data store. Failover uses HACMP or Solaris Clusters. **This has NOT been tested by the ISIM Development Team.**

The High Availability Disaster Recovery (HADR) feature in IBM DB2 replicates data from a primary database to a standby database. It allows automatic failover and fail-back between two independent DB2 instances. One instances is called the primary. The other is called the alternate.

Each DB2 instance has its own back-end data store. They do not share disks. DB2 replicates the data between the instances. When a failure condition occurs on the primary system, DB2 automatically fails over and reroutes activity to the alternate system. (A sample script, auto_takeover_pub.ksh, is provided that can be used for this purpose).

When the JDBC driver used by WAS makes an initial connection to a HADR-enabled system, the DB2 server tells the JDBC driver what node is configured for failover. When a failure or manual switch occurs, the JDBC driver detects the change, connects to the alternate system, and resumes normal operation. A similar process happens when a fail-back occurs.

Prerequisites

Related systems must meet prerequisites for the hardware, operating system and database system in order to enable HADR.

(See HADR system requirements in the Knowledge Center for DB2 11.1 for Linux, UNIX, and Windows.)

Key Requirements

- The database name must be the same on the primary and standby databases
- TCP/IP communications must be available between the primary and standby databases.
- The database server on primary and standby databases must be at the same version, level, and bit size (32 or 64 bits).
- The operating system on the hosts for the primary and standby databases must be at the same version and level.

Configure DB2 for HA/DR

After installing DB2 on the primary system and standby system, configure HADR using the following steps

Start the DB2 Server on the Primary and Standby Systems.

If it has not already been done, run the ISIM middleware configuration utility on the primary database system.
(See IBM Security Identity Manager 6.0 FP6 Knowledge Center for details on the middlware configuration utility)

Specify the TCP/IP Communication Ports.

On Unix or Linux system, it is in /etc/services.
On Windows, it is c:\windows\system32\drivers\etc\services. Add two lines to the file on both systems to define the ports.

Example:

- db2_hadr_1 51000/tcp
- db2_hadr_2 51001/tcp

Each line specifies a connection service name, port number, and communications protocol. Make sure the port numbers are not used by other processes on the system.

Link the Primary Database System to the Standby Database System.

Update the following parameters:

db2 update db cfg for <DBNAME> using LOGARCHMETH1 LOGRETAIN

db2 update db cfg for <DBNAME> using HADR_LOCAL_HOST <IP ADDRESS OF PRIM>

db2 update db cfg for <DBNAME> using HADR_LOCAL_SVC <PORT # on PRIM>

db2 update db cfg for <DBNAME> using HADR_REMOTE_HOST <IP ADDRESS OF STNDBY>

db2 update db cfg for <DBNAME> using HADR_REMOTE_SVC <PORT # on STNDBY>

db2 update db cfg for <DBNAME> using HADR_REMOTE_INST <INSTNAME OF STNDBY>

db2 update db cfg for <DBNAME> using LOGINDEXBUILD ON

db2 update alternate server for database <DBNAME> using hostname <IP ADDRESS OF STNDBY> port <PORT # on STNDBY>

Set Synchronization Level

The level of synchronization between the primary and standby database is determined by the synchronization mode parameter. By default, the mode is NEARSYNC. You can use the command to update it to a different mode.

(See Synchronizing the primary and secondary databases in the Knowledge Center for DB2 11.1 for Linux, UNIX, and Windows to determine which mode you want to choose in your high availability solution. If you are using another version, consult the Knowledge Center for your version.)

db2 update db cfg for <DBNAME> using HADR_SYNCMODE SYNC

Back up the Primary Database.

db2 backup database <DBNAME>

Transfer the Backup Image from the Primary System to the Standby System.

Restore the Database on the Standby System.

db2 restore database <DBNAME>

Link the Standby Database with the Primary Database System.

db2 update db cfg for <DBNAME> using HADR_LOCAL_HOST <IP ADDRESS ON STANDBY>

db2 update db cfg for <DBNAME> using HADR_LOCAL_SVC <PORT # ON STANDBY>

db2 update db cfg for <DBNAME> using HADR_REMOTE_HOST <IP ADDRESS ON PRIM>

db2 update db cfg for <DBNAME> using HADR_REMOTE_SVC <PORT # ON PRIM>

db2 update db cfg for <DBNAME> using HADR_REMOTE_INST <INSTNAME ON PRIM>

Start the HADR on Both Databases.

Start the standby system first
db2 start hadr on database <DBNAME> as standby

Then the primary system
db2 start hadr on database <DBNAME> as primary

Verify HADR is Configured Properly on Both Database Systems.

db2pd -db <DBNAME> -hadr

Note:
In the command output, you should see HADR_ROLE=PRIMARY for the primary system and HADR_ROLE=STANDBY for the standby system.
You should also see HADR_STATE=PEER and HADR_CONNECT_STATUS=CONNECTED for both.

Create a heartbeat mechanism to monitor primary database and trigger failover if the primary database is down.

IBM Tivoli System Automation is the recommend cluster manager for automating HADR failover (see https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/DB2HADR/page/cluster%20managers).

Use the provided sample script auto_takeover_pub.ksh to test failover. The script periodically checks the primary server. When it detects that the primary has failed, it runs the following command:

db2 takeover hadr on db $DBNAME by force

The command promotes the standby database to be the primary.

Based on testing in IBM Performance labs, when the first DB connection that WAS attempts during fail-over is in progress, this will result in a timeout exception indicating that pending workflow threads have timed out.

This exception will result in the remainder of the JDBC connection pool being purged (per the default configuration) and the next JDBC request will be routed to the alternate server. Next is a hung thread exception from the SIBJMSRAThreadPool that will bubble up to ITIM. These hung threads (according to testing in IBM Performance labs) will be retried depending on what ITIM component was initiating the JDBC query.

Out of the box, the JDBC driver will not persist the fail-over information between WAS restarts. For example, consider the case where a failure occurs and DB2 fails over to the backup system. The system will continue to operate as expected while WAS is up. If WAS is brought down and back up again, the JDBC driver will attempt to connect to the failed DB2 instance. Because the initial connection fails, the JDBC driver is unaware of the DB2 fail-over system and is unable to connect to it. The appendix of this WAS DB2 HADR article talks about how to persist the connection information across WAS restarts.

Configure IBM Security Directory Server (ISDS) Replication

IBM Security Directory Server stores users, accounts, and organizational data. To provide failover capability, install it as a cluster using a IBM Security Directory server on each of two separate systems. Data is synchronized between the two servers.

Only one server can be active at a time. The other must be in a failover-only mode. It must be running but not accessed by ISIM. Use a failover-only load balancer between ISIM and the IBM Security Directory Server nodes. The load balancer routes traffic to one directory server until that server fails. The load balancer then routes the traffic to the other active node.

Keeping in mind that only one back-end directory server should be active at a time then when

failover occurs, the load balancer should terminate any other active connections to the failed server.

For ISDS 6.4 and higher, run the ldapreplcfg command to configure ISDS replication: https://www.ibm.com/support/knowledgecenter/SSVJJU_6.4.0/com.ibm.IBMDS.doc_6.4/ds_ag_srv_adm_repl_topology_config_tool.html

The following steps illustrate how to configure a peer-to-peer replication. In the example, the server host names are hadrsds1 and hadrsds2. The LDAP suffix declaration is dc=com. You must run the ISIM middleware configuration utility tool for each server. It creates the instances to use the same user password, LDAP suffix and encryption seed.

(For details on running middleware configuration utility, see Running the middleware configuration utility in the IBM Knowledge Center for IBM Security Identity Manager 6.0.x)

Edit ibmslapd.conf file on both ISDS servers.

Find ibm-slapdServerId: and set it as in the following example.

On hadrsds1:
ibm-slapdServerId: peer1

On hadrsds2:
ibm-slapdServerId: peer2

Start both ISDS servers

ibmslapd -I <instance name>

Create file /tmp/rep1.ldif on each ISDS system

Using the following content:

```
dn: cn=Master Server, cn=configuration
objectclass: ibm-slapdReplication
cn: Master Server
ibm-slapdMasterDN: cn=any
ibm-slapdMasterPW: secret
```

Add the entries into ibmslapd.conf

idsldapadd -D cn=root -w <password> -i /temp/rep1.ldif

Stop the ISDS servers

ibmslapd -I itimldap -k

Create file /tmp/rep2.ldif on each ISDS system.

Using the following content:

dn: cn=replication,cn=IBMpolicies
objectclass: container
dn: cn=simple,cn=replication,cn=IBMpolicies
objectclass:ibm-replicationCredentialsSimple
cn:simple
replicaBindDN:cn=any
replicaCredentials:secret
description:Bind method of the peer master (hadrsds1) to the peer master(hadrsds2)

Add the entries on each ISDS system

(Ignore the error message about one of the entries already in existence.)

idsldif2db -r no -i /temp/rep2.ldif -I <instance name>

Restart the servers.

Create file /tmp/dccom.ldif on each ISDS System

Using the following content

dn: dc=com
objectclass: top
objectclass: domain
dc: com

Add the dc=com suffix object.

ldapadd -h localhost -p 389 -D cn=root -w <password> -c -v -f /tmp/dccom.ldif

Use the ISDS Web Admin tool to connect to each ISDS server.

- Log in as an administrative user.
- Select Directory management > Manage entries
- Click the dc=com check-box and select Action > Edit ACL
- Select Non-filtered ACLs > Add
- Subject DN> Enter cn=root
- Click Grant all
- Click Ok
- Click Owners

- Subject DN> Enter cn=root
- Click Add
- Click Ok
- Click Logout

Create file /tmp/dccontext.ldif on each ISDS System

Using the following content

```
dn: dc=com
changetype: modify
add: objectclass
objectclass: ibm-replicationcontext
```

Add the context to dc=com on each ISDS System

idsldapmodify -D cn=root -w <password> -i /tmp/dccontext.ldif

Stop the ISDS servers

ibmslapd -I <instance name> -k

Create the file /tmp/rep3.ldif on each ISDS System

Using the following content

```
dn: ibm-replicaGroup=default,dc=com
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default

dn: ibm-replicaServerId=peer1,ibm-replicaGroup=default,dc=com
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: peer1
ibm-replicationServerIsMaster: true
cn: peer1
description: peer1 (peer master) ibm-replicaSubentry

dn: ibm-replicaServerId=peer2,ibm-replicaGroup=default,dc=com
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: peer2
ibm-replicationServerIsMaster: true
cn: peer2
```

```
description: peer2 (peer master) ibm-replicaSubentry

#peer1 to peer2 agreement
dn: cn=peer2,ibm-replicaServerId=peer1, ibm-replicaGroup=default,dc=com
objectclass: top
objectclass: ibm-replicationAgreement
cn: peer2
ibm-replicaConsumerId: peer2
ibm-replicaUrl: ldap://hadrsds2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: peer1(master) to peer2(master) agreement

#peer2 to peer1 agreement
dn: cn=peer1,ibm-replicaServerId=peer2, ibm-replicaGroup=default,dc=com
objectclass: top
objectclass: ibm-replicationAgreement
cn: peer1
ibm-replicaConsumerId: peer1
ibm-replicaUrl: ldap://hadrsds1:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: peer2(master) to peer1(master) agreement

cn=ibmpolicies
dn: ibm-replicaGroup=default,cn=ibmpolicies
objectclass: top
objectclass: ibm-replicaGroup
ibm-replicaGroup: default

dn: ibm-replicaServerId=peer1,ibm-replicaGroup=default,cn=ibmpolicies
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: peer1
ibm-replicationServerIsMaster: true
cn: peer1
description: peer1 (peer master) ibm-replicaSubentry

dn: ibm-replicaServerId=peer2,ibm-replicaGroup=default,cn=ibmpolicies
objectclass: top
objectclass: ibm-replicaSubentry
ibm-replicaServerId: peer2
ibm-replicationServerIsMaster: true
cn: peer2
description: peer2 (peer master) ibm-replicaSubentry

#peer1 to peer2 agreement
dn: cn=peer2,ibm-replicaServerId=peer1, ibm-replicaGroup=default,cn=ibmpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: peer2
ibm-replicaConsumerId: peer2
ibm-replicaUrl: ldap://hadrsds2:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: peer1(master) to peer2(master) agreement
```

#peer2 to peer1 agreement
dn: cn=peer1,ibm-replicaServerId=peer2, ibm-replicaGroup=default,cn=ibmpolicies
objectclass: top
objectclass: ibm-replicationAgreement
cn: peer1
ibm-replicaConsumerId: peer1
ibm-replicaUrl: ldap://hadrsds1:389
ibm-replicaCredentialsDN: cn=simple,cn=replication,cn=IBMpolicies
description: peer2(master) to peer1(master) agreement

Add the replica agreement entries on each ISDS System

idsldif2db -r no -i /tmp/rep3.ldif -I <instance_name>

Start the ISDS servers

ibmslapd -I <instance_name>

Install ISIM and point to either ISDS server as the target.

Give the peer ISDS server a few minutes after the ISIM install completes to catch up on all the replication entries.

If there is existing data under suffix dc=com before replication is configured, the data should be exported.

idsdb2ldif -I <INSTANCE_NAME> -o <OUTPUT FILE>

Transfer the exported ldif file to the destination server and import it into the ISDS instance.

idsldif2db -I <INSTANCE_NAME> -i <INPUT_FILE>

Configuring the WAS Node

Configure the following areas on the WAS node:

- Messaging engine
- Data source
- Hot-standby WAS node (optional)

Configuring the Messaging Engine

Configure the messaging engine and application server to prepare for events that are in process when the connection to the data store is lost.

If the messaging engine connection is lost while it is running and while it has established exclusive locks on its data store, incomplete transactions are processed by the stand-by data source only after the messaging engine restarts.

In order to ensure that new and in-flight transactions are processed after failover, each messaging engine must be configured so that JDBC connections failover to the standby DB2 server when the connection is lost.
Use the WebSphere Administrative Console to set 'sib.msgstore.jdbcFailoverOnDBConnectionLoss' to 'true'.

If it is set to 'false', the WebSphere server that houses the messaging server does not restart. New and pending ISIM requests are not processed.

Configure the messaging engine using the following procedure:

- Select Service integration > Buses > bus_name > Messaging engines > engine_name > Additional Properties > Custom properties.
- Click New.
- Type sib.msgstore.jdbcFailoverOnDBConnectionLoss in the Name field and true in the Value field.
- Click OK.
- Save the changes to the master configuration.
- Restart the application server.

In a clustered environment, repeat the previous steps for every messaging engine in the cluster.

To see a list of messaging engines, go to Buses > isim_bus > Messaging engines.

For example, in a two node cluster with application and messaging servers on each , there are three messaging engines that needed to be modified:

- App_Cluster.000-isim_bus
- App_Cluster.001-isim_bus
- Msg_Cluster.000-isim_bus

Configuring the Data Source for Automatic Client Rerouting

On a single-node WebSphere application server, ISIM has two data sources:

ISIM Data Source
Defines connection to the data store that stores the ISIM transactional, historical, and access catalog data

ISIM Bus Data Source
Defines connection to the SIB store that stores the persistent data used by messaging engine.

On a WebSphere cluster, ISIM has one more data source: ISIM Bus Shared DataSource.

To facilitate failover so that Websphere can reroute new and pending requests to the new standby database, configure each data source:

- ISIM Data Source
- ISIM Bus Data Source,
- ISIM Bus Shared DataSource

In Websphere Administrative Console

- Select Data source properties panel > Resources > JDBC > Data sources > ISIM Bus DataSource > WebSphere Application Server data source properties

- Set values for Alternate server name and Alternate port number. Use the hostname and port_number of the configured DB2 standby server.

- Under DB2 automatic client reroute options, specify the Alternate server name and Alternate port number.

Configuring a Hot-Standby WAS Node (optional)

Some customers require an additional layer of protection. For example, they may want to deploy additional nodes as standby nodes on a system located at a different site. If the primary site servers are down, the standby system takes over. The hot-standby node is always running but does not participate in reconciliation or shared workflow processes. Customers may need to test the functionality on the hot-standby node periodically.

To configure a node to be the hot-standby node, you can set enrole.appServer.standby to be true in ISIM's enroleStartup.properties file. This change requires that you restart the WAS server. The setting disables the SharedWorkflow, RemoteServices, and PolicyAnalysis message listeners and the scheduler on the node. Once the WAS node is restarted, you can see a message “"ISIM Server started in standby mode” in ISIM trace log. You can directly access ISIM on this node to perform ISIM operations. The requests should all be processed. However, all the scheduled activities, shared workflow, or policy analysis related activities are actually processed in the peer node. If the peer node is down, those activities stay in the queue and the request stays in pending status. Later, when the peer node is up, those activities are processed, the request is fulfilled.

The configuration can be further fine grained to disable one or more listeners by specifying the listener name in property enrole.appServer.standby.inactiveMessageListeners.
Valid values:

- AdhocSync
- MailServices
- Migration
- PolicyAnalysis
- RemoteServices
- SharedWorkflow

HACMP and Solaris Clusters

HACMP and Solaris Clusters allow for scripted fail-over of software running on two different

servers. In this configuration there are two separate DB2 instances each on their own physical servers but accessing the same filesystem or SAN. Both instances are configured to use the same tablespace containers and log paths but only one system is active at a time. When a failure is detected, the failed system is shut down and the passive system is started, takes over the IP address of the failed system, and starts the DB2 instance.

This configuration relies on redundancy within the underlying SAN to remove it as a single point of failure.

Note that this configuration is not tested by ISIM development team.

Failover Scenarios

This section walks you through failover scenarios and demonstrates the messages that you receive during the process.

DB2 Failover

The primary database server can go offline for a variety of reasons, including hardware failure or a network issue. Websphere Application server then can no longer connect to the primary database. When ISIM is configured to use automatic client reroute and DB2 is configured to fail over to the standby, the disruption to the running application is minimized.

This section traces how ISIM responds when the primary database fails over to the standby database.

1. When the standby database fails to connect to the primary database, Db2diag.log on standby database shows a message like the following.

```
2015-08-26-14.37.32.456718-300 I667291E522          LEVEL: Severe
PID     : 5104               TID : 140637873628928 PROC : db2sysc 0
INSTANCE: db2admin           NODE : 000           DB   : ITIMDB
HOSTNAME: qddb2s
EDUID   : 48                 EDUNAME: db2hadrs.0.0 (ITIMDB) 0
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrEduAcceptEvent, probe:20280
MESSAGE : ZRC=0x810F0077=-2129723273=SQLO_COMM_ERR_EHOSTUNREACH
          "No route to host"
DATA #1 : <preformatted>
Failed to connect to primary.
```

2. The takeover by force command is issued by TSA or other heartbeat mechanism.

```
2015-08-26-14.37.46.010364-300 E673927E503          LEVEL: Event
PID     : 5104               TID : 140637978486528 PROC : db2sysc 0
INSTANCE: db2admin           NODE : 000           DB   : ITIMDB
APPHDL  : 0-80               APPID: *LOCAL.db2admin.150826193746
AUTHID  : DB2ADMIN           HOSTNAME: qddb2s
EDUID   : 23                 EDUNAME: db2agent (ITIMDB) 0
FUNCTION: DB2 UDB, base sys utilities, sqeDBMgr::StartUsingLocalDatabase, probe:13
START   : Received TAKEOVER HADR command.
```

3. Standby database starts to take over. It switches its role from standby to primary. DB2 runs the backward phase of database rollforward recovery. In-doubt transactions may be detected at this stage and rolled back.

2015-08-26-14.37.46.078639-300 I675401E437 LEVEL: Info
PID : 5104 TID : 140637873628928 PROC : db2sysc 0
INSTANCE: db2admin NODE : 000 DB : ITIMDB
HOSTNAME: qddb2s
EDUID : 48 EDUNAME: db2hadrs.0.0 (ITIMDB) 0
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrStbyTkHandleInitialRequest, probe:46000
MESSAGE : Standby has initiated a takeover by force.

4. When the standby database completes its takeover, the database is open for connections. At this point, the automatic client reroute can establish a successful connection to the standby system, which now the primary. Db2diag.log on standby database shows messages like this.

2015-08-26-14.37.46.746362-300 I713077E435 LEVEL: Info
PID : 5104 TID : 140637873628928 PROC : db2sysc 0
INSTANCE: db2admin NODE : 000 DB : ITIMDB
HOSTNAME: qddb2s
EDUID : 48 EDUNAME: db2hadrp.0.1 (ITIMDB) 0
FUNCTION: DB2 UDB, High Availability Disaster Recovery, hdrStbyTkHandleDoneDrain, probe:46840
MESSAGE : Standby has completed takeover (now primary).

5. When the JDBC driver detects that a database has failed over, it triggers an exception, com.ibm.db2.jcc.am.ClientRerouteException, which is mapped to com.ibm.websphere.ce.cm.StateConnectionException and propagated to the client application. The WAS SystemOut.log shows a message like the following:

[8/26/15 14:38:51:861 CDT] 000000df ConnectionEve A J2CA0056I: The Connection Manager received a fatal connection error from the Resource Adapter for resource enroleDataSource. The exception is: com.ibm.db2.jcc.am.ClientRerouteException: [jcc][t4][2027][11212][3.64.82] A connection failed but has been re-established. The host name or IP address is "qddb2s.rtp.raleigh.ibm.com" and the service name or port number is 50,002.

6. When the messaging engine loses its connection to the data store, the HAManager stops the messaging engine and restarts it when the database connection is successful. It produces messages like the following.

[8/26/15 14:51:28:138 CDT] 00000038 SibMessage E [itim_bus:qdisimNode01.server1-itim_bus] CWSID0046E: Messaging engine qdisimNode01.server1-itim_bus detected an error and cannot continue to run in this server.

[8/26/15 14:51:28:138 CDT] 00000038 SibMessage I [itim_bus:qdisimNode01.server1-itim_bus] CWSID0016I: Messaging engine qdisimNode01.server1-itim_bus is in state Stopping.

[8/26/15 14:51:48:063 CDT] 00000038 SibMessage I [itim_bus:qdisimNode01.server1-itim_bus] CWSID0016I: Messaging engine qdisimNode01.server1-itim_bus is in state Stopped.

[8/26/15 14:52:18:572 CDT] 000001ae SibMessage I [itim_bus:qdisimNode01.server1-itim_bus]

CWSID0016I: Messaging engine qdisimNode01.server1-itim_bus is in state Starting.

[8/26/15 14:53:05:786 CDT] 000001ae SibMessage I [itim_bus:qdisimNode01.server1-itim_bus] CWSID0016I: Messaging engine qdisimNode01.server1-itim_bus is in state Started.

The message bean has a retry mechanism. If a transaction error occurs, the container attempts to redeliver the message until either the message is delivered or it reaches the maximum number of retries. This process is performed for all container-managed beans.

<LogText><![CDATA[Exception processing entry: uid=PrimerUser782: CTGIMD012E An error occurred submitting the add request for the uid=PrimerUser782 entry.
Error: standardException.remoteException; java.rmi.RemoteException: ; nested exception is:
com.ibm.itim.workflow.engine.AsyncProcessingFailure: standardException.JMSException;
CWSIA0241E: An exception was received during the call to the method JmsManagedConnectionFactoryImpl.createConnection:
com.ibm.websphere.sib.exception.SIResourceException: CWSIT0019E: No suitable messaging engine is available on bus itim_bus that matched the specified connection properties {multicastInterface=none, connectionProximity=Server, targetSignificance=Required, subscriptionProtocol=Unicast, targetType=BusMember, busName=itim_bus}. Reason for failure: CWSIT0104E: The client attempted to connect to the messaging engine qdisimNode01.server1-itim_bus on bus itim_bus but the connection could not be created because the messaging engine is not started..]]></LogText>
<Source FileName="com.ibm.itim.remoteservices.ejb.mediation.PersonRequestHandler" Method="addOrModifyPerson"/>

Summary

When a request is processed during DB2 fail-over, the request typically takes a longer time to complete. In some cases, the request final status may be warning instead of success. However, all the messages for the request are eventually processed successfully.

DB2 - Fail-back to Original Configuration

When failover occurs, the standby DB2 server is promoted to primary. Furthermore, the old primary is then demoted to standby. In order to fail-back to the original configuration, these steps need to be taken:

- Stop all WebSphere Nodes in cluster
- Take a backup of current primary DB server
- Restore secondary using backup of primary using "without rolling forward" clause
- Start HA-DR on secondary
- Start HA-DR on primary
- Verify HA-DR connectivity (using hadrstatus_pub.sh)
- Standby is now Primary. Primary is now standby (back to original configuration)

- Login to ISIM and verify new user is still present

WebSphere Application Server Node Failover

The WAS configuration with ISIM includes an application cluster and a messaging cluster. Each server on the application cluster has its own local messaging engine. All the members in the messaging cluster share a messaging engine.

ISIM uses point-to-point asynchronous messaging to send or receive data as messages between internal modules. Each local messaging engine has a set of default local queue points that hold the same type of messages waiting to be consumed.

There are seven local messaging engine queues:

1. Adhoc sync
2. Workflow
3. Partitioning service
4. Remote services
5. Remote pending
6. Mail services
7. Import and export

There are three shared messaging engine queues:

1. Shared workflow
2. Policy analysis
3. Policy simulation

All messages in ISIM are persistent messages. They are stored within the ISIM database in SIB-specific tables. When a messaging engine in the SIB connects to the SIB tables in the database it locks the SIBOWNER table to ensure it has exclusive access to the datastore. This prevents another messaging engine from picking up messages specific to a given topic or queue and allows the engine to detect backend failures.

The reliability level is “assured persistent” for all ISIM message types except policy simulation. The “assured persistent” reliability level guarantees that no messages are lost during server failure. If the server fails when a message is in the local queue, the message will stay in the queue of the failed server and is not processed until the server is up and running again.

The shared messaging engine has a failover mechanism. When a message is in the shared queue during the server failure, the message is processed on the failover server in the messaging cluster.

You can simulate failover by stopping the application server and cluster server on one node

during reconciliation.

When the application server receives the stop command, it performs a sequence of actions, including:

- shutting down TCP channels
- shutting down the XD JMS message pacing controller
- shutting down each running application.

It also stops its messaging engine and the connection for its messaging engine to the messaging engine of the cluster server.

The following messages are typically seen in WebSphere SystemOut log file during failover. They show a DSML reconciliation that started at September 18, 2015 8:56:49 AM to bring in 1K new user records to Identity Manager. Two minutes later, failover occurred. Several transactions were pending on “Enforce Policy for Person”. After the failover was complete, the rest of new user workflow processes were complete successfully. For those pending new user workflow processes that were processed during failover, they stayed in pending stage. The whole reconciliation request status is pending.

[9/18/15 8:58:19:064 CDT] 0000004c ServerCollabo A   WSVR0023I: Server hadrdmn1 is stopping
[9/18/15 8:58:26:113 CDT] 00000030 SibMessage   I   [itim_bus:hadrdmn1.000-itim_bus] CWSIT0029I: The connection for messaging engine hadrdmn1.000-itim_bus in bus itim_bus to messaging engine mc_hadrdmn1.000-itim_bus stopped.
[9/18/15 8:58:27:468 CDT] 0000004c SibMessage   I   [itim_bus:hadrdmn1.000-itim_bus] CWSID0016I: Messaging engine hadrdmn1.000-itim_bus is in state Stopping.
[9/18/15 8:58:27:494 CDT] 0000004c SibMessage   I   [itim_bus:hadrdmn1.000-itim_bus] CWSIT0029I: The connection for messaging engine hadrdmn1.000-itim_bus in bus itim_bus to messaging engine hadrdmn1.001-itim_bus stopped.
[9/18/15 8:58:27:622 CDT] 0000004c SibMessage   I   [itim_bus:hadrdmn1.000-itim_bus] CWSID0016I: Messaging engine hadrdmn1.000-itim_bus is in state Stopped.
[9/18/15 8:58:27:801 CDT] 0000004c ServerCollabo A   WSVR0024I: Server hadrdmn1 stopped

On the messaging server of the same node, the same shutdown sequence is followed.

[9/18/15 8:58:20:482 CDT] 0000004a ServerCollabo A   WSVR0023I: Server mc_hadrdmn1 is stopping
[9/18/15 8:58:26:108 CDT] 0000004a SibMessage   I   [itim_bus:mc_hadrdmn1.000-itim_bus] CWSIT0029I: The connection for messaging engine mc_hadrdmn1.000-itim_bus in bus itim_bus to messaging engine hadrdmn1.001-itim_bus stopped.
[9/18/15 8:58:26:111 CDT] 0000004a SibMessage   I   [itim_bus:mc_hadrdmn1.000-itim_bus] CWSIT0029I: The connection for messaging engine mc_hadrdmn1.000-itim_bus in bus itim_bus to messaging engine hadrdmn1.000-itim_bus stopped.
[9/18/15 8:58:26:429 CDT] 0000004a SibMessage   I   [itim_bus:mc_hadrdmn1.000-itim_bus] CWSID0016I: Messaging engine mc_hadrdmn1.000-itim_bus is in state Stopped.
[9/18/15 8:58:26:975 CDT] 0000004a ServerCollabo A   WSVR0024I: Server mc_hadrdmn1 stopped

On the second node, the messaging server detected that the application server and messaging

server on the first node stopped. It restarted the messaging engine, which requires a sequence of actions that includes

- Connecting to the messaging database
- Obtaining an exclusive lock on the data store
- Reconciling WCCM destination and link configuration.

After the messaging engine restarts, it establishes the connection to the messaging engine on the second application server.

[9/18/15 8:58:35:841 CDT] 00000053 NGUtil$Server I ASND0003I: Detected server mc_hadrdmn1 stopped on node hadrdmn1Node01
[9/18/15 8:58:36:972 CDT] 00000053 NGUtil$Server I ASND0003I: Detected server hadrdmn1 stopped on node hadrdmn1Node01
[9/18/15 8:58:37:884 CDT] 000001eb SibMessage I [itim_bus:mc_hadrdmn1.000-itim_bus] CWSID0016I: Messaging engine mc_hadrdmn1.000-itim_bus is in state Starting.
[9/18/15 8:59:05:195 CDT] 000001eb SibMessage I [itim_bus:mc_hadrdmn1.000-itim_bus] CWSID0016I: Messaging engine mc_hadrdmn1.000-itim_bus is in state Started.
[9/18/15 8:59:08:534 CDT] 0000020e SibMessage I [:] CWSIT0028I: The connection for messaging engine mc_hadrdmn1.000-itim_bus in bus itim_bus to messaging engine hadrdmn1.001-itim_bus started.

When the messaging server on the first node stops, the messaging engine on the application server of the second node loses the connection to the messaging engine on the messaging server of the first node. Transaction errors and warnings are produced until the messaging engine fails over to the second node.

[9/18/15 8:58:35:387 CDT] 0000002f SibMessage I [itim_bus:hadrdmn1.001-itim_bus] CWSIT0029I: The connection for messaging engine hadrdmn1.001-itim_bus in bus itim_bus to messaging engine mc_hadrdmn1.000-itim_bus stopped.
[9/18/15 8:58:35:845 CDT] 00000053 NGUtil$Server I ASND0003I: Detected server mc_hadrdmn1 stopped on node hadrdmn1Node01
[9/18/15 8:58:36:972 CDT] 00000053 NGUtil$Server I ASND0003I: Detected server hadrdmn1 stopped on node hadrdmn1Node01
[9/18/15 8:58:39:025 CDT] 000002f8 RegisteredRes W WTRN0052E: An attempt by the transaction manager to call one phase commit on a transactional resource has resulted in an XAER_RMFAIL error. The resource was com.ibm.ws.tx.jta.JTAXAResourceImpl@4c76cc93#{XidImpl: formatId(57415344), gtrid_length(36), bqual_length(54), data(0000014fe0bfe5db0000000173b43f14faaf1d977d64f632cd3a9039a25a6a0577d9dfb10000014fe0bfe5db0000000173b43f14faaf1d977d64f632cd3a9039a25a6a0577d9dfb10000000100000000000000000000001)}
[9/18/15 8:58:40:358 CDT] 000002f9 RegisteredRes W WTRN0048W: An attempt by the transaction manager to call commit on a transactional resource has resulted in an XAER_RMFAIL error. The resource was com.ibm.ws.tx.jta.JTAXAResourceImpl@77d5f84#{XidImpl: formatId(57415344), gtrid_length(36), bqual_length(54), data(0000014fe0bfd4df0000000173b43d54faaf1d977d64f632cd3a9039a25a6a0577d9dfb10000014fe0bfd4df0000000173b43d54faaf1d977d64f632cd3a9039a25a6a0577d9dfb10000000100000000000000000000001)}
[9/18/15 8:58:46:147 CDT] 000002dd XARecoveryDat W WTRN0005W: The XAResource for a transaction participant could not be recreated and transaction recovery may not be able to

complete properly. The resource was [com.ibm.ws.sib.ra.recovery.impl.SibRaXaResourceInfo@261591410 <busName=itim_bus> <meName=mc_hadrdmn1.000-itim_bus> <meUuid=38ECD7B388DABA29> <xaRecoveryAlias=itim_jms> <useServerSubject=false> <providerEndpoints=null>]. The exception stack trace follows: com.ibm.ws.Transaction.XAResourceNotAvailableException: com.ibm.websphere.sib.exception.SIResourceException: CWSIT0019E: No suitable messaging engine is available on bus itim_bus that matched the specified connection properties {connectionMode=Recovery, targetSignificance=Required, targetTransportChain=null, targetType=MEUuid, busName=itim_bus, providerEndpoints=null, targetGroup=38ECD7B388DABA29}. Reason for failure: CWSIT0103E: No messaging engine was found that matched the following parameters: bus=itim_bus, targetGroup=38ECD7B388DABA29, targetType=MEUuid, targetSignificance=Required, transportChain=InboundSecureMessaging, proximity=Bus.
[9/18/15 8:59:08:547 CDT] 000000ad SibMessage I [:] CWSIT0028I: The connection for messaging engine hadrdmn1.001-itim_bus in bus itim_bus to messaging engine mc_hadrdmn1.000-itim_bus started.

When the first node is recovered from the failure and comes back online, both the application server and messaging server can be restarted. In the application server SystemOut log, messages are produced for the messaging-server status.

[9/21/15 10:04:33:770 CDT] 00000067 SibMessage I [itim_bus:hadrdmn1.001-itim_bus] CWSID0016I: Messaging engine hadrdmn1.001-itim_bus is in state Joined.
[9/21/15 10:04:33:771 CDT] 00000067 SibMessage I [itim_bus:hadrdmn1.000-itim_bus] CWSID0016I: Messaging engine hadrdmn1.000-itim_bus is in state Joined.
[9/21/15 10:04:35:733 CDT] 0000006d SibMessage I [itim_bus:hadrdmn1.000-itim_bus] CWSID0056I: Connection to database is successful
[9/21/15 10:04:36:454 CDT] 0000007c SibMessage I [itim_bus:hadrdmn1.000-itim_bus] CWSIS1538I: The messaging engine, ME_UUID=43A5D1389D3AB519, INC_UUID=06C387C5F06F6560, is attempting to obtain an exclusive lock on the data store.
[9/21/15 10:04:36:562 CDT] 0000007c SibMessage I [itim_bus:hadrdmn1.000-itim_bus] CWSIS1537I: The messaging engine, ME_UUID=43A5D1389D3AB519, INC_UUID=06C387C5F06F6560, has acquired an exclusive lock on the data store.
[9/21/15 10:04:39:276 CDT] 0000006d SibMessage I [itim_bus:hadrdmn1.000-itim_bus] CWSID0016I: Messaging engine hadrdmn1.000-itim_bus is in state Started.
[9/21/15 10:04:40:469 CDT] 000000aa SibMessage I [:] CWSIT0028I: The connection for messaging engine hadrdmn1.000-itim_bus in bus itim_bus to messaging engine hadrdmn1.001-itim_bus started.

Once the server was started, all the messages that remained in the server's local queue were processed. The reconciliation request status changed to Success when all messages were processed successfully.

Configuring for Disaster Recovery (DR)

By definition, a DR configuration has two data centers connected by a WAN. In a DR-only configuration without HA, each data center has its own set of middleware (one database, one LDAP, one ISDI) and one or more WAS nodes. The data centers work on the same set of data. ISIM only supports active/passive DR configurations - meaning that only one site is active at the time. The failover site is configured and kept ready to be brought online. It cannot be active at the same time as the primary site.

Disaster recovery, by definition, is a system architectural problem and it is for this reason that both DB2 and WAS experts recommend a File System Based Replication solution for DR.

A File System (Storage) based replication DR (disk mirroring) solution would include the creation of a Replication Consistency Group that would contain:

1. Websphere Transaction Logs

2. DB2 ISIM Data

3. DB2 Transaction Logs

4. Process Server Configuration

Seldom do customers need just DR without HA, but there are valid use cases and it does come up periodically.

In an DR deployment your infrastructure will mirror the infrastructure configured in an HA deployment (for example, the load balancers used).

ISIM Nodes

Each data center must have its own ISIM cluster. **It is explicitly recommended to NOY have a WAS-ND cell span two data centers**. Within a WAS-ND cell there may be one or more WAS-ND nodes running the ISIM application in a cluster. Because of how the WAS messaging engine and in turn ISIM picks up and processes scheduled messages from the ISIM database, the ISIM applications must be stopped in the fail-over data center. Having them online will result in the fail-over nodes picking up and attempting to process scheduled messages which will fail because the fail-over DB2 instance is read-only.

The SIB messaging system isolates messages to a specific messaging engine using a UUID for each messaging engine. As a result, in a remote fail-over for the ISIM nodes to pick up and work on pending messages, the message engine UUID must be the same between the ISIM nodes in the primary data center and the fail-over ISIM nodes in the fail-over data center.

This requires that each server in the primary data center must have a twin in the fail-over data center. If the number of nodes between the data centers are mismatched (eg: 4 nodes in the primary and 2 nodes in the fail-over) either the messages on the second two nodes in the primary data center will not be processed in the case of a fail-over or you need to split the 2 machines in the secondary data center into 4 separate machines such that they match).

To do this, follow these steps:

1. Install and configure WAS within the primary data center

2. Install ISIM on the nodes in the primary data center

3. Install and configure WAS within the secondary data center

4. Back up the profile on one WAS instance in the primary data center and restore it to one of the instance within the fail-over data center using these instructions (we'll call this the

primary node's twin)

5. Change the hostname on the newly-restored profile using the procedures defined in the link above

6. Repeat the back up and restore procedures for each node

7. Install ISIM on the nodes in the secondary data center

In this configuration, only one node in a twin pair can be active at a time. For instance, if you have nodes A in the primary data center and A' in the secondary data center, A and A' can not both be up at the same time. In order to bring A' up, node A must be shut down.

ISIM uses a two-phase commit process via the WAS transaction manager to ensure data consistency between the various components (JMS, DB2) used in a transaction. In brief, with an XA transaction, the "root transaction manager" or "global commit coordinator" (WAS in this case) is the final authority for transactions, propagating transaction commands; such as prepare, commit, rollback to the resource managers (DB2). As a result, if the WAS transaction logs are not exactly aligned with the DB2 logs, we have a "time gap" in a failure scenario where some transactions are locked by DB2 awaiting a commit or rollback instruction from WAS (as required by the XA specification).

The only way to provide zero data loss and in turn insure that all the logs are exactly aligned is to employ disk replication/mirroring between the primary server(s) and the back up servers, with all the logs, both WAS and DB2, being in the same disk replication consistency group.

If disk mirroring isn't an option, then coordination of the command used by DB2 "db2 rollforward db <dbname> to end of logs" with a disk snapshot of the WAS transaction logs should provide a high degree of log synchronization.

DB2 log shipping is implemented as a user exit program, so coordination of the DB2 and file system commands could be accomplished via this exit.

DB2

HADR, as discussed in the HA section above, is the preferred way of setting up a DR-only scenario.

Alternatively, Q-replication can be used.

DB2 does have best practices concerning disaster recovery.

Directory Server

The directory server in a DR environment is the same as that for HA: both systems should be in a master-master replication but only one active at a time.

Configuring for Both HA and DR

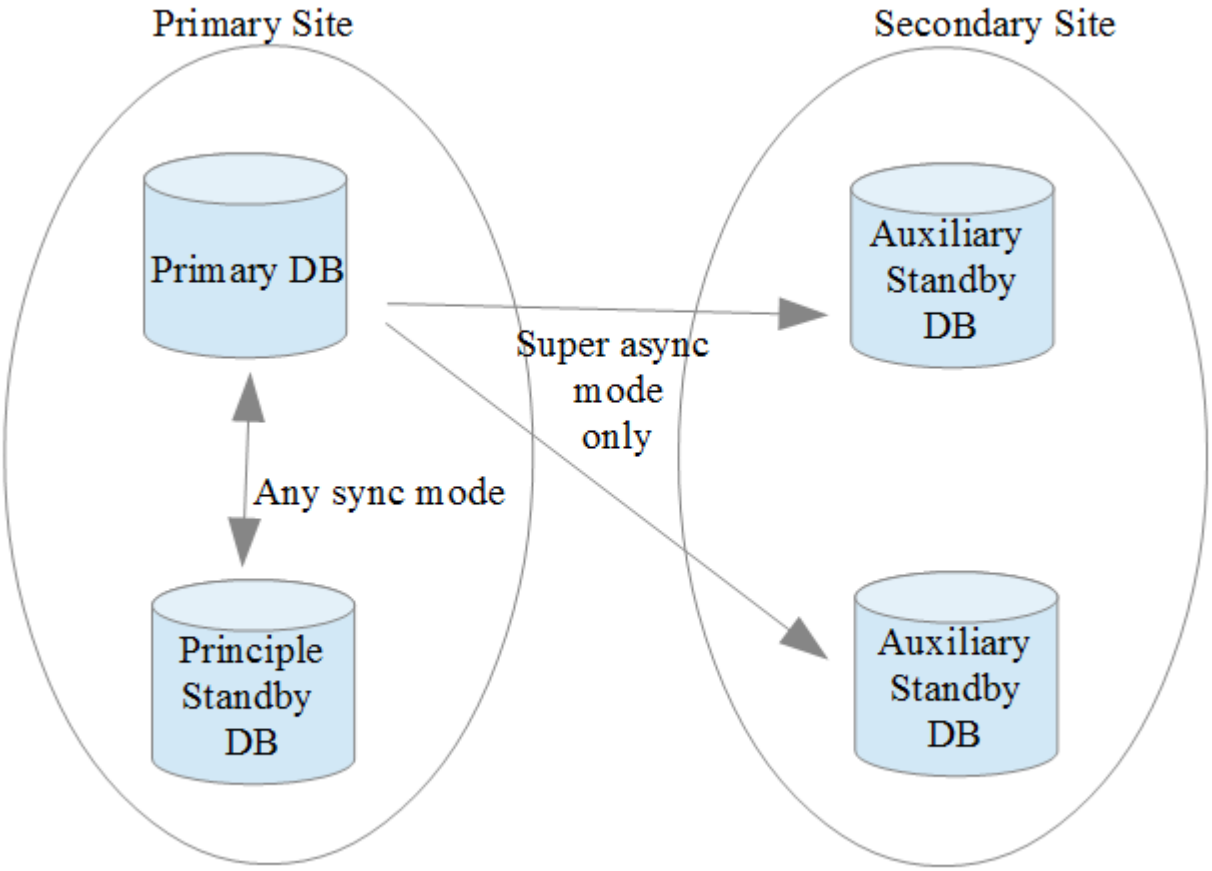
Creating an HA+DR environment requires using some of the same building blocks as discussed for HA-only or DR-only.

ISIM Nodes

The ISIM node configuration in an HA+DR configuration is the same as that for DR only – see above.

DB2

DB2 11.1 supports up to three standby databases. With this, you can achieve HA+DR objectives with a single technology. HADR with multiple standby databases provides very fast failover, typically less than 1 minute, and the option of zero data loss, which provides the greatest protection on data. This is the recommended way to achieve HA+DR. Other options are Q replication and log shipping. Q replication tends to be complicated to configure. Log shipping is easier to deploy, but the target database always at least 1 full log file behind.



The diagram above shows the HADR multiple standby topology. The principal standby is equivalent to the standby today, it is co-located with the primary database. Two auxiliary

standbys are placed at the secondary site. They can be dynamically added. Data on the auxiliary standbys are always fed from the primary. The synchronization mode between the primary and the principal standby can be one of the supported mode. The mode of the auxiliary standbys is always SUPERSYNC.

Use the following commands to set local configuration and target list information on each database node.

db2 update db cfg for <DBNAME> using HADR_LOCAL_HOST <HOST NAME>

db2 update db cfg for <DBNAME> using HADR_LOCAL_SVC <SERVICE NAME OR PORT #>

db2 update db cfg for <DBNAME> using HADR_SYNCMODE <SYNCMODE>

db2 update db cfg for <DBNAME> using HADR_TARGET_LIST

<principalhostname:principalservicename| auxhostname1:auxservicename1| auxhostname2:auxservicename2>

The remote database parameters (HADR_REMOTE_HOST, HADR_REMOTE_SVC AND HADR_REMOTE_INST) are set automatically when HADR starts. You can use command line to optionally set them.

On the primary database, run the following commands:

db2 update db cfg for <DBNAME> using HADR_REMOTE_HOST <IP ADDRESS OF PRINCIPAL STNDBY>

db2 update db cfg for <DBNAME> using HADR_REMOTE_SVC <PORT # on PRINCIPAL STNDBY>

db2 update db cfg for <DBNAME> using HADR_REMOTE_INST <INSTNAME OF PRINCIPAL STNDBY>

On the standby database, run the same set of commands. The parameter value should point to the primary database.

Use the following command to start each standby database.

db2 start hadr on database <DBNAME> as standby

After each standby database starts, use this command to start the primary database:

db2 start hadr on database <DBNAME> as primary

(For more information on DB2 HADR multiple standby, see HADR multiple standby databases in the DB2 Knowledge Center.)

Directory Server

The directory server in a HA+DR environment is the same as that for HA. All systems should be in

a master-master replication. Only one system is active at a time.

Further DR Environment Operations

Failing over gracefully from production to standby DR environment

Failing over gracefully to a DR environment requires a few steps to ensure data consistency. The following procedure should be followed

1. Shut down all ISIM nodes to ensure WAS completes all pending transactions.
2. Force DB2 to do a manual failover if using HADR. If you are using log shipping, ensure all pending logs are transferred to the failover database and that the failover database is brought fully online.
3. Confirm that all LDAP replication queues are empty in the primary data center.
4. Start the ISIM nodes in the failover data center.
5. Validate failover transaction consistency (see below)
6. Route UI traffic to the DR data center

Fail-back to Original Configuration

When failover occurs, the standby DB2 server is promoted to primary; furthermore, the old primary is then demoted to standby. In order to failover back to the original configuration, these steps were taken:

1. Stop all WebSphere nodes in the cluster.
2. Back up the current primary DB server.
3. Restore the backup data to the secondary. Use the without rolling forward clause.
4. Start HA-DR on the secondary server.
5. Start HA-DR on the primary server.
6. Verify HA-DR connectivity (using the script hadrstatus_pub.sh)
7. Login to ISIM and verify that the new user is still present

Validating Transaction Consistency After a Failover

When a failover occurs, both in a controlled stop and in a disaster, it is important to validate that there are no inconsistencies between the WAS transaction logs and the pending transactions

within the dependent middleware, such as DB2.

Use the following steps to validate transaction log consistency:

Resolve any in-doubt transactions within DB2. See Resolving indoubt transactions manually.

Review the WAS transaction logs, both for any indoubt transactions, which will be identified by the global transaction ID, that might be reflected in DB2 (step 1 above) or any transaction(s) WAS didn't complete. (See Transactions needing manual completion.)

As necessary, manually complete any transactions, both in DB2 (refer to 1 above) and WAS. (See Transactions needing manual completion.)

Installing and Validating Maintenance Packages

To install and validate maintenance packages in the failover environment, start the failover environment in a restricted configuration. All ISIM nodes in the failover data center must temporarily be pointed to the middleware (LDAP and DB2) instances in the primary data center. Use virtual IPs or host files to point to the middleware in the primary data center. The ISIM nodes can then perform write requests to the middleware.

After ISIM nodes are configured to access the primary data center's middleware, edit the ISIM_HOME/data/enroleStartup.properties files.

Locate the following line:

enrole.startup.names=Scheduler,PasswordExpiration,DataServices,PostOffice,ReconcilerCleanup,RemotePending,PolicyAnalysis,PasswordSynchStore

Change that line to the following line:

enrole.startup.names=PasswordExpiration,DataServices,PostOffice,ReconcilerCleanup,RemotePending,PolicyAnalysis,PasswordSynchStore

Doing this ensures that scheduled messages, including reconciliation requests, are not processed.

Stop one ISIM node in the primary data center. On its twin in the failover data center, start the nodeAgent and the ISIM Application server instance, but do not start the ISIM JMS server instance. ISIM then does not pick up and process transactions from the ISIM shared queue. Install the WAS or ISIM maintenance on this node and validate it was successful by doing a password change.

After the node has been validated, shut it down and start its twin in the primary data center. Update the ISIM_HOME/data/enroleStartup.properties and ISIM_HOME/data/enRole.properties files to restore the enrole.startup.names parameter to its original value.

References

DB2 Information

DB2 UDB Backup and Recovery with ESS Copy Services
http://www.redbooks.ibm.com/abstracts/sg246557.html

DB2 11.1 HA/DR Multiple Standby Databases
https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.admin.ha.doc/doc/c0059994.html

DB2 10.1 HADR MultipleStandby Databases
http://public.dhe.ibm.com/software/dw/data/dm-1206hadrmultiplestandby/HADR_Multiple_Standbys_jn20.pdf

High Availability and Disaster Recovery Options for DB2 for Linux, UNIX, and Windows
http://www.redbooks.ibm.com/redbooks/pdfs/sg247363.pdf

IBM DB2 11.1 for Linux, Unix and Windows Documentation
https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.welcome.doc/doc/welcome.html

Storage Specific Information

IBM System Storage DS6000 Series: Copy Services in Open Environments
http://www.redbooks.ibm.com/abstracts/sg246783.html

IBM DS8870 Copy Services for Open Systems
http://www.redbooks.ibm.com/abstracts/sg246788.html

IBM System Storage FlashCopy Manager and PPRC Manager Overview
http://www.redbooks.ibm.com/abstracts/redp4065.html

IBM TotalStorage Productivity Center for Replication Using DS8000
http://www.redbooks.ibm.com/redpieces/abstracts/sg247596.html

WebSphere Application Server Information

WebSphere Application Server Network Deployment 8.5.5
http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/as_ditamaps/was855_welcome_ndmp.html

WebSphere Application Server V8: Administration and Configuration Guide
http://www.redbooks.ibm.com/abstracts/sg247971.html

WebSphere Application Server V8.5 Concepts, Planning, and Design Guide
http://www.redbooks.ibm.com/redbooks/pdfs/sg248022.pdf