



IBM® DB2® for Linux®, UNIX®, and Windows®

# Best practices

## Business Application Continuity Offering

Paul Dorwin  
*Software Developer, IBM*

Jason Woods  
*Software Developer, IBM*

Issued: January 2015

Executive summary .....	3
Introduction .....	3
Definitions.....	4
Database configuration .....	5
Example for setting up member subsets.....	5
Client configuration.....	6
Java client configuration .....	6
Non-Java client configuration .....	7
Application rerouting.....	8
Preferred member offline.....	8
Warm standby member offline .....	8
Scenarios where failback will not occur .....	9
Inflight transactions .....	9
Role Switching.....	9
Verifying and monitoring compliance.....	11
Verifying proper failover and failback .....	11
Verifying role switch .....	11
Maintaining Compliance .....	13
Conclusion .....	14
Further reading.....	14
Contributors.....	14
Notices .....	15
Trademarks .....	16
Contacting IBM .....	16

## Executive summary

You can use the IBM® DB2® Business Application Continuity Offering to implement a two member DB2 pureScale® cluster in a DB2 Enterprise Server Edition (ESE) or DB2 Workgroup Server Edition (WSE) environment where one DB2 member may be used to service applications and the other member may be used to perform a set of permitted administrative activities for the cluster and the databases within it. The first member is called the preferred member, while the other member is called the warm standby member.

In the event that the preferred member servicing applications fails or is stopped to perform scheduled maintenance, the applications will automatically failover to the warm standby member, providing continuous availability of the applications. When the failed member is once again available, the applications will fail back to that member. During the failback period, applications are allowed to run on both members. It is recommended to monitor applications after a failure to verify proper failback.

If you follow the best practices described in this paper you are eligible to pay for a minimum license entitlement on the warm standby member. These terms are available if you purchase the Business Application Continuity Offering along with DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, or if you have entitlement to DB2 Advanced Enterprise Server Edition (AESE) or DB2 Advanced Workgroup Server Edition (AWSE).

## Introduction

This paper describes how to setup, monitor, and maintain compliance with the Business Application Continuity Offering by performing the following tasks:

- Creating member subsets on the DB2 pureScale instance. The database will then be able to route applications to the appropriate member.
- Using transaction level workload balancing for applications to provide faster failback after an outage of the preferred member.
- Configuring Java clients with the correct Java properties for applications to connect to the DB2 Server member subset.
- Configuring the `db2dsdriver.cfg` file on Non-Java clients so that the application can connect to the DB2 server member subset.
- Describing member failures and monitoring methods to ensure license compliance.

## Definitions

The following definitions will assist in the remainder of this paper:

### Preferred member

- The member used for normal application processing. If desired, this member can also be used to execute any of the administrative tasks for the instance.
- If both members in the cluster are configured with the same amount of CPU and memory resources, then either member can be used as the preferred member. If the members are configured such that one has more CPU and memory resources, then that member should be selected as the preferred member.

### Warm standby member

- During normal operation, this member can only be used for administrative tasks. During maintenance or failure of the preferred member, the normal processing of applications will be rerouted to this member.

### Administrative tasks

- Business Application Continuity compliance monitoring
- Database commands *backup, restore, inspect, create, reorg, runstats, and drop*
- Monitoring, including DB2 Explain and any diagnostic or problem determination activities
- Usage of Database Definition Language (DDL)
- Database and database manager configuration
- Log based capture utilities for the purpose of data capture
- Security setup and administration.

### Normal operation

- Normal operation refers to the typical state when no members are offline due to failures or maintenance activities.

### Failover operation

- When the preferred member fails or needs to be stopped to perform maintenance activities, applications are rerouted to the warm standby member to continue their work.

### Failback operation

- When the preferred member becomes available, the applications are routed back to the preferred member. During this transition time, applications are temporarily allowed to be serviced by both the preferred member and the warm standby member.

## Database configuration

To maintain compliance with the Business Application Continuity Offering, we recommend you use [member subsets](#) to route application connections to the preferred member. Use the [create](#), [alter](#), and [drop](#) member subset procedures to manage member subsets.

Each database will have a member subset associated with it. There will be a database alias for the member subset. All database member subsets must use the same preferred member as the only member in the subset.

### *Example for setting up member subsets*

The database in this example is called SALES. The following SQL statements can be run from one of the member hosts to create the member subsets and corresponding database aliases.

- SALES\_P will be the database alias associated with the preferred member, which for this example will be member 0.
- SALES\_W will be the database alias associated with the warm standby member, which for this example will be member 1.

```
db2 -t
CONNECT TO SALES ;
CALL SYSPROC.WLM_CREATE_MEMBER_SUBSET (
    'SALES_PREFERRED_SUBSET',
    '<databaseAlias>SALES_P</databaseAlias>',
    '( 0 )'
) ;
CALL SYSPROC.WLM_CREATE_MEMBER_SUBSET (
    'SALES_WARM_STANDBY_SUBSET',
    '<databaseAlias>SALES_W</databaseAlias>',
    '( 1 )'
) ;
CONNECT RESET ;
TERMINATE ;
```

By default the member subsets will be created as inclusive member subsets. When the member in the subset is not online the applications will be routed to the member that is not in the subset. When the member in the subset is once again available, applications will be routed back to the member in the member subset.

## Client configuration

Each client host must have one of the [IBM Data Server Drivers](#) installed. This paper describes the use of the [IBM Data Server Driver Package](#). You can download the package for your specific version of DB2 from the following [IBM support page](#).

Workload balancing will be used to direct applications to the appropriate member. Use the SALES\_P alias to direct normal applications to the preferred member. Use the SALES\_W alias to direct administrative tasks to the warm standby member.

### *Java client configuration*

You can find details about configuring your Java client in the IBM Knowledge Center:

- [Common IBM Data Server Driver for JDBC and SQLJ properties for all supported database products](#) describes the properties available for the IBM Data Server Driver for JDBC and SQLJ.
- [Java client support for high availability on IBM data servers](#) describes the high availability features available for the Java client.
- [Configuration of DB2 for Linux, UNIX, and Windows automatic client reroute support for Java clients](#) explains how to configure the Java client for automatic client reroute.
- [DB2BaseDataSource class](#) describes the various properties available for the class.

Workload balancing is only available in the Java client type 4 driver. The following code fragment shows the various properties that you need to setup in order to provide automatic failover and failback in a Java program. If an application performs both normal processing and administrative tasks, then that application must use the preferred alias SALES\_P. The alternate server and port properties are present for the case where the Java application is starting while the member identified in the member subset is offline.

```
// Create DB2SimpleDataSource object
com.ibm.db2.jcc.DB2SimpleDataSource ds = new
com.ibm.db2.jcc.DB2SimpleDataSource();

// Use only one of the following set of properties
// to setup use of preferred member or warm standby member.
// Setup the preferred member properties if this
// driver is for applications.
// ds.setDatabaseName("SALES_P");
// ds.setServerName("preferredHost");
// ds.setClientRerouteAlternateServerName(warmStandbyHost);
```

```

// Setup the warm standby member properties if this
// driver is for administrative tasks.
// ds.setDatabaseName("SALES_W");
// ds.setServerName("warmStandbyHost");
// ds.setClientRerouteAlternateServerName("preferredHost");

// Common properties
ds.setPortNumber(50001);
ds.setClientRerouteAlternatePortNumber( Integer.toString(50001) );
ds.setUser("username");
ds.setPassword("password");

// Enable transaction-level workload balancing
ds.driverType = 4;
ds.enableSysplexWLB = true;
ds.enableSeamlessFailover = 1;

```

### *Non-Java client configuration*

You need to configure the IBM data server driver configuration file [db2dsdriver.cfg](#). This file contains the information that will enable your non-Java clients to find the associated member subset aliases. See [Non-Java client support for high availability for connections](#) and the various configuration/examples in the IBM Knowledge Center for additional information. For Perl scripts follow the configuration instructions in the document [DB2 Perl Database Interface for LUW](#) on the IBM Support website.

The db2dsdriver.cfg file will contain the following configuration information:

```

<configuration>
  <dsncollection>
    <dsn alias="SALES_P" name="SALES_P" host="preferredHost"
port="50001">
      <parameter name="Authentication" value="Server_Encrypt"/>
    </dsn>
    <dsn alias="SALES_W" name="SALES_W" host="warmStandbyHost"
port="50001">
      <parameter name="Authentication" value="Server_Encrypt"/>
    </dsn>
  </dsncollection>
  <databases>
    <database name="SALES_P" host="preferredHost" port="50001">
      <wlb>
        <parameter name="enableWLB" value="true"/>
      </wlb>
    </database>
  </databases>
</configuration>

```

```

        </wlb>
    </database>
    <database name="SALES_W" host="warmStandbyHost" port="50001">
        <wlb>
            <parameter name="enableWLB" value="true"/>
        </wlb>
    </database>
</databases>
<parameters>
    <parameter name="CommProtocol" value="TCPIP"/>
</parameters>
</configuration>

```

Each database in your pureScale cluster will have corresponding *<dsn>* and *<database>* entries in `db2dsdriver.cfg`. It is recommended to not catalog the `tcPIP` node or db alias on the client host as they will override the configuration in `db2dsdriver.cfg`.

## Application rerouting

When an application first connects to the database, a server list is returned to the client. When the primary member becomes unavailable due to maintenance or failure, the client uses the server list to reroute applications to an alternate member.

### *Preferred member offline*

If the preferred member is undergoing maintenance or fails, then the applications using the preferred alias will be rerouted to the warm standby member. When the preferred member is again available, the applications will be rerouted, or failed back, to the preferred member. This movement may be on transaction or connection boundaries and thus be transparent to the application. Applications that do not fail back automatically need to be forced off the warm standby member and restarted on the preferred member.

### *Warm standby member offline*

As with the preferred member case, maintenance or failure of the warm standby member will cause administrative applications using the warm standby member alias to be rerouted to the preferred member. Once the warm standby member is available, the administrative applications will be failed back to the warm standby member. However, there is no requirement to force administrative applications to fail back to the warm standby member.



## *Scenarios where failback will not occur*

The documentation topic [DB2 client support for the DB2 pureScale Feature](#) in the IBM Knowledge Center points out the following restrictions for workload balancing. These features are problematic if a failure occurs during COMMIT or ROLLBACK operations.

- The connection uses global variables.
- An encrypted password is used.
- Open WITH HOLD cursors are used.
- Declared temporary tables (DGTT) are used.
- A transform group was set.
- The session authorization ID was changed.
- PL/SQL packages or SQL/PL modules are used.
- Cursor variables are used.
- Sequence values are used and DB2\_ALLOW\_WLB\_WITH\_SEQUENCES communication variable is not enabled.
- Created temporary tables (CGTTs) that were created with the PRESERVE ROWS option are used.

Applications that make use of these features will be failed over to the warm standby member if the preferred member is brought down for maintenance or fails. However, if the failure occurred during a COMMIT or ROLLBACK operation, then, automatic client reroute is non-seamless and failback to the preferred member will not automatically occur. These applications will need to be forced off the warm standby member and restarted on the preferred member.

Applications that use connection level workload may not failback in a timely manner. These applications may also need to be forced off the warm standby member and restarted on the preferred member.

## *Inflight transactions*

If a member fails, any inflight transactions that are not capable of being seamlessly rerouted will return a SQLCODE -30108 error to the application, indicating that the transaction was aborted but connectivity has been re-established to the alternate member.

## *Role Switching*

After the preferred member has failed and applications have failed over to the administrative member, role switching can be used to change the role of preferred member to warm standby member and warm standby member to preferred member. The benefit of a role switch is that applications will not have to fail back after the failed member has been repaired. Also, if your applications are using any of the features listed previously, you should consider using role switching to change the roles of the members for your pureScale instance. The preferred member and warm standby member can be changed in the member subsets of each database using the [WLM ALTER MEMBER SUBSET](#) procedure. Once the member subsets have been

changed, the server list at the clients will be updated. Applications can then be routed to the appropriate member.

Continuing with our examples above, the following commands will change roles within the preferred member and warm standby member subsets:

```
db2 -t
CONNECT TO SALES ;
CALL SYSPROC.WLM_ALTER_MEMBER_SUBSET(
    'SALES_PREF_SUBSET', NULL, '( drop 0, add 1 )' );
CALL SYSPROC.WLM_ALTER_MEMBER_SUBSET(
    'SALES_WARM_STANDBY_SUBSET', NULL, '( drop 1, add 0 )' );
CONNECT RESET ;
TERMINATE ;
```

When performing maintenance on the preferred member, you can use the role switching method described in this section to move applications off the member. A role switch can also be performed prior to repairing a failed preferred member. After the member becomes available, it will be the warm standby member, removing the need for the applications to be rerouted a second time.

In order to maintain compliance with the Business Application Continuity Offering, the role switch must be performed on all databases in the pureScale instance. If there are a large number of databases, a role switch can take some time to complete because a connection needs to be established with each database.

## Verifying and monitoring compliance

After configuring your pureScale cluster with member subsets, you need to verify that it is properly setup. You can do so using the sample Java test program `subsetVerify.java` and the sample Perl script `subsetVerify.pl`, provided in the file `BAC_samples.zip` that accompanies this paper on the following webpage: <https://ibm.biz/BdEqma>. Each of the samples performs the following sequence of steps:

- Connect to the specified database alias
- Drop and create table `t1`
- Loop performing the following operations:
  - Insert a value into `t1`, commit.
  - Query the system catalog to determine on which member the application is running.
  - If the member has changed, output the member identified in the query.

In the following sections, the preferred member is member 0 and the warm standby member is member 1. This verification can first be done in a test environment. It should also be done in the production environment prior to going into production.

### *Verifying proper failover and failback*

Perform the following sequence of steps to verify proper failover and failback:

- Start the test application on each client host using the preferred alias `SALES_P`.
- Verify the test application is connected to the correct member.
- Simulate a failure using one of the following methods:
  - Issue the command `db2stop member 0 quiesce`.
  - A failure can be simulated by running the command `kill -9 <pid of the 'db2sysc 0'>` on the preferredHost. Cluster automation will restart the member.
- Verify that the test application output indicates that it is now using the preferred alias and that it is connected to the warm standby member.
- Recover the failed member. The method to recover is dependent on the method used to simulate the failure.
  - If you issued the `db2stop` command, issue `db2start member 0`.
  - If you issued a single `kill -9` command, wait for the cluster automation to restart member 0.
- Verify that the test application output indicates it is once again connected to and using the preferred member.

### *Verifying role switch*

You can verify that applications will move to another member by changing the member in the member subset using the following steps:

- Start the test application on each client host using the preferred alias `SALES_P`.
- Issue the following commands on the preferredHost:

```
db2 -t
CONNECT TO SALES ;
CALL SYSPROC.WLM_ALTER_MEMBER_SUBSET(
    'SALES_PREF_SUBSET', NULL, '( drop 0, add 1 )' );
CONNECT RESET ;
TERMINATE ;
```

- Verify that the test application output indicates that it has been rerouted to member 1.
- Issue the following commands: on the preferredHost

```
db2 -t
CONNECT TO SALES ;
CALL SYSPROC.WLM_ALTER_MEMBER_SUBSET(
    'SALES_PREF_SUBSET', NULL, '( drop 1, add 0 )' );
CONNECT RESET ;
TERMINATE ;
```

- Verify that the test application output indicates that it has been rerouted back to member 0.

## Maintaining Compliance

As previously described, the Business Application Continuity Offering allows for application workloads and queries to be running on only one DB2 member in a two member DB2 pureScale cluster. Specifically, all of the workloads must be running on the same member (referred to as the preferred member). Additionally, certain database administrative tasks can be run on the second member (referred to as the warm standby member). The DB2 software does not strictly enforce these workload compliance rules, but this document has provided you with the steps to properly configure your environment for compliance. You can also query each database within the pureScale cluster to ensure that the environment is behaving as expected. This can be done periodically or whenever a new application is deployed into the environment.

To query the databases, connect to each database (on either of the two members) and execute the following SQL statement. The **where uow.member = 1** clause should be using the member number of the warm standby member. This query is intended to identify applications that are using the warm standby member. Only applications performing administrative tasks should be present in the result set.

```
select uow.member,
       uow.coord_member as COORD,
       uow.application_handle as APP_HANDLE,
       substr(uow.application_id,1,40) as APP_ID,
       substr(uow.workload_name, 1,40) as WORKLOAD_NAME,
       substr(uow.client_wrkstnname, 1,10) as CLIENT_HOST,
       substr(uow.client_applname, 1,20) as CLIENT_APP_NAME
from table(mon_get_unit_of_work( NULL, -2 ) ) as uow
where uow.member = 1 and uow.uow_start_time is not null and
      uow.uow_stop_time is null
```

The member on which applications are running can be found in the COORD column of the result set. Other information identifying the application and the client is included as well. Under normal operations, all of the applications should be running on the preferred member. Note that the number identifying the member may change over time, depending on whether failovers without corresponding failbacks have been previously performed. What is important is that they are all running on the same member. While administrative tasks are typically run on the warm standby member, they are allowed to be running on either of the two members in the cluster.

If applications are running on the warm standby member, either force them off of this member and restart them on the preferred member, or perform a role switch on the preferred member subset for all database.

## Conclusion

This paper discussed the main features of the Business Application Continuity Offering. The setup of the databases using member subsets was discussed. The client configurations for Java and non-Java clients were presented. The paper also discussed the tasks necessary to handle member failures and for monitoring compliance with the Offering. Throughout the paper links to topics in the IBM Knowledge Center were included for further investigation.

## Further reading

- Information Management best practices:  
<http://www.ibm.com/developerworks/data/bestpractices/>
- DB2 for Linux, UNIX, and Windows best practices:  
<http://www.ibm.com/developerworks/data/bestpractices/db2luw/>
- [Workload balancing, automatic client reroute, and client affinities concepts and administration](#)
- [Workload balancing, automatic client reroute, and client affinities](#)

## Contributors

Christian Garcia-Arellano  
*DB2 software development*

Jimmy Stamkos  
*DB2 quality assurance*

Michael Springgay  
*DB2 software development*

Lei Pan  
*DB2 quality Assurance*

## Notices

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

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

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

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

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

Without limiting the above disclaimers, IBM provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any recommendations or techniques herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environment does so at their own risk.

This document and the information contained herein may be used solely in connection with the IBM products discussed in this document.

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

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE: © Copyright IBM Corporation 2015. All Rights Reserved.

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

## ***Trademarks***

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

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

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

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

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

## ***Contacting IBM***

To provide feedback about this paper, write to [db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com)

To contact IBM in your country or region, check the IBM Directory of Worldwide Contacts at <http://www.ibm.com/planetwide>

To learn more about IBM Information Management products, go to <http://www.ibm.com/software/data/>