IBM

High Availability and Disaster Recovery Configurations

for IBM SmartCloud Control Desk and IBM Maximo Products

- Learn how to set up high availability and disaster recovery configuration options
- Design a multisite deployment with load balancing
- Configure middleware components for availability

Axel Buecker
Daniel McConomy
Alfredo Ferreira
Niraj Vora

Redbooks

ibm.com/redbooks

International Technical Support Organization

High Availability and Disaster Recovery Configurations for IBM SmartCloud Control Desk and IBM Maximo Products

February 2013

SG24-8109-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (February 2013)

This edition applies to Version 7.5 of IBM SmartCloud Control Desk (product number 5725-E24) and the IBM Tivoli Process Automation Engine Version 7.5.0.2.

Contents

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.

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.

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:
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. 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 IBM's application programming interfaces.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US 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 http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	IBM®	Redbooks (logo) ®
DB2®	Maximo®	Tivoli®
GPFS™	pureScale®	WebSphere®
IBM SmartCloud™	Redbooks®	

The following terms are trademarks of other companies:

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

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

Microsoft, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

Preface

In today's global environment, more and more organizations need to reduce their downtime to the minimum possible and look for continuous availability of their systems. Products based on the IBM® Tivoli® Process Automation Engine (TPAE), such as IBM Maximo® Asset Management, Maximo Industry Solutions, and IBM SmartCloud™ Control Desk, often play a role in such environments and thus also have continuous availability requirements. As part of that, it is important to understand the High Availability (HA) and Disaster Recovery (DR) capabilities of IBM SmartCloud Control Desk and IBM Maximo Products, and how to assure that all the components of an HA/DR solution are properly configured and tested to handle outages. By outlining some of the topologies we have tested, and the documentation we created, we hope to demonstrate how robust the IBM SmartCloud Control Desk and IBM Maximo infrastructure can be.

This IBM Redbooks® publication covers alternative topologies for implementing IBM SmartCloud Control Desk and IBM Maximo in High Availability and Disaster Recovery configurations.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Axel Buecker is a Certified Consulting Software IT Specialist at the ITSO, Austin Center. He writes extensively and teaches IBM classes worldwide about areas of software security architecture and network computing technologies. He has a degree in Computer Science from the University of Bremen, Germany. He has 26 years of experience in a variety of areas related to workstation and systems management, network computing, and e-business solutions. Before joining the ITSO in March 2000, Axel worked for IBM in Germany as a Senior IT Specialist in Software Security Architecture.

Daniel McConomy is a Systems Configuration Specialist in the IBM Tivoli Process Automation Engine Quality Assurance Development group. He has held this position since 2009. He has been working for IBM since 2007 and started in Level 2 IBM Maximo Support. His recent work involved testing and configuring High Availability and Disaster Recovery solutions for IBM Tivoli Process Automation Engine based products. His specialties include IBM DB2®, Oracle, IBM WebSphere®, LDAP, and Linux/UNIX systems. Daniel is LPIC-1 (Linux Professional Institute) certified and a Novell Certified Linux Administrator.

Alfredo Ferreira is an IBM SmartCloud Control Desk specialist for the IBM Brazil IT Delivery department. He graduated in Systems Analysis and Development at Faculdade de Informática e Administração Paulista (FIAP). Alfredo has been working with IBM Service Management products for four years and has experience in Java development, IBM AIX, IBM WebSphere Application Server, IBM WebSphere MQ, and IBM DB2.

Niraj Vora is a Senior IT Architect in the United States. He has 17 years of experience in the IT field. He holds a Bachelor of Electronics Engineering degree from the University of Mumbai. He has extensive experience deploying enterprise service management applications for IBM commercial clients. He has experience deploying IBM Maximo solutions for various clients. He is a certified IT Specialist, and an IBM Certified Advanced DB2 UDB database administrator.

Thanks to the following people for their contributions to this project:

Richard Conway, Shari Deiana, Editor
International Technical Support Organization, Austin Center

Alfred Schwab, Editor
International Technical Support Organization, Poughkeepsie Center

Thomas Alcott, Ana Biazetti, Alex Chung, Pam Denny, Robert Dunyak, Belinda Fuller, Samuel Hokama, Bruce Jackson, Thomas Lumpp, Leonardo Matos, Markus Mueller, Steven Raspudic, Martin Reitz, Lohitashwa Thyagaraj,

Cheryl Thrailkill
IBM

Aurelien Jarry
Le Groupe Createch, IBM Business Partner, Canada

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Part 1

Business context and solution design

In this part we introduce concepts for increasing availability for the IBM SmartCloud Control Desk and IBM Maximo environment and provide an introduction to recommended HA/DR topologies.

1

Business context

In this chapter we provide a description and overview of high availability and disaster recovery planning for IBM SmartCloud Control Desk and IBM Maximo.

The topics discussed in this chapter are:

- Introduction to high availability and disaster recovery
- Business drivers for increased availability
- Disaster recovery planning
- Target audience

1.1 Introduction to high availability and disaster recovery

High availability (HA) and disaster recovery (DR) are important for organizations that are running mission-critical applications and that must maintain high levels of access to system content. By implementing a highly available environment for your solution, you can minimize the effects that a component or overall system failure can have on daily operations. By implementing a multisite disaster recovery environment for your solution, you can minimize the effects of a complete solution failure on a site due to a natural or man-made disaster.

Applicability: Although this book was specifically created and tested for the IBM SmartCloud Control Desk product, its concepts and configurations are also valid for other IBM Maximo-based products, including IBM Maximo Asset Management and its Industy Solution modules, which are based on the IBM Tivoli Process Automation Engine.

The solution relies on the high availability of the underlying components, such as web server, application server, database, LDAP server, and the IBM Tivoli Process Automation Engine. A cluster manager can be used to monitor and automate system failover.

You can configure the components of your solution environment for high availability in various ways. Different high availability configurations handle failover differently. For this reason, it is important to choose the correct configuration to suit the needs for your organization.

1.2 Business drivers for increased availability

The most common business drivers for increased availability of a particular solution are cost of downtime, service-level agreements, and user satisfaction. Although these are the most common, other business drivers might exist:

- Cost of system outage

  Critical applications and processes can be impacted during system downtime. This can lead to potential loss of revenue as operations may be at a standstill. The benefits of creating system redundancy often outweigh the financial impact of an outage. Maintaining a high availability and disaster recovery solution can be compared with having a good insurance policy.

- Service-level agreement

  SmartCloud Control Desk is used to manage enterprise assets, IT environments, and availability of systems. These tasks are commonly referenced in service-level agreements (SLAs). Therefore, contractual obligations can mandate a certain level of system availability to meet SLAs.

- User satisfaction

  Frequent and unexpected outages during system utilization can directly impact user satisfaction. Users who rely on SmartCloud Control Desk for daily operations may lose confidence in the solution if their productivity is affected.

1.3 Disaster recovery planning

When planning for disaster recovery, keep in mind the following items:

- Identify *types of risks*

  An important part of preparing for a disaster is understanding the type of risks your organization faces. Various potential threats exist that could compromise your environment. Such threats include natural disasters, human error, malicious activity, and software or hardware failure. Other potential risks may exist and should be considered when planning.

- Define a realistic *recovery time objective* (RTO)

  RTO should be defined based on the impact that can be caused by a system outage. RTO relates to the maximum amount of time a system failover procedure can take without severely impacting business operations.

- Identify *essential components*

  SmartCloud Control Desk, just like IBM Maximo and other products, relies on several layers of middleware to properly function. Determining which components are critical for system availability is required for resource planning on alternate sites.

- Determine the *alternate location* for system recovery

  Based on the types of risks identified, plan an alternative location for system recovery. This location should be isolated from the risks that may affect the primary site wherever possible.

  Keep in mind: Distance may affect network latency and overall system performance.

- Define internal *failover procedures*

  In the event of an unplanned outage, a well-documented procedure should be in place in the organization to meet the RTO defined. Administrators should be familiar with the procedure and coordination amongst them should be in place. Often a lack of procedure is responsible for extended periods of downtime in an unplanned outage situation.

- Plan downtime for *implementation and testing*

  When implementing and testing an HA or DR topology, some initial downtime will be required. Testing your failover procedures should be planned at a frequency determined by the organization to ensure they are working properly and meet the RTO defined.

1.4 Target audience

Enterprise IT architects, system administrators, and support teams for IBM SmartCloud Control Desk and IBM Maximo solutions are the target groups for this document. They should be familiar with the IBM Maximo middleware components and their configuration, high availability concepts, as well as other concepts outlined in this book. It is important for all involved parties to understand the solution before the need for a disaster recovery failover procedure is met.

2

Solution design

In this chapter we outline the high availability and disaster recovery topologies that are covered in this book.

The topics discussed are:

- Solution design overview
- Assumptions
- Configuration options

2.1 Solution design overview

In this chapter we provide multiple options for implementing *high availability* and *disaster recovery* solutions. It includes the integration of various clustering techniques designed to make IBM SmartCloud Control Desk (SCCD) and IBM Maximo highly available.

We also present an overview of the different high availability and disaster recovery topologies, where some of the features and products are briefly described. These topologies can help you increase the availability of your IBM SmartCloud Control Desk and IBM Maximo deployment.

This first section introduces and explains high availability and disaster recovery terminology.

2.1.1 High availability models

The availability of any application is measured by its overall uptime. If the users experience errors, time-outs due to the system load, or the application cannot connect to the database, then the application is not considered highly available.

Network outage, hardware failure, operating system or other software-related errors, and power interruptions are examples of failure that can lead to unavailability to the users. In case of such failure, the highly available solution must:

- Shield the application from the failure without appreciable performance degradation.
- Fail over to another server on the cluster.
- Recover from the failure to return the application to normal operations.

In addition, in a highly available application, the impact of maintenance activities on the availability must be minimized.

Highly available applications can be designed in various modes. The standard model dictates how the application behaves when a failure occurs. The following list summarizes the attributes of high availability models and the system recovery time:

- Load-balanced

  In this model both the primary and secondary nodes are active. System transactions are processed in parallel.

- Hot standby

  The software is installed and available on both the primary and secondary nodes. The secondary system is up and running, but no transactions are processed unless the primary node fails. Both systems have access to identical data.

- Warm standby

  The software is installed and available on the secondary node, which is up and running. If a failure occurs on the primary node, the software components are started on the secondary node. Data is regularly replicated on the secondary system or stored on a shared disk.

- Cold standby

  The secondary node acts as a backup for an identical primary system. The secondary node is configured and installed only when the primary system fails. Data is restored from the primary node and the secondary system is restarted. Data is usually backed up and restored from an external storage system.

The cold standby model is not described in this book.

High availability considerations

The following items should be considered for high availability applications:

- Redundancy

  Design redundancy for every component of your application that can take over the workload in case of a failure. If a component of the application is not redundant, then that component can become a *single point of failure* for the application.

- Load balancing

  The application has to be able to distribute workload to all components to avoid system overload.

- Performance

  Performance of the application should be taken into consideration to avoid application unavailability.

- Clustering

  Clustering techniques should be used to connect various components that work together as a single system. The clustering software should be able to transfer workload if one of the components fails.

- Scheduled maintenance

  Maintenance activities should be minimized to reduce impact to the user as much as possible.

- System monitoring

  System monitoring should be implemented to detect failed components and transfer workload to active components.

- Failover procedures

  Procedures need to be developed that are used for failover in case of any application failure. It is important to test these procedures frequently and update them as necessary.

Tip: For more information about designing highly available applications, refer to:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/topic/com.ibm.mbs.doc/gp_highavail/c_ctr_high_availability.html

2.1.2 Disaster recovery models

The term disaster recovery is used to describe any activities needed to restore the application in case of any events that result in complete application failure.

A disaster recovery plan should include the following actions:

- Plan for a secondary site that can be used in the event of an emergency. This site should be geographically separated from the primary site.
- Define the necessary infrastructure to recover the application.
- Back up the database and application data to an off-site location, which could be used to restore and recover the application.

Additional considerations must be addressed as follows:

- Define the recovery time objective (RTO).

  How much time can be spent recovering the application?

- Define the recovery point objective (RPO).

  How much data can the organization afford to lose after recovery?

- Define the backup strategy.

  How much time can pass between backups? How many backup copies do you need to preserve and what is the retention policy?

- Define the required storage space for the backups and servers.

- Configure standby systems on a remote site for disaster recovery.
- Develop and test internal failover procedures.

You have several options for designing a disaster recovery plan. Based on business need, you may decide to have a full backup to guard against any data loss. A secondary site can serve as standby and take over the operations in case of a disaster. Database and file system replication can be used to synchronize data across both sites.

In this book we describe a disaster recovery topology using a multisite active-passive and hybrid-active setup. You can find more details about these specific topologies in 2.3.3, “Multisite active-passive topology” on page 16 and 2.3.4, “Multisite hybrid-active topology” on page 18.

Important: The disaster recovery example is not meant to replace the system backups. Backups should be taken at regular intervals in addition to the disaster recovery solution.

2.2 Assumptions

The topologies described in this book are built on the following assumptions.

- Storage
- User authentication
- Load balancer

2.2.1 Storage

To avoid a single point of failure, a data storage replication mechanism must be in place to achieve local high availability for the required data. Cross-site data replication should be in place for disaster recovery.

There are several forms of data sharing and replication; the most common are:

- SAN mirroring with cross-site replication
- IBM General Parallel File System (GPFS™)
- Redundant Array of Independent Disks (RAID) for local high availability
- Network File System (NFS) with replication

For more information about SAN, refer to the following IBM Redbooks publications:

- *IBM System Storage DS8000 Copy Services for Open Systems*, SG24-6788
- *IBM XIV Storage System: Copy Services and Migration*, SG24-7759
- *IBM System Storage DS Storage Manager Copy Services Guide*, SG24-7822
- *SAN Volume Controller and Storwize V7000 Replication Family Services*, SG24-7574

2.2.2 User authentication

All IBM SmartCloud Control Desk-supported user authentication mechanisms can be used in the topologies described.

Currently supported user authentication mechanisms are:

- External
  - IBM Tivoli Directory Server (ITDS)
  - Microsoft Active Directory
- Internal

  IBM SmartCloud Control Desk internal database authentication

If external authentication is in place, ensure that it is highly available to avoid a single point of failure. For disaster recovery, the users and groups should be replicated across sites.

Tip: For more information about Tivoli Directory Server high availability for IBM SmartCloud Control Desk, refer to

`http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=/com.ibm.mbs.doc/gp_highavail/c_ctr_ha_directory_server.html`

2.2.3 Load balancer

An external load balancer is suggested for environments with more than one web server and is required for the hybrid-active topology. There are several hardware and software solutions available for load balancing. Ensure that the load balancing solution is highly available to avoid a single point of failure. In a multisite active-passive topology, a load balancer can redirect users to the second site after failover occurs.

2.3 Configuration options

This publication covers three potential high availability and disaster recovery configuration scenarios. It is important to review and weigh the pros and cons of each configuration to help determine which solution is best for your organization.

Table 2-1 Topologies covered in this book

Topology type	Pros	Cons
Local high availability	► Fast failover times ► Protection from process and system failure ► WAN communication independent	► No protection from complete site failure ► Not a valid disaster recovery topology
Multisite active-passive	► Protection from complete site failure ► Database and file system replication can decrease complexity. ► Scheduled downtime for system maintenance can be decreased by switching to secondary site. ► Persisted integration transaction recovery on site failure ► Upgrades to the Tivoli Process Automation Engine itself are not included. Examples of such maintenance include middleware fixpack updates, OS updates, and hardware upgrades.	► Secondary site is unused while primary is active. ► High-speed reliable WAN link between sites for replication is required. ► Secondary site will increase system infrastructure costs.

Topology type	Pros	Cons
Multisite hybrid-active	► Protection from complete site failure ► Certain workloads can be distributed and shared amongst both sites.	► Potential increased licensing costs as secondary site will be operational ► Increased configuration complexity and potential for configuration error ► Remote database connection could decrease performance on secondary site. ► High-speed reliable WAN link between sites for replication is required. ► Upon site failure, the entire workload will now run on one site only and could impact performance. ► Potential loss of integration transactions on site failure with WebSphere Application Server Service Integration Bus.

An important factor when selecting a topology is the complexity of the configuration. When increasing complexity, you are also increasing the potential for misconfiguration or human error. Sometimes a more simple approach is ideal in a disaster scenario.

2.3.1 Software versions

This book outlines and provides examples of several configuration topologies for high availability and disaster recovery with IBM SmartCloud Control Desk. Although other software levels are supported, the topologies in this book are based on the following software versions:

- Operating system - SUSE Linux Enterprise Server 11 SP2
- Cluster manager - IBM Tivoli System Automation for Multiplatforms 3.2.2
- Directory server - IBM Tivoli Directory Server v6.3
- Web server - IBM HTTP Server v7 Fixpack 25
- Application server - IBM WebSphere Application Server Network Deployment v7 Fixpack 25
- Database servers
  - IBM DB2 Enterprise Server Edition v9.7 Fixpack 5
  - Oracle 11g Release 2

- Application
  - IBM SmartCloud Control Desk 7.5.0.0
  - IBM Tivoli Process Automation Engine 7.5.0.2
- Messaging - WebSphere MQ 7.0.1.9

More information: Supported software version information for IBM SmartCloud Control Desk can be found at the Product Configuration Matrix page:

http://www-01.ibm.com/support/docview.wss?uid=swg27014419

2.3.2 Local high availability topology

Implementing a local high availability solution helps keep critical applications running in a single site. Local high availability implies that redundancy exists for all IBM SmartCloud Control Desk and IBM Maximo middleware components required to keep the system fully operational.

Introducing service (virtual) Internet Protocol (IP) addresses and hostnames can provide transparency to connecting applications while eliminating the need for reconfiguration upon failover. A cluster manager, such as IBM Tivoli System Automation for Multiplatforms (SA MP), can be used to monitor and automate failover when a system or process fails.

While designing a local high availability topology it is important to identify and eliminate all single points of failure. Hardware and software failures should both be considered and accounted for.

Figure 2-1 shows an example local high availability topology.

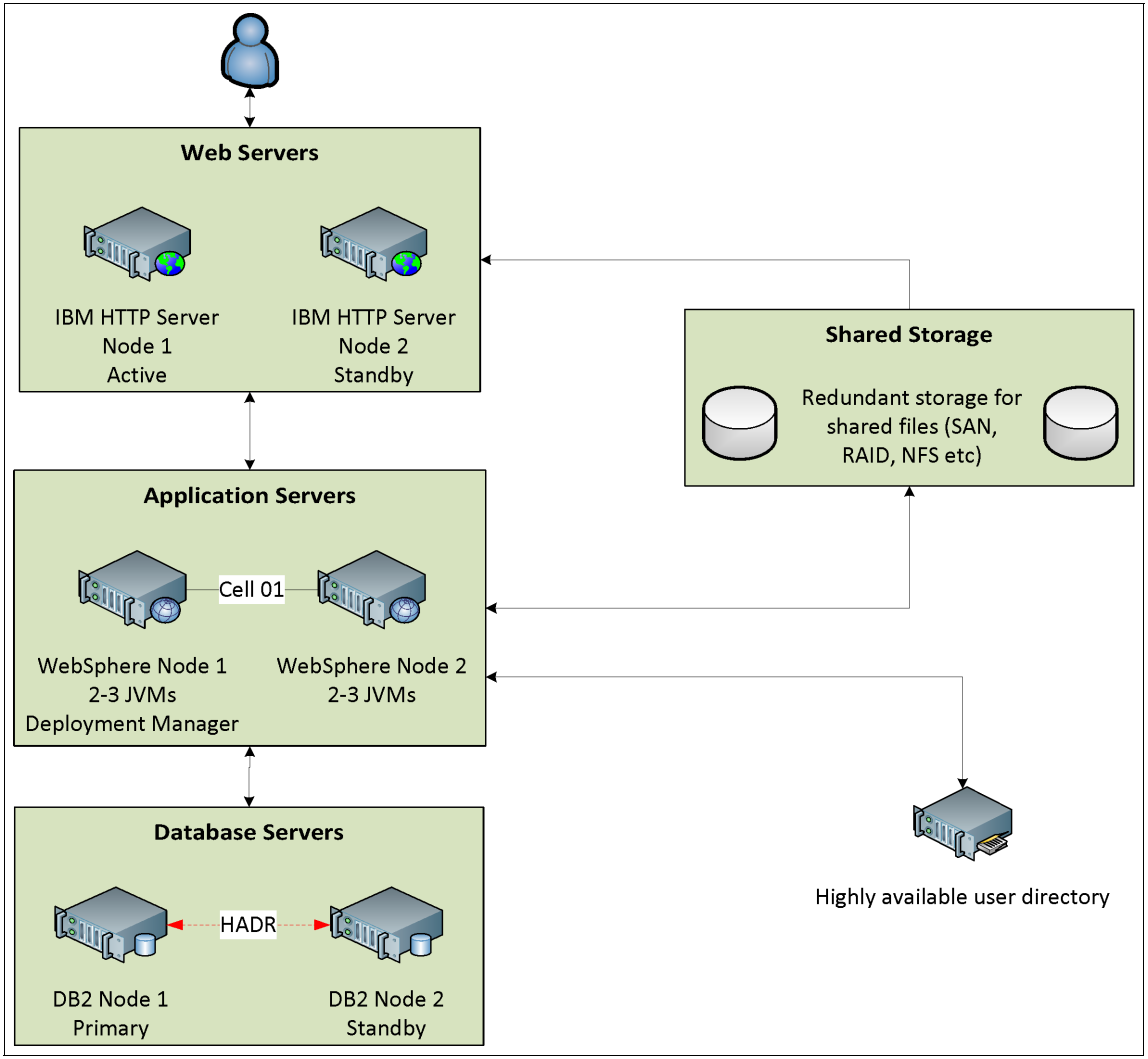

Figure 2-1 An example of a local high availability topology

More detailed configuration information and configuration instructions can be found in Chapter 3, “Local high availability topology” on page 25.

2.3.3 Multisite active-passive topology

The multisite active-passive topology consists of two sites that are connected by a high speed Wide Area Network (WAN) link. The secondary site is designed to

be passive, or offline, and does not process any transactions. The secondary site should be designed with the same capacity and storage as the primary site.

All the components that are necessary to ensure that the application is functional should be replicated to the secondary site. Replication techniques, such as file system replication using Storage Area Network (SAN) mirroring or any other disk storage mechanism, should be deployed to ensure that the secondary site is synchronized with the primary site at all times.

Consideration: Attachments and search indexes for IBM SmartCloud Control Desk should be stored on a replicated file system and mirrored to the secondary site.

A redundant database can be set up on the secondary site and synchronized using the mirroring techniques or using replication features provided by the database product. For example, a DB2 redundant database can be synchronized with the primary database using DB2 HADR (High Availability and Disaster Recovery) technology. Oracle Real Application Clusters (RACs) with Active Data Guard can be used to synchronize the database with a remote site.

For some organizations, a localized high availability solution does not provide enough protection and requires a secondary site for disaster recovery. The multisite active-passive topology can provide the benefits of a local high availability environment combined with the added protection from a complete site failure.

In the case of a disaster that would affect an entire site, a secondary site can be brought online to continue operations where the primary left off. Database and file-based replication mechanisms can be used to keep the standby site synchronized with the primary.

An active-passive topology can also help reduce scheduled downtime for system maintenance. If hardware upgrades or maintenance is required on the primary site, the standby site can be brought online to continue processing while the primary is down.

Figure 2-2 shows an example of an active-passive disaster recovery topology.

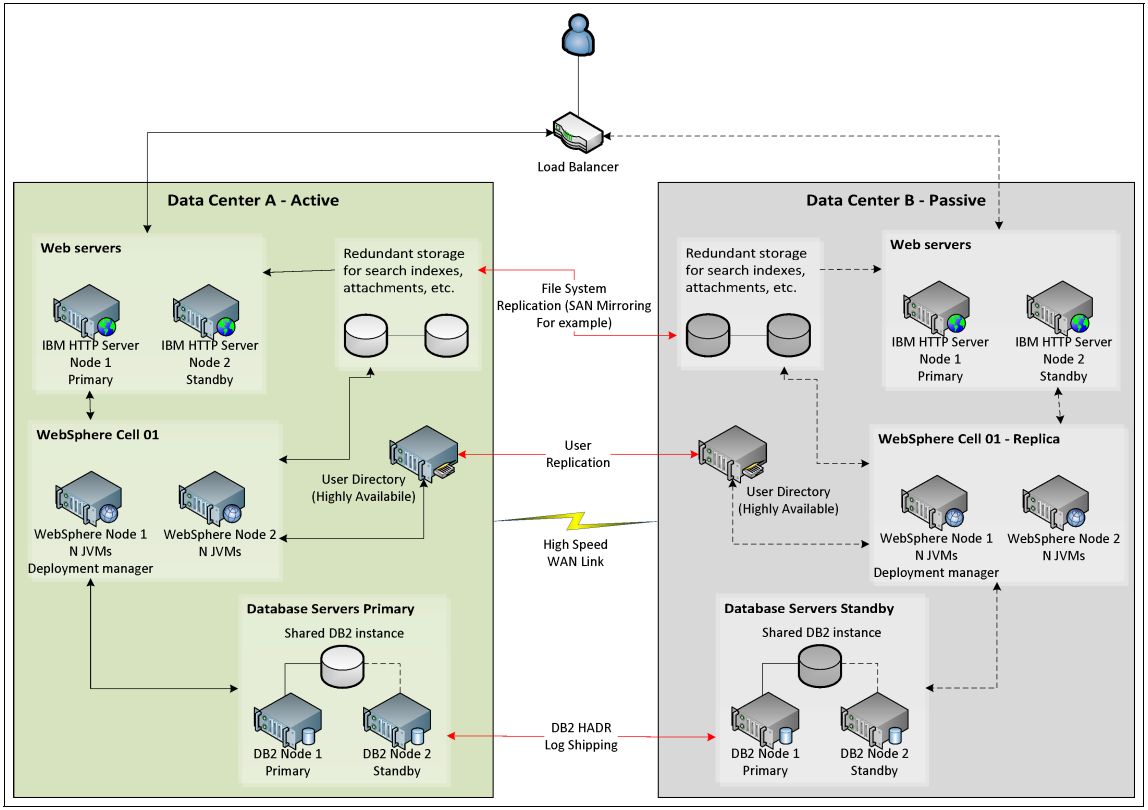

Figure 2-2 An example of a multisite active-passive topology

More detailed information and configuration instructions for a multisite active-passive topology can be found in Chapter 4, “Implementing a passive disaster recovery site” on page 149.

2.3.4 Multisite hybrid-active topology

The multisite hybrid-active topology consists of two sites that are connected by a high-speed WAN link. The secondary site is designed to be semi-active. This means the secondary site accepts transactions but all the database transactions are processed at the primary site. The secondary site should be designed with the same capacity and storage as the primary site.

All components that are necessary to ensure that the application is functional should be replicated to the secondary site using techniques as described here. The middleware layer for WebSphere Application Server consists of two or more

separate cells dispersed across the sites. The separate cells manage their own set of Java Virtual Machine (JVM) processes, which have to be maintained independently. All the JVMs process transactions using the primary database server. In case a primary site failure occurs, the JVMs can connect to the secondary database server, which is replicated with the primary database server. It is important to note that during normal operation the secondary database server is in passive mode—it cannot be configured in read-write mode.

Important: In this topology WebSphere Application Server consists of two or more separate cells. During a maintenance change window, care should be taken to make sure that the same updates are applied to all cells.

These topologies assume that there is a high-speed network link connecting both sites with mirroring techniques to keep the necessary data synchronized.

Often, organizations who invest in a secondary site for disaster recovery like to utilize these resources to take some of the processing load off the primary site. Certain resources can be brought online on the secondary site to help balance the load between the two sites. Although this scenario is often perceived to be ideal, the increased configuration complexity and additional licensing costs are often overlooked.

IBM SmartCloud Control Desk relies heavily on the database layer for transactions. If the secondary site's application is online, it needs to connect to the primary site's database because the standby database is only available in read-only mode. For this reason, this is not considered a true active-active configuration. This cross site connection to the database can cause performance problems due to network latency.

Let us look at some important considerations before choosing a hybrid-active topology.

- Licensing costs

  Licenses for products may be based on the active resources being utilized in the IT environment. In a hybrid-active configuration the licensing costs can increase because both sites utilize active resources at the same time.

  Important: Although this book touches on the topic of product licensing, official license information should be obtained from your IBM sales representative.

- Network communication between sites

  If two sites are simultaneously connecting to a single database, it is important that a high-speed reliable WAN link exists between them.

- Performance in a single site

  To be truly considered a full disaster recovery solution, each site should be able to handle the full load of all users and transactions. If one site were to fail and all requests routed to the single remaining site, then this single site needs to handle the load. Some organizations may decide that decreased performance in a disaster scenario is acceptable, but must be careful to ensure the increased load will not crash the systems.

- Integrations

  When integrating external systems with your IBM SmartCloud Control Desk solution, there may be communications that rely on the WebSphere Application Server's internal service integration bus. This bus is limited to the scope of a cell and requires a messaging engine to send and consume messages. It is not generally recommended to stretch a single cell across multiple sites, so there is the potential for lost or stuck transactions if both sites were processing at the time of failure. Cross-replication of the messaging engine datastore between both sites might help recover these transactions, but this greatly increases the complexity of the topology.

- Additional load balancing

  When implementing a hybrid-active topology, users and requests need to be balanced between the sites. It is usually ideal for this to be transparent to the users, and handled by an external load balancer. Hardware or software load balancers may need to be integrated into the environment to enable this feature.

Figure 2-3 shows an example of a hybrid-active topology that spans across two sites.

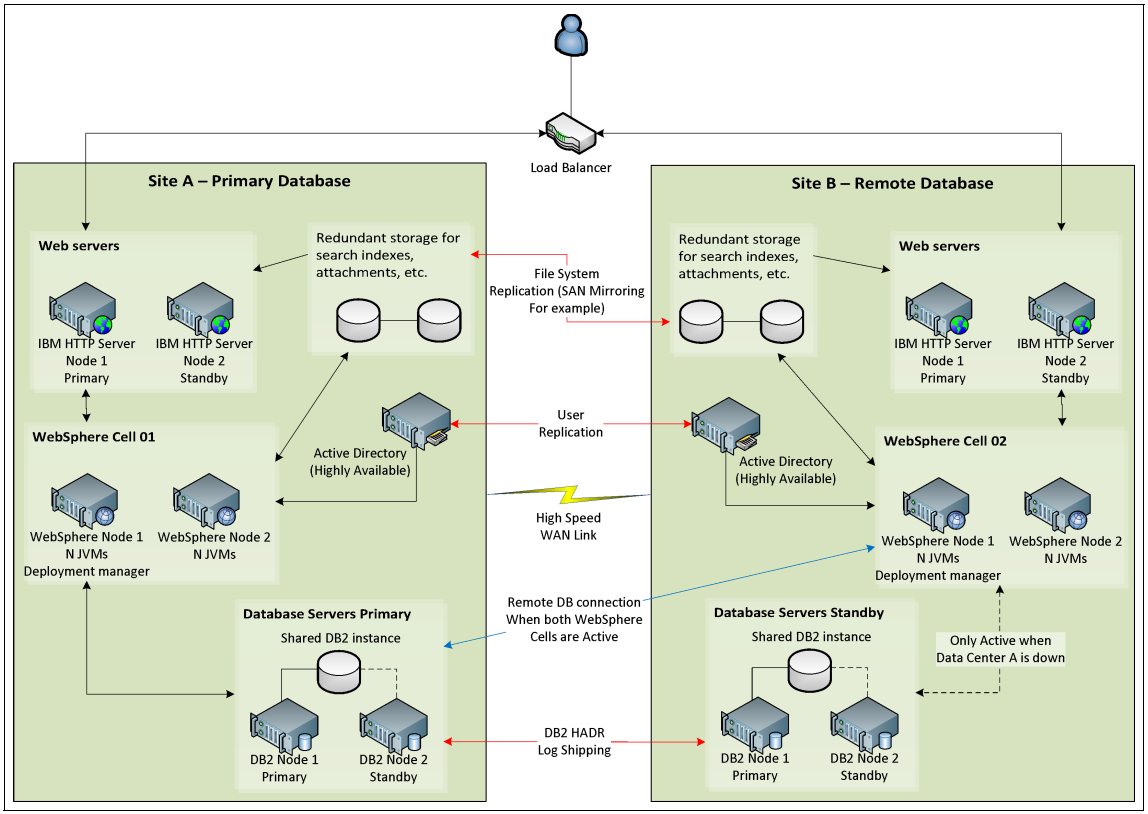

Figure 2-3 A high-level example of a multisite hybrid-active topology

More detailed information and configuration instructions for a multisite hybrid-active topology can be found in Chapter 5, “Utilizing multiple sites with a hybrid-active configuration” on page 189.

2.4 Conclusion

This chapter outlined the various configuration examples for high availability and disaster recovery for IBM SmartCloud Control Desk and IBM Maximo. By understanding the pros and cons of each topology, an organization can begin to design and implement a configuration that works for them.

Part 2

High availability and disaster recovery configuration

This part describes topologies and the configuration steps required for high availability and disaster recovery. Failover scenarios and testing are also covered.

3

Local high availability topology

In this chapter we provide information and examples of configuring a local high availability environment for IBM SmartCloud Control Desk.

- Introduction to local high availability
- Prerequisites
- Automated failover with a cluster manager
- Web server
- Application server
- Database
- IBM SmartCloud Control Desk
- Integration framework
- Failover testing

3.1 Introduction to local high availability

High availability implies redundancy that supports failover. Although local high availability does not provide the disaster protection of a multisite topology, it does provide protection from system, process and hardware failure. By defining and eliminating all single points of failure, you can strengthen your IBM SmartCloud Control Desk infrastructure and decrease system downtime.

Implementing a cluster manager into your topology will allow for automated failure detection and failover. The cluster manager is highly customizable and can dramatically decrease failover times versus manual methods. Introducing service (virtual) Internet Protocol (IP) addresses can mask these system failovers from connecting applications and users.

Local high availability is also a great place to start when trying to achieve a full disaster recovery plan. For example, when multiple sites are introduced, having local high availability on these sites can prevent the need to execute a full disaster recovery procedure when there is just a component failure on the overall solution.

Figure 3-1 on page 27 shows the topology configured in this chapter.

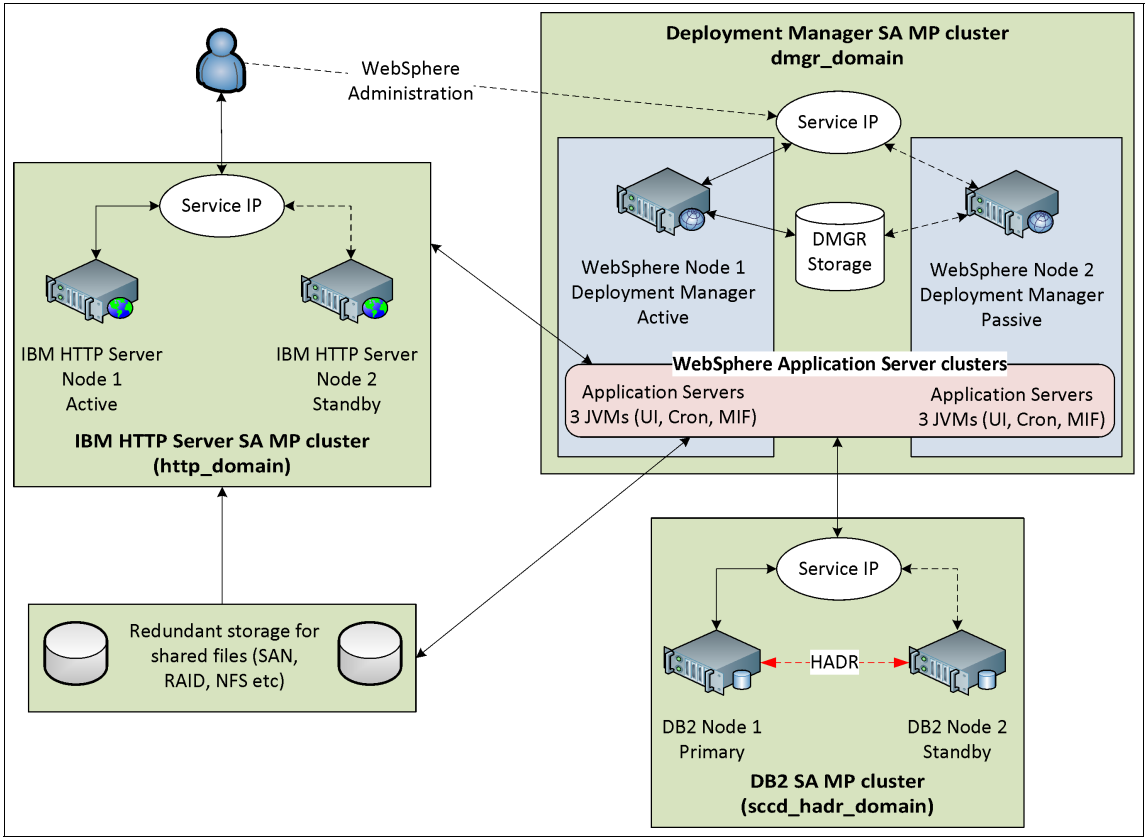

Figure 3-1 Example of local high availability topology configured in this chapter

3.2 Prerequisites

Before following the directions outlined here for configuring IBM SmartCloud Control Desk and middleware components, there are some prerequisites that should be in place:

- Storage

  A highly available storage solution should be in place for attachments, search index files, integration framework files and any other files that must be shared amongst the nodes. Redundancy through storage replication and local mirroring is a common method to ensure storage is not a single point of failure. There are many ways to create storage redundancy that are not covered specifically in this book.

- User directory

  A highly available user directory should be in place when using LDAP authentication with IBM SmartCloud Control Desk. Two supported LDAP servers are IBM Tivoli Directory Server (ITDS) and Microsoft Active Directory (MSAD). Examples of methods for achieving high availability with IBM Tivoli Directory Server can be found at:

  http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_ctr_ha_directory_server.html

- Licenses

  Appropriate software licenses for all products used are required. Middleware licensing should be confirmed with your IBM sales representative before implementing highly available environments.

- Service IP addresses and hostnames

  To allow components to failover and this process be transparent to connecting applications, service (virtual) IP addresses with associated hostnames should be available and defined in the Domain Name Server (DNS) servers for the network. The topology described in this chapter uses three service IPs.

- Additional servers

  Highly available environments will require additional hardware resources to create redundancy for all components. Each potential point of failure should have a backup system that can take over. The hardware and software specifications of the backup servers should be identical (or as close as possible) to the specifications of the primary servers to ensure they can successfully take over.

- Installation media

  The installation media for the outlined components should be obtained prior to the installation procedures.

3.3 Automated failover with a cluster manager

Multiple systems (nodes) are configured together as a cluster and can utilize a cluster manager to detect system status and make decisions based on this status. The cluster manager uses one or more resource groups with a defined set of policies to automate the failover procedure when a failure is detected. Introducing a service IP into the cluster configuration allows for transparency to the connecting applications and users and eliminates the need for configuration changes upon failover. The cluster manager itself uses detection methods such as heartbeats, tie breakers and quorum to determine system status and to make decisions as to how to act on this status.

3.3.1 Cluster manager concepts

Let us take a closer look at some of the components and concepts related to a cluster manager:

- Cluster

  A group of connected systems (nodes) that work together as a single functional system in the perspective of the user. Clustering allows servers to back each other up when failures occur by picking up the workload of the failed server.

- Cluster member

  A single node that is defined within the cluster

- Cluster manager

  An application or tool used to combine the nodes of a cluster and detect the status of the processes as defined by cluster resources and policies. The cluster manager drives the automated failover procedures and is highly configurable to the administrator. Many cluster managers are available that should function properly with IBM SmartCloud Control Desk, but our examples all use Tivoli System Automation for Muliplatforms as the cluster management tool.

- IP address takeover

  The ability to transfer an Ethernet interface's IP address from one machine to another when a server fails; to a client application, the two machines appear at different times to be the same server. Your cluster manager can be configured to automatically apply this service IP (or virtual IP) to the active node only. Upon failure, the cluster manager should remove the alias interface (and service IP along with it) from the primary node and apply it to the secondary node where services will be restored. Transactions and connections to the applications are always through the service IP address so the configuration of connecting components never has to change. In many cases, the startup of services is dependent on the application of the service IP.

- Heartbeating

  A communication mechanism implemented by the cluster manager on each node to allow each system to detect whether other cluster members are alive or down. The nodes will send heartbeats to each other as a low-level system status.

- Resources

  Applications or pieces of hardware that can be defined to the cluster manager. These can be manually defined, or some cluster managers such as Tivoli System Automation for Muliplatforms can use the concept of harvesting

which will allow it to detect and define some resources on the system, such as the Ethernet interface.

- Resource groups

  One or more defined resources lumped together as a functional group. This group serves a common purpose and the status of the resource group is related to the status of the members.

- Policy

  A set of instructions defined to the resource group that dictates what procedure will occur when failures are detected

- Relationships

  Relationships defined within the resources of a cluster. An example of a relationship would be a dependency. When implementing a service IP resource, other applications defined in the resource group may be dependent on the service IP being active. This dependency ensures that services start in the correct order.

- Quorum

  Determines which node is the active node and will process requests. There can be several types of quorums and they can function in specific ways depending on the number of nodes available in the cluster. To better understand the concept of a quorum, review the IBM Redbooks publication referenced at the end of this section.

- Tie breaker

  Determines which node has quorum and can access shared resources. When there is an even number of nodes within a cluster (commonly two) the cluster will require a tie breaker to determine which node will have quorum. An example of a tie breaker is a network tie breaker. If two nodes are connected and sending heartbeats but the connection is disrupted, no heartbeats will make it to either system. From the perspective of each node, they do not know whether it is the other node that has failed or whether its own Ethernet interface is down. The network tie breaker will attempt to ping a defined IP (you can have several defined) and the node that can ping the IP will win the tie breaker and have quorum.

More information: These were just a few of the many concepts related to cluster managers. For more information, review *End-to-end Automation with IBM Tivoli System Automation for Multiplatforms*, SG24-7117.

3.3.2 Tivoli System Automation for Multiplatforms

Tivoli System Automation for Multiplatforms is the cluster manager used in the local high availability configuration outlined in this chapter. Although System Automation for Multiplatforms is the official acronym for Tivoli System Automation for Multiplatforms, it is also referenced as TSAMP or TSA on many websites. Tivoli System Automation for Multiplatforms comes prebundled with IBM DB2 v9.5 and later for use with the database components. For other products such as IBM HTTP Server and IBM WebSphere Application Server, Tivoli System Automation for Multiplatforms must be *purchased* and *installed* separately. To install System Automation for Multiplatforms you can follow this procedure:

1. Obtain the Tivoli System Automation for Multiplatforms v3.2.2 installation media and license from IBM Passport advantage if you are entitled. You can also access a trial of Tivoli System Automation for Multiplatforms from:

   `http://www-01.ibm.com/software/tivoli/products/sys-auto-multi/`

2. Launch the `installSAM` executable from the Tivoli System Automation for Multiplatforms folder.
3. Follow the instructions to install System Automation for Multiplatforms.
4. Users that will be running System Automation for Multiplatforms for monitoring or start/stop of services, must **`export CT_MANAGEMENT_SCOPE=2`** as a global environment variable. It is suggested to put this export command into the startup profile for these users. Placing this command in `/etc/profile`, for example, ensures that this is exported globally at system startup. Running **`env | grep CT_MANAGEMENT_SCOPE`** should show that the variable is set.

As an optional step, there are some predefined policies you can install for System Automation for Multiplatforms that can be used for monitoring and configuring your automation. This example is for Linux and these policies can be found at:

`https://www-304.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW10SA02`

1. Download the policies from this link.
2. Install the RPM file by running:

   ```
   rpm -ivh sam.policies-1.3.3.0-1138.i386.rpm
   ```

3. The policy files should now exist in `/usr/sbin/rsct/sapolicies`.

Additional policies for other platforms and products can be found at:

`https://www-304.ibm.com/software/brandcatalog/ismlibrary/search?rc=TivoliSystemAutomation&catalog.start=0#rc=TivoliSystemAutomation#catalog.start=0`

Troubleshooting: If there are issues with the install of SAMP, or more information is needed, consult the System Automation for Multiplatforms Information Center at:

http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=%2Fcom.ibm.samp.doc_3.2.2%Fwelcome.html

Manual mode

When performing maintenance tasks on clusters controlled by SAMP, it may be desirable to put System Automation for Multiplatforms into manual mode, which allows you to stop and start services without System Automation for Multiplatforms interference. Issue the command **`samctrl -M t`** to enter manual mode. When maintenance is complete, **`samctrl -M f`** will enable System Automation for Multiplatforms automation once again.

3.4 Web server

The web server component is important in an IBM SmartCloud Control Desk environment because it acts as a single access point for users. It also provides load balancing capabilities to the application servers through the WebSphere Application Server Plug-in. Although the plug-in supports several different web servers, IBM HTTP Server is the IBM SmartCloud Control Desk supported solution. In a highly available topology, it is important that the HTTP Server is not a single point of failure. This section outlines the installation and configuration procedures for IBM HTTP Server high availability.

Important: After configuring the web server for high availability, make sure all web interactions (web services and the user interface) utilize the service IP address.

3.4.1 IBM HTTP Server

IBM HTTP Server (IHS) is the web server solution that comes bundled with the IBM SmartCloud Control Desk installation files. This topology example uses two separate IHS servers for web server high availability.

For this book the variables shown in Table 3-1 on page 33 are assumed. These values are not mandatory for all installations and might vary in other environments.

Table 3-1 Variables

Name	Description	Value
IHS_ROOT	IHS installation path	`/opt/IBM/HTTPServer`
IHS_SAMP_DOMAIN	IHS System Automation for Multiplatforms domain name	`http_domain`
IHS_SVC_IP	IHS service IP	`9.12.4.67`
ihshost1	Primary node hostname	`ti2022-l1.itso.ibm.com`
ihshost2	Standby node hostname	`ti2022-l2.itso.ibm.com`
GATEWAY_IP	Gateway IP	`9.12.4.1`

Installing IBM HTTP Server

Before you can configure IBM HTTP Server for high availability, you must install the product on both nodes.

1. Provision two systems (nodes) for the primary and standby IBM HTTP servers.
2. Obtain the IBM HTTP Server v7 and Plug-in installation media. This should have been included in the install package for IBM SmartCloud Control Desk; see Figure 3-2 on page 34.
3. On the first node, launch the installer launchpad.sh for IBM HTTP Server.
4. Proceed through the installation wizard. When the wizard asks if you would like to install the IBM HTTP Server Plug-in for IBM WebSphere Application Server, select this. Fill out the fields using a unique name for the web server definition. This example shows `webserver1`. Use the hostname for one of the IBM WebSphere Application server nodes that you will be using in 3.5.1, “WebSphere Application Server variables” on page 41. Click **Next**.

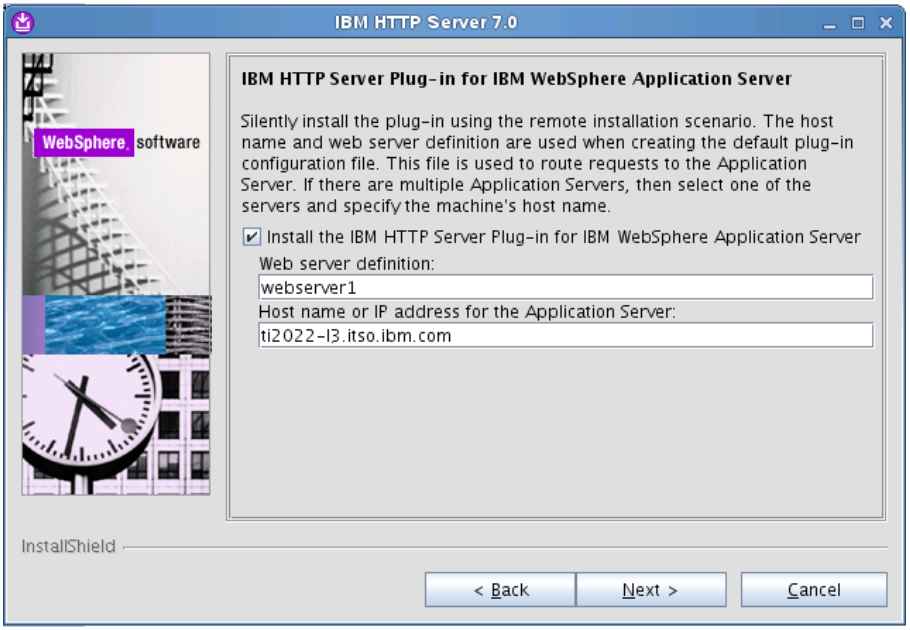

Figure 3-2 IBM HTTP Server Plug-in for WebSphere selection

5. A summary table will be displayed with the installation information. Review and click **Next**.
6. Repeat this procedure to install IBM HTTP Server v7 on the second node. Make sure that you use a unique name for the web server definition when installing the second server. This configuration example uses `webserver2` as the definition name.
7. On each server there should be a plug-in configuration file located in `IHS_ROOT/Plugins/bin` with the name of the web server definition created. For example, in step 4 a web server definition was created so the file `IHS_ROOT/Plugins/bin/configurewebserver1.sh` should now exist. On the second node, `configurewebserver2.sh` should exist. Make note of these files because they will be copied and used during “Federating web servers” on page 55.

Automating IBM HTTP Server failover with SAMP

Now that IHS is installed and configured you can use a cluster manager to automate failover. This example provides detailed instructions for how to use System Automation for Multiplatforms for automation.

1. Install Tivoli System Automation for Multiplatforms (SAMP) on both nodes. 3.3.2, “Tivoli System Automation for Multiplatforms” on page 31 outlines this procedure.
2. Prepare the servers to run in an System Automation for Multiplatforms domain. On both nodes run the command:

   preprpnode ihshost1 ihshost2
3. Create the System Automation for Multiplatforms domain. On one of the nodes run:

   mkrpdomain IHS_SAMP_DOMAIN ihshost1 ihshost2
4. Start the new System Automation for Multiplatforms domain. On one of the nodes run:

   startrpdomain IHS_SAMP_DOMAIN
5. Running the **lsrpdomain** command should show that your domain is listed and online; see Example 3-1.

   Example 3-1 lsrpdomain output

```
ti2022-l1:~ # lsrpdomain
Name        OpState RSCTActiveVersion MixedVersions TSPort GSPort
http_domain Online  3.1.1.3           No            12347  12348
```

   Now it is time to create the resource groups and resources for IHS.
6. Create a new `ihs.def` file or modify and use the existing one in `/usr/sbin/rsct/sapolicies/ihs`. Use the proper node names and ensure that the scripts in the monitor, stop and start command paths exist; Example 3-2.

   Example 3-2 An example ihs.def file

```
PersistentResourceAttributes::
Name="ihs-rs"
StartCommand="/usr/sbin/rsct/sapolicies/ihs/ihs start IHS_ROOT"
StopCommand="/usr/sbin/rsct/sapolicies/ihs/ihs stop IHS_ROOT"
MonitorCommand="/usr/sbin/rsct/sapolicies/ihs/ihs status IHS_ROOT"
MonitorCommandPeriod=10
MonitorCommandTimeout=5
NodeNameList={"ihshost1","ihshost2"}
StartCommandTimeout=11
StopCommandTimeout=11
UserName="root"
RunCommandsSync=1
ResourceType=1
```

Important: The MonitorCommand, StartCommand and StopCommand files referenced were installed with the optional System Automation for Multiplatforms policies from 3.3.2, “Tivoli System Automation for Multiplatforms” on page 31. If the optional policies were not installed, then this file will need to be created manually. Information about creating policies can be found at:

http://www.ibm.com/developerworks/wikis/display/tivoli/Tivoli+System+Automation+for+Multiplatforms+Best+Practices

7. Create or modify the existing definition file in `/usr/sbin/rsct/sapolicies/ihs` for the Service IP. Be sure to use the correct IP address (the service IP you will use), netmask and node hostnames for your environment; Example 3-3.

Example 3-3 An example ihsip.def

```
PersistentResourceAttributes::
Name="ihs-ip"
ResourceType=1
IPAddress=IHS_SVC_IP
NetMask=255.255.240.0
ProtectionMode=1
NodeNameList={"ihshost1","ihshost2"}
```

8. Create the resource group in System Automation for Multiplatforms for IHS by running:

```
mkrg ihs-rg
```

9. Create the IHS Application resource using your `ihs.def` file by running:

```
mkrsrc -f ihs.def IBM.Application
```

10. Add the IHS Application resource to the `ihs-rg` resource group by running:

```
addrgmbr -g ihs-rg IBM.Application:ihs-rs
```

11. Create the IHS ServiceIP resource using your `ihsip.def` file by running:

```
mkrsrc -f ihsip.def IBM.ServiceIP
```

12. Add the IHS ServiceIP resource to the `ihs-rg` resource group by running:

```
addrgmbr -g ihs-rg IBM.ServiceIP:ihs-ip
```

13. Create a network equivalency resource that can detect the status of the nodes’ Ethernet interfaces by running:

```
mkequ ihs-ip-equ IBM.NetworkInterface:eth0:ihshost1,eth0:ihshost2
```

14. Create a dependency relationship that specifies that the ServiceIP depends on the status of the network equivalency by running:

```
mkrel -p DependsOn -S IBM.ServiceIP:ihs-ip -G
IBM.Equivalency:ihs-ip-equ ihs-ip-rel-equ
```

15. Create a dependency relationship that specifies that the IHS application depends on the status of the ServiceIP by running:

```
mkrel -p DependsOn -S IBM.Application:ihs-rs -G IBM.ServiceIP:ihs-ip
ihs-rs-rel-ip
```

16. Now that all the resources and relationships are created, run the **lssam** command to view the status of the ihs-rg resource group.

```
ti2022-l1:~ # lssam
Offline IBM.ResourceGroup:ihs-rg Nominal=Offline
        |- Offline IBM.Application:ihs-rs
                |- Offline IBM.Application:ihs-rs:ti2022-l1
                '- Offline IBM.Application:ihs-rs:ti2022-l2
        '- Offline IBM.ServiceIP:ihs-ip
                |- Offline IBM.ServiceIP:ihs-ip:ti2022-l1
                '- Offline IBM.ServiceIP:ihs-ip:ti2022-l2
Online IBM.Equivalency:ihs-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l1
        '- Online IBM.NetworkInterface:eth0:ti2022-l2
```

Figure 3-3 Example lssam output after creating the resources

17. You can now issue the **chrg -o online ihs-rg** command, which will change the nominal status of the resource group to *online*. This will enable the service IP on the first node and bring the IHS application online.

18. Run **lssam** again. It may show pending online while the services are brought up. Running **lssam** again in a few moments should show that the application and service IP are online on the first node; Figure 3-4.

```
ti2022-l1:~ # lssam
Pending online IBM.ResourceGroup:ihs-rg Nominal=Online
        |- Pending online IBM.Application:ihs-rs
                |- Pending online IBM.Application:ihs-rs:ti2022-l1
                '- Offline IBM.Application:ihs-rs:ti2022-l2
        '- Online IBM.ServiceIP:ihs-ip
                |- Online IBM.ServiceIP:ihs-ip:ti2022-l1
                '- Offline IBM.ServiceIP:ihs-ip:ti2022-l2
Online IBM.Equivalency:ihs-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l1
        '- Online IBM.NetworkInterface:eth0:ti2022-l2
ti2022-l1:~ # lssam
Online IBM.ResourceGroup:ihs-rg Nominal=Online
        |- Online IBM.Application:ihs-rs
                |- Online IBM.Application:ihs-rs:ti2022-l1
                '- Offline IBM.Application:ihs-rs:ti2022-l2
        '- Online IBM.ServiceIP:ihs-ip
                |- Online IBM.ServiceIP:ihs-ip:ti2022-l1
                '- Offline IBM.ServiceIP:ihs-ip:ti2022-l2
Online IBM.Equivalency:ihs-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l1
        '- Online IBM.NetworkInterface:eth0:ti2022-l2
```

Figure 3-4 Example lssam output after switching the nominal status to online

19. If the ServiceIP and Application both show online in the status, then the IBM HTTP Server should be up and running on the primary node. The service IP should also be added as an alias to the Ethernet interface. Running **ifconfig** on the active node should show the alias; refer to Example 3-4.

Example 3-4 Example ifconfig with service IP applied (eth0:0 in this case)

```
ti2022-l1:~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:BC:70:A9
          inet addr:9.12.5.169  Bcast:9.12.15.255  Mask:255.255.240.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:92730 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24333 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8758460 (8.3 Mb)  TX bytes:2836407 (2.7 Mb)

eth0:0    Link encap:Ethernet  HWaddr 00:50:56:BC:70:A9
          inet addr:9.12.4.67  Bcast:9.12.15.255  Mask:255.255.240.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

20. You should now be able to access the IBM HTTP Server through `http://IHS_SVC_IP/`.

21. Issuing the **`rgreq -o move ihs-rg`** command switches the resources to the second node and they should go offline on the first node; see Figure 3-5.

```
ti2022-l1:~ # rgreq -o move ihs-rg
Action on resource group "ihs-rg" returned Token "0xab2c9171dffa66c3ad659950bd4e0b00" .
ti2022-l1:~ # lssam
Online IBM.ResourceGroup:ihs-rg Nominal=Online
        |- Online IBM.Application:ihs-rs
                |- Offline IBM.Application:ihs-rs:ti2022-l1
                '- Online IBM.Application:ihs-rs:ti2022-l2
        '- Online IBM.ServiceIP:ihs-ip
                |- Offline IBM.ServiceIP:ihs-ip:ti2022-l1
                '- Online IBM.ServiceIP:ihs-ip:ti2022-l2
Online IBM.Equivalency:ihs-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l1
        '- Online IBM.NetworkInterface:eth0:ti2022-l2
```

Figure 3-5 Example of lssam output after moving the resources to second node

Now that the resources and policy are configured, youl need to add a tie breaker. In a 2-node cluster configuration, when the nodes lose contact with each other, they will not be able to figure out which one failed and which one should obtain *quorum.* This example uses a *network tie breaker* to help resolve this problem. When specifying a network tie breaker, we use an IP address that should always be ping-able from the cluster nodes. The gateway (router) is usually a good candidate for this.

22. Create the tie breaker resource by running this command:

```
mkrsrc IBM.TieBreaker Type="EXEC" Name="networktb"
DeviceInfo='PATHNAME=/usr/sbin/rsct/bin/samtb_net
Address=GATEWAY_IP Log=1' PostReserveWaitTime=30;
```

23. Activate this network tie breaker resource in the domain by running:

```
chrsrc -c IBM.PeerNode OpQuorumTieBreaker="networktb"
```

24. You can verify the status of this tie breaker by running the **`lsrsrc -c IBM.PeerNode`** and **`lsrsrc -Ab IBM.TieBreaker`** commands; see Example 3-5.

Example 3-5 Example lsrsrc output for the tie breaker

```
ti2022-l1:~ # lsrsrc -c IBM.PeerNode
Resource Class Persistent Attributes for IBM.PeerNode
resource 1:
        CommittedRSCTVersion  = ""
        ActiveVersionChanging = 0
        OpQuorumOverride      = 0
        CritRsrcProtMethod    = 1
        OpQuorumTieBreaker    = "networktb"
        QuorumType            = 0
```

```
        QuorumGroupName       = ""
        Fanout                = 32
        OpFenceGroup          = ""
        NodeCleanupCommand    = ""
        NodeCleanupCriteria   = ""
ti2022-l1:~ # lsrsrc -Ab IBM.TieBreaker
Resource Persistent and Dynamic Attributes for IBM.TieBreaker
resource 1:
        Name                = "Fail"
        Type                = "Fail"
        DeviceInfo          = ""
        ReprobeData         = ""
        ReleaseRetryPeriod  = 0
        HeartbeatPeriod     = 0
        PreReserveWaitTime  = 0
        PostReserveWaitTime = 0
        NodeInfo            = {}
        ActivePeerDomain    = "http_domain"
        ConfigChanged       = 0
resource 2:
        Name                = "networktb"
        Type                = "EXEC"
        DeviceInfo          = "PATHNAME=/usr/sbin/rsct/bin/samtb_net
Address=9.12.4.1 Log=1"
        ReprobeData         = ""
        ReleaseRetryPeriod  = 0
        HeartbeatPeriod     = 0
        PreReserveWaitTime  = 0
        PostReserveWaitTime = 30
        NodeInfo            = {}
        ActivePeerDomain    = "http_domain"
        ConfigChanged       = 0
```

More information: Sometimes a network tie breaker is not the best solution or cannot be used. There are other types of tie breakers available. Consult Chapter 11 of the *SAMP Administrator's and User's Guide* found at:

http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=%2Fcom.ibm.samp.doc_3.2.2%Fwelcome.html

Troubleshooting

If the IHS or service IP resources do not start or show in *FAILED OFFLINE* status, you may have done something wrong when creating the resources.

The `/var/log/messages` file may have some information relating to the failed startup of the service. You can also manually run the start command **`/usr/sbin/rsct/sapolicies/ihs/ihs start IHS_ROOT`** to see whether it starts this way. If not, there is most likely a problem with your `ihs` script or the IHS itself. Starting IHS with **`IHS_ROOT/bin/apachectl start`** may also give some indication of the problem.

3.5 Application server

The application server is responsible for applying all business rules and acting as the interface between data and the user. IBM SmartCloud Control Desk is deployed and runs in this layer utilizing Java J2EE technologies.

The supported application servers for IBM SmartCloud Control Desk are IBM WebSphere Application Server and Oracle WebLogic Server. In this section we cover the setup steps to enable local high availability topology with *WebSphere Application Server*, where we take a closer look at the following details:

- WebSphere Application Server variables
- WebSphere Application Server internal architecture
- Installing WebSphere Application Server
- Installing deployment manager
- Automating deployment manager failover with SA MP
- Troubleshooting
- Installing application server profile on nodes
- Automating nodeagent restart with SA MP
- Federating web servers
- Cluster configuration

3.5.1 WebSphere Application Server variables

For this book we used the variables shown in Table 3-2. These values are not mandatory for all installations and might vary in other environments.

Table 3-2 Variables

Name	Description	Value
WAS_HOME	WebSphere Application Server installation path	`/opt/IBM/WebSphere/AppServer`
WAS_DMGR_PATH	Deployment manager profile installation path	`/opt/was_dmgr_files/profiles/Dmgr01`
WAS_DMGR_SVC_IP	Deployment manager service IP	`9.12.4.152`

Name	Description	Value
WAS_SAMP_DOMAIN	System Automation for Multiplatforms domain name	dmgr_domain
washost1	Node 1 hostname	ti2022-l3.itso.ibm.com
washost2	Node 2 hostname	ti2022-l4.itso.ibm.com
GATEWAY_IP	Gateway IP	9.12.4.1

3.5.2 WebSphere Application Server internal architecture

To provide a highly available solution, the WebSphere Application Server will be configured as shown in Figure 3-6.

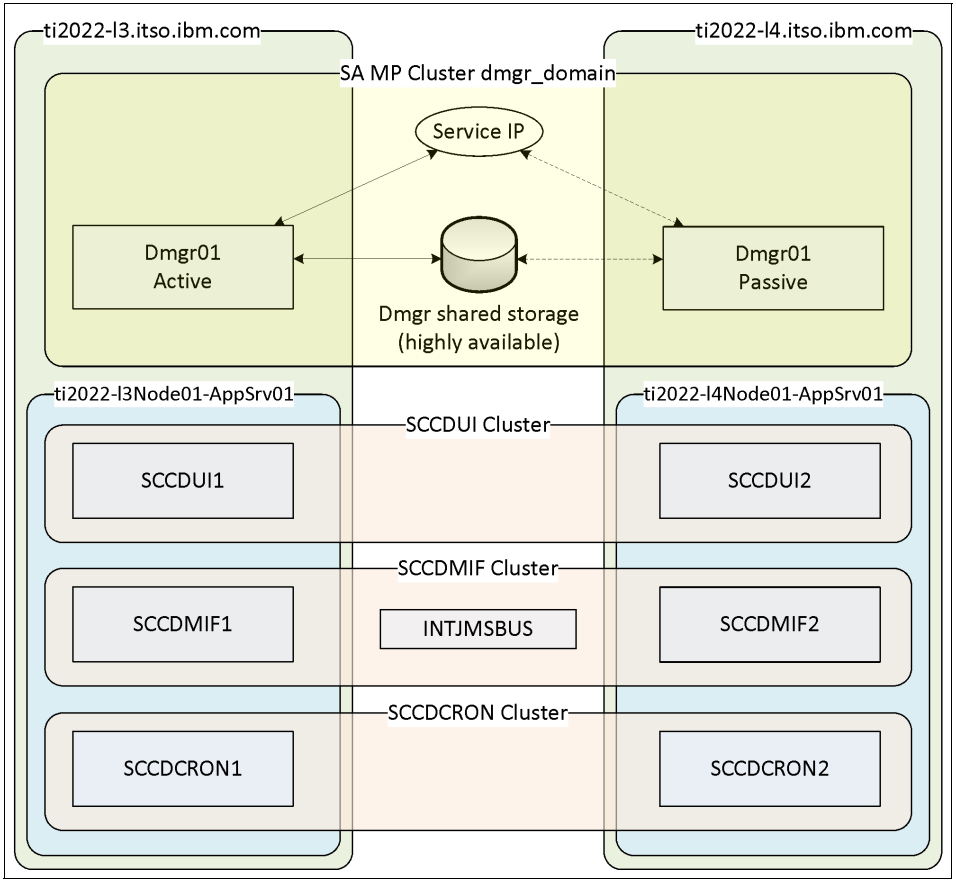

Figure 3-6 WebSphere Application Server internal architecture

3.5.3 Installing WebSphere Application Server

Before configuring WebSphere Application Server for high availability, the product must be installed on all nodes. If any profile was created during installation, remove it using the **manageprofiles** command.

Tip: For more information about the **manageprofiles** command, refer to:

http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rxml_manageprofiles.html

3.5.4 Installing deployment manager

In order to provide a highly available deployment manager (Dmgr), the installation must be done on a shared disk. This shared disk must be mounted on the same path on all nodes. The path chosen for this book is the variable `WAS_DMGR_PATH`. Make sure that the shared disk is highly available to avoid a single point of failure.

The installation of the deployment manager profile as `Dmgr01` may only occur in one node. Server `washost1` will be used as shown in Example 3-6.

Example 3-6 washost1 deployment manager installation

```
ti2022-l3:WAS_HOME/bin # ./manageprofiles.sh \
>   -create \
>     -profileName Dmgr01 \
>     -profilePath $WAS_DMGR_PATH \
>     -templatePath $WAS_HOME/profileTemplates/management \
>     -serverType DEPLOYMENT_MANAGER \
>     -enableAdminSecurity true \
>     -adminUserName admin \
>     -adminPassword admin
INSTCONFSUCCESS: Success: Profile Dmgr01 now exists. Please consult
WAS_DMGR_PATH/logs/AboutThisProfile.txt for more information about this
profile.
```

After installing deployment manager, start it with the **startManager** command, validate and apply security configurations as needed.

Important: If IBM SmartCloud Control Desk installation will be used, the deployment manager install path must end with `/profiles/Dmgr01`.

3.5.5 Automating deployment manager failover with SA MP

Now that the deployment manager is installed and configured, you can use a cluster manager to automate failover. This example provides detailed instructions for using System Automation for Multiplatforms for automation.

1. Install System Automation for Multiplatforms on both nodes. 3.3.2, “Tivoli System Automation for Multiplatforms” on page 31 outlines this procedure.
2. Prepare the servers to run in a System Automation for Multiplatforms domain. On both nodes run the command:

   preprpnode washost1 washost2
3. Create the System Automation for Multiplatforms domain. On one of the nodes run:

   mkrpdomain WAS_SAMP_DOMAIN washost1 washost2
4. Start the new System Automation for Multiplatforms domain. On one of the nodes run:

   startrpdomain WAS_SAMP_DOMAIN
5. Running the **lsrpdomain** command should show that your domain is listed and online; Example 3-7.

   Example 3-7 lsrpdomain output

   ```
   ti2022-l3:~ # lsrpdomain
   Name        OpState RSCTActiveVersion MixedVersions TSPort GSPort
   dmgr_domain Online  3.1.1.3           No            12347  12348
   ```

   Now it is time to create the resource groups and resources for the deployment manager.
6. If administrative security is enabled, modify the scripts installed by System Automation for Multiplatforms policies in 3.3.2, “Tivoli System Automation for Multiplatforms” on page 31. The original script, /usr/sbin/rsct/sapolicies/was/wasctrl-dmgr, needs to be modified to support username and password.

   a. Add the following between lines 40 and 41 (Example 3-8).

      Example 3-8 Additional parameters

      ```
      USER=$3
      PASSWORD=$4
      ```

b. Replace the current `stop_server` function with the example in Example 3-9.

Example 3-9 Modified function stop_server

```
function stop_server
{
        if [ ! "$USER" = "" ]; then
           DMGR_OPTS='-user '$USER' -password '$PASSWORD
        fi
        if [ "$OS" = "Linux" ]; then
           ${DMGR_HOME}/bin/stopManager.sh -timeout 180 $DMGR_OPTS
        else
           ${DMGR_HOME}/bin/stopManager.sh -quiet $DMGR_OPTS
        fi
        RC=$?
}
```

7. Create a new `dmgr-jvm.def` file with System Automation for Multiplatforms parameters. Use the proper node names and ensure that the scripts in the monitor, stop and start command paths exist.

Example 3-10 An example dmgr-jvm.def file

```
PersistentResourceAttributes::
Name=dmgr-jvm
ResourceType=1
StartCommand=/usr/sbin/rsct/sapolicies/was/wasctrl-dmgr start WAS_DMGR_PATH
StopCommand=/usr/sbin/rsct/sapolicies/was/wasctrl-dmgr stop WAS_DMGR_PATH admin admin
MonitorCommand=/usr/sbin/rsct/sapolicies/was/wasctrl-dmgr status WAS_DMGR_PATH
StartCommandTimeout=120
StopCommandTimeout=60
MonitorCommandTimeout=9
MonitorCommandPeriod=30
ProtectionMode=1
NodeNameList={'washost1','washost2'}
UserName=root
```

Important: The MonitorCommand, StartCommand and StopCommand files referenced were installed with the optional System Automation for Multiplatforms policies from 3.3.2, “Tivoli System Automation for Multiplatforms” on page 31. If the optional policies were not installed, then these file need to be created manually. Information about creating policies can be found at:

http://www.ibm.com/developerworks/wikis/display/tivoli/Tivoli+System+Automation+for+Multiplatforms+Best+Practices

8. Create a new `dmgr-ip.def` file for the Service IP. Be sure to use the correct IP address (the service IP you will use), netmask and node hostnames for your environment.

Example 3-11 An example dmgr-ip.def file

```
PersistentResourceAttributes::
Name="dmgr-ip"
ResourceType=1
IPAddress=WAS_DMGR_SVC_IP
NetMask=255.255.240.0
ProtectionMode=1
NodeNameList={"washost1","washost2"}
```

9. Create the resource group in System Automation for Multiplatforms for the deployment manager by running:

```
mkrg dmgr-rg
```

10. Create the deployment manager `Application` resource using your `dmgr-jvm.def` file by running:

```
mkrsrc -f dmgr-jvm.def IBM.Application
```

11. Add the deployment manager `Application` resource to the `dmgr-rg` resource group by running:

```
addrgmbr -g dmgr-rg IBM.Application:dmgr-jvm
```

12. Create the deployment manager `ServiceIP` resource using your `dmgr-ip.def` file by running:

```
mkrsrc -f dmgr-ip.def IBM.ServiceIP
```

13. Add the deployment manager `ServiceIP` resource to the `dmgr-rg` resource group by running:

```
addrgmbr -g dmgr-rg IBM.ServiceIP:dmgr-ip
```

14. Create a network equivalency resource that can detect the status of the nodes’ Ethernet interfaces by running:

```
mkequ dmgr-ip-equ IBM.NetworkInterface:eth0:washost1,eth0:washost2
```

15. Create a dependency relationship which specifies that the `ServiceIP` depends on the status of the network equivalency by running:

```
mkrel -p DependsOn -S IBM.ServiceIP:dmgr-ip -G
IBM.Equivalency:dmgr-ip-equ dmgr-ip-rel-equ
```

16. Create a dependency relationship which specifies that the deployment manager depends on the status of the `ServiceIP` by running:

```
mkrel -p DependsOn -S IBM.Application:dmgr-jvm -G
IBM.ServiceIP:dmgr-ip dmgr-jvm-rel-ip
```

17. Now that all the resources and relationships are created, run the **`lssam`** command to view the status of the `dmgr-rg` resource group, as shown in Figure 3-7.

```
ti2022-l3:~ # lssam
Offline IBM.ResourceGroup:dmgr-rg Nominal=Offline
        |- Offline IBM.Application:dmgr-jvm
                 |- Offline IBM.Application:dmgr-jvm:ti2022-l3
                 '- Offline IBM.Application:dmgr-jvm:ti2022-l4
        '- Offline IBM.ServiceIP:dmgr-ip
                 |- Offline IBM.ServiceIP:dmgr-ip:ti2022-l3
                 '- Offline IBM.ServiceIP:dmgr-ip:ti2022-l4
Online IBM.Equivalency:dmgr-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l3
        '- Online IBM.NetworkInterface:eth0:ti2022-l4
ti2022-l3:~ #
```

Figure 3-7 Example lssam output after creating the resources

18. Ensure that the deployment manager is down and update its hostname to use the service IP using the `wsadmin` tool, as shown in Example 3-12.

Example 3-12 Updating deployment manager's hostname

```
ti2022-l3:WAS_DMGR_PATH/bin # ./wsadmin.sh -lang jython -conntype NONE
WASX7357I: By request, this scripting client is not connected to any server process.
Certain configuration and application operations will be available in local mode.
WASX7031I: For help, enter: "print Help.help()"
wsadmin>AdminTask.changeHostName('[-interactive]')
Change Host Name

Change the host name of a node

*Node Name (nodeName): washost1CellManager01
*Host Name (hostName): WAS_DMGR_SVC_IP
System Name (systemName):
```

```
Regenerate Certificates (regenDefaultCert):

Change Host Name

F (Finish)
C (Cancel)

Select [F, C]: [F] F
WASX7278I: Generated command line: AdminTask.changeHostName('[-nodeName
ti2022-l3CellManager01 -hostName zdomino.itso.ibm.com ]')
''
wsadmin>AdminConfig.save()
''
wsadmin>exit
```

19. You can now issue the **chrg -o online dmgr-rg** command, which will change the nominal status of the resource group to *online*. This will enable the service IP on the first node and bring the deployment manager online.

Important: After activating System Automation for Multiplatforms for WebSphere Application Server deployment manager, all **start** and **stop** commands must be issued through SA MP. The **startManager** and **stopManager** commands should only be used in System Automation for Multiplatforms *manual* mode.

20. Run **lssam** again. It may show pending online while the services are brought up. Running **lssam** again in a few moments should show that the application and service IP are online on the first node, as shown in Figure 3-8.

```
ti2022-l3:~ # lssam
Pending online IBM.ResourceGroup:dmgr-rg Request=Move Nominal=Online
        |- Pending online IBM.Application:dmgr-jvm
                |- Pending online IBM.Application:dmgr-jvm:ti2022-l3
                '- Offline IBM.Application:dmgr-jvm:ti2022-l4
        '- Online IBM.ServiceIP:dmgr-ip
                |- Online IBM.ServiceIP:dmgr-ip:ti2022-l3
                '- Offline IBM.ServiceIP:dmgr-ip:ti2022-l4
Online IBM.ResourceGroup:nodeagent-ti2022-l3-rg Nominal=Online
        '- Online IBM.Application:nodeagent-ti2022-l3
                '- Online IBM.Application:nodeagent-ti2022-l3:ti2022-l3
Online IBM.ResourceGroup:nodeagent-ti2022-l4-rg Nominal=Online
        '- Online IBM.Application:nodeagent-ti2022-l4
                '- Online IBM.Application:nodeagent-ti2022-l4:ti2022-l4
Online IBM.Equivalency:dmgr-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l3
        '- Online IBM.NetworkInterface:eth0:ti2022-l4
```

Figure 3-8 Example lssam output after switching the nominal status to online

21. If the ServiceIP and Application both show online in the status, then the deployment manager should be up and running on the primary node. The service IP should also be added as an alias to the Ethernet interface. Running **ifconfig** on the active node should show the alias; see Example 3-13 on page 49.

Example 3-13 Example ifconfig with service IP applied (eth0:0 in this case)

```
ti2022-l3:~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:BC:70:AB
          inet addr:9.12.5.152  Bcast:9.12.15.255  Mask:255.255.240.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8364877 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12631835 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000

eth0:0    Link encap:Ethernet  HWaddr 00:50:56:BC:70:AB
          inet addr:9.12.4.152  Bcast:9.12.15.255  Mask:255.255.240.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

22. You should now be able to access the Integrated Solutions Console through:

https://WAS_DMGR_SVC_IP:9043/ibm/console

23. Issuing the **rgreq -o move dmgr-rg** command will switch the resources to the second node—they should go offline on the first node and online on the second node; see Figure 3-9.

```
ti2022-l3:~ # rgreq -o move dmgr-rg
Action on resource group "dmgr-rg" returned Token "0xac8b349f59234681230593503
9510400" .
ti2022-l3:~ # lssam
Online IBM.ResourceGroup:dmgr-rg Nominal=Online
        |- Online IBM.Application:dmgr-jvm
                |- Offline IBM.Application:dmgr-jvm:ti2022-l3
                '- Online IBM.Application:dmgr-jvm:ti2022-l4
        '- Online IBM.ServiceIP:dmgr-ip
                |- Offline IBM.ServiceIP:dmgr-ip:ti2022-l3
                '- Online IBM.ServiceIP:dmgr-ip:ti2022-l4
Online IBM.Equivalency:dmgr-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l3
        '- Online IBM.NetworkInterface:eth0:ti2022-l4
ti2022-l3:~ #
```

Figure 3-9 Example of lssam output after moving the resources to the second node

Now that the resources and policy are configured, add a tie breaker. When specifying a network tie breaker, use an IP address that should always be ping-able from the cluster nodes. The gateway (router) is usually a good candidate for this.

24. Create the tie breaker resource by running this command:

```
mkrsrc IBM.TieBreaker Type="EXEC" Name="networktb"
DeviceInfo='PATHNAME=/usr/sbin/rsct/bin/samtb_net Address=GATEWAY_IP
Log=1' PostReserveWaitTime=30
```

25. Activate this network tie breaker resource in the domain by running:

```
chrsrc -c IBM.PeerNode OpQuorumTieBreaker="networktb"
```

For more information about cluster management, refer to 3.3.1, “Cluster manager concepts” on page 29.

3.5.6 Troubleshooting

If the deployment manager or service IP resources do not start or show in the FAILED OFFLINE status, you may have done something wrong when creating the resources.

The `/var/log/messages` file may have some information relating to the failed startup of the service. You can also manually run the start command **`/usr/sbin/rsct/sapolicies/was/wasctrl-dmgr start WAS_DMGR_PATH`** to see if it starts this way. If not, there is most likely a problem with your `wasctrl-dmgr`

script or the deployment manager itself. Starting the deployment manager with **WAS_DMGR_PATH/bin/startManager.sh** may also give some indication of the problem.

3.5.7 Installing application server profile on nodes

After installing the deployment manager profile, install the node profile for all nodes for further cluster creation. An application server profile named AppSrv01 is created for both, as shown in Example 3-14.

Example 3-14 washost1 application server profile installation

```
ti2022-l3:WAS_HOME/bin # ./manageprofiles.sh \
>   -create \
>     -profileName AppSrv01
INSTCONFSUCCESS: Success: Profile AppSrv01 now exists. Please consult
WAS_HOME/profiles/AppSrv01/logs/AboutThisProfile.txt for more
information about this profile.
```

Run the same command on all nodes.

After installing both application server profiles, they must be federated with deployment manager. To accomplish this, start deployment manager with the **chrg -o online dmgr-rg** command.

After starting deployment manager, run the **addNode** command on all nodes. To federate the washost1 node is shown in Example 3-15.

Example 3-15 washost1 node federation

```
ti2022-l3:WAS_HOME/profiles/AppSrv01/bin # ./addNode.sh \
>   washost1 \
>     -username admin \
>     -password admin
ADMU0116I: Tool information is being logged in file
WAS_HOME/profiles/AppSrv01/logs/addNode.log
ADMU0128I: Starting tool with the AppSrv01 profile
CWPKI0308I: Adding signer alias "CN=ti2022-l3.itso.ibm.com, OU=R" to local
           keystore "ClientDefaultTrustStore" with the following SHA digest:
           B5:28:1E:4A:7C:13:E6:DC:A0:6F:D6:70:1D:15:45:32:E7:A2:0C:9F
CWPKI0308I: Adding signer alias "datapower" to local keystore
           "ClientDefaultTrustStore" with the following SHA digest:
           A9:BA:A4:B5:BC:26:2F:5D:2A:80:93:CA:BA:F4:31:05:F2:54:14:17
ADMU0001I: Begin federation of node ti2022-l3Node01 with Deployment Manager at
           ti2022-l3:8879.
```

```
ADMU0009I: Successfully connected to Deployment Manager Server: ti2022-l3:8879
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU2010I: Stopping all server processes for node ti2022-l3Node01
ADMU0512I: Server server1 cannot be reached. It appears to be stopped.
ADMU0024I: Deleting the old backup directory.
ADMU0015I: Backing up the original cell repository.
ADMU0012I: Creating Node Agent configuration for node: ti2022-l3Node01
ADMU0014I: Adding node ti2022-l3Node01 configuration to cell: ti2022-l3Cell01
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0018I: Launching Node Agent process for node: ti2022-l3Node01
ADMU0020I: Reading configuration for Node Agent process: nodeagent
ADMU0022I: Node Agent launched. Waiting for initialization status.
ADMU0030I: Node Agent initialization completed successfully. Process id is:
           25345

ADMU0300I: The node ti2022-l3Node01 was successfully added to the
           ti2022-l3Cell01 cell.

ADMU0306I: Note:
ADMU0302I: Any cell-level documents from the standalone ti2022-l3Cell01
           configuration have not been migrated to the new cell.
ADMU0307I: You might want to:
ADMU0303I: Update the configuration on the ti2022-l3Cell01 Deployment Manager
           with values from the old cell-level documents.

ADMU0306I: Note:
ADMU0304I: Because -includeapps was not specified, applications installed on
           the standalone node were not installed on the new cell.
ADMU0307I: You might want to:
ADMU0305I: Install applications onto the ti2022-l3Cell01 cell using wsadmin
           $AdminApp or the Administrative Console.

ADMU0003I: Node ti2022-l3Node01 has been successfully federated.
```

Repeat these steps for all nodes in the topology.

3.5.8 Automating nodeagent restart with SA MP

Now that the nodeagents are installed and configured, you can use a cluster manager to automate restart. This example provides detailed instructions on how to use System Automation for Multiplatforms for automation.

1. Install System Automation for Multiplatforms on both nodes. 3.3.2, “Tivoli System Automation for Multiplatforms” on page 31 outlines this procedure.
2. Prepare the servers to run in an System Automation for Multiplatforms domain. On both nodes run the command:

   preprpnode washost1 washost2
3. If administrative security is enabled, modify the scripts installed by System Automation for Multiplatforms policies in 3.3.2, “Tivoli System Automation for Multiplatforms” on page 31. The original script, /usr/sbin/rsct/sapolicies/was/wasctrl-na, needs to be modified to support username and password:

   a. Add the lines shown in Example 3-16 between line 60 and 61.

   Example 3-16 Additional parameters

```
NA_USER=$5
NA_PASSWORD=$6
```

   b. Replace the current stop case statement shown in Example 3-17. The example only shows an excerpt of the complete script file, and the “...” placeholder represents content that has been left out intentionally.

Example 3-17 Modified stop case statement

```
...
stop)
                ${LOGGER} -i -p info -t $0 "NodeAgent stopping..."
                if [ ! "$NA_USER" = "" ]; then
                  NA_OPTS='-user '${NA_USER}' -password '${NA_PASSWORD}
                fi
                if [ "$OS" = "Linux" ]; then
                  ${NA_HOME}/bin/stopNode.sh -timeout 180 ${NA_OPTS}
                else
                  ${NA_HOME}/bin/stopNode.sh -quiet ${NA_OPTS}
                fi
                RC=$?
...
```

4. Create a new nodeagent-washost1.def file with System Automation for Multiplatforms parameters. Use the proper node names and ensure that the scripts in the monitor, stop and start command paths exist; see Example 3-18.

Example 3-18 An example nodeagent-washost1.def file

```
PersistentResourceAttributes::
Name=nodeagent-washost1
ResourceType=1
StartCommand=/usr/sbin/rsct/sapolicies/was/wasctrl-na start
WAS_HOME/profiles/AppSrv01 8878
StopCommand=/usr/sbin/rsct/sapolicies/was/wasctrl-na stop WAS_HOME/profiles/AppSrv01
8878 nodeagent admin admin
MonitorCommand=/usr/sbin/rsct/sapolicies/was/wasctrl-na status
WAS_HOME/profiles/AppSrv01 8878
StartCommandTimeout=60
StopCommandTimeout=60
MonitorCommandTimeout=19
MonitorCommandPeriod=30
ProtectionMode=1
RunCommandsSync=0
NodeNameList={'washost1'}
UserName=root
```

5. Create the resource group in System Automation for Multiplatforms for the washost1 nodeagent by running:

   ```
   mkrg nodeagent-washost1-rg
   ```

6. Create the washost1 nodeagent `Application` resource using your `nodeagent-washost1.def` file by running:

   ```
   mkrsrc -f nodeagent-washost1.def IBM.Application
   ```

7. Add the washost1 nodeagent `Application` resource to the `nodeagent-washost1-rg` resource group by running:

   ```
   addrgmbr -g nodeagent-washost1-rg IBM.Application:nodeagent-washost1
   ```

8. You can now issue the **`chrg -o online nodeagent-washost1-rg`** command, which will change the nominal status of the resource group to *online*. This will bring the nodeagent online.

   Important: After activating System Automation for Multiplatforms for the WebSphere Application Server nodeagent, all **`start`** and **`stop`** commands must be issued through SA MP. The **`startNode`**, **`stopNode`** and Integrated Solutions Console commands should only be used in System Automation for Multiplatforms *manual* mode.

Repeat these steps for all nodes in the topology.

3.5.9 Federating web servers

Federating the web servers into the WebSphere Deployment Manager allows them to be mapped to the Tivoli Process Automation Engine modules during deployment. Having the web servers in WebSphere also allows for generating and propagating the web server plug-ins for load balancing.

Copy the configuration scripts generated during IBM HTTP Server installation in “Installing IBM HTTP Server” on page 33 to `WAS_DMGR_PATH/bin`. Run all configuration files copied as shown in Example 3-19.

Example 3-19 webserver1 federation

```
ti2022-l3:WAS_DMGR_PATH/bin # ./configurewebserver1.sh \
>   -profileName Dmgr01 \
>   -user admin \
>   -password admin \
>   -ihsAdminPassword admin
Using the profile Dmgr01
Using WAS admin userID admin
WASX7209I: Connected to process "dmgr" on node ti2022-l3CellManager01
using SOAP connector;  The type of process is: DeploymentManager
WASX7303I: The following options are passed to the scripting
environment and are available as arguments that are stored in the argv
variable: "[webserver1, IHS, IHS_ROOT, IHS_ROOT/conf/httpd.conf, 80,
MAP_ALL, IHS_ROOT/Plugins, unmanaged, ti2022-l1.itso.ibm.com-node,
ti2022-l1.itso.ibm.com, linux, 8008, httpadmin, admin]"

Input parameters:

   Web server name             - webserver1
   Web server type             - IHS
   Web server install location - IHS_ROOT
   Web server config location  - IHS_ROOT/conf/httpd.conf
   Web server port             - 80
   Map Applications            - MAP_ALL
   Plugin install location     - IHS_ROOT/Plugins
   Web server node type        - unmanaged
   Web server node name        - ti2022-l1.itso.ibm.com-node
   Web server host name        - ti2022-l1.itso.ibm.com
   Web server operating system - linux
   IHS Admin port              - 8008
   IHS Admin user ID           - httpadmin
   IHS Admin password          - admin
   IHS service name            - ""
```

```
Creating the unmanaged node ti2022-l1.itso.ibm.com-node .
Unmanged node ti2022-l1.itso.ibm.com-node is created.

Creating the web server definition for webserver1.
Parameters for administering IHS web server can also be updated using
wsadmin script or admin console.
Web server definition for webserver1 is created.

Start computing the plugin properties ID.
Plugin properties ID is computed.

Start updating the plugin install location.
Plugin install location is updated.

Start updating the plugin log file location.
Plugin log file location is updated.

Start updating the RemoteConfigFilename location.
Plugin remote config file location is updated.

Start updating the RemoteKeyRingFileName location.
Plugin remote keyring file location is updated.

Start saving the configuration.

Configuration save is complete.

Computed the list of installed applications.

Start saving the configuration.

Configuration save is complete.
```

3.5.10 Cluster configuration

To enable high availability capabilities and divide the workload among application servers, three clusters will be created:

- SCCDUI - User interface cluster
- SCCDMIF - Integration framework cluster
- SCCDCRON - Cron tasks cluster

Follow these steps to create the clusters and application servers:

1. Log into the Integrated Solutions Console and navigate to **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Select **New**.
3. Type `SCCDUI` for the Cluster name as shown in Figure 3-10.

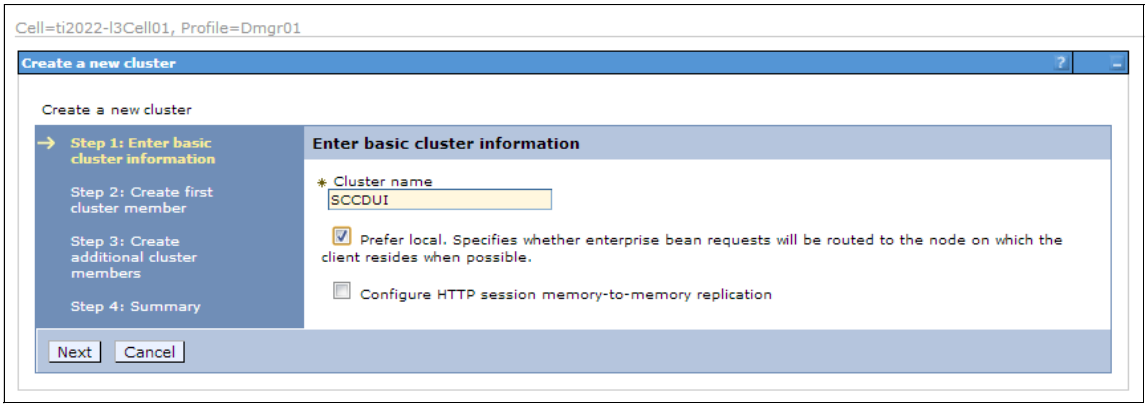

Figure 3-10 Cluster SCCDUI creation

4. Select **Next**.
5. Type `SCCDUI1` for Member name as shownin Figure 3-11 on page 58.

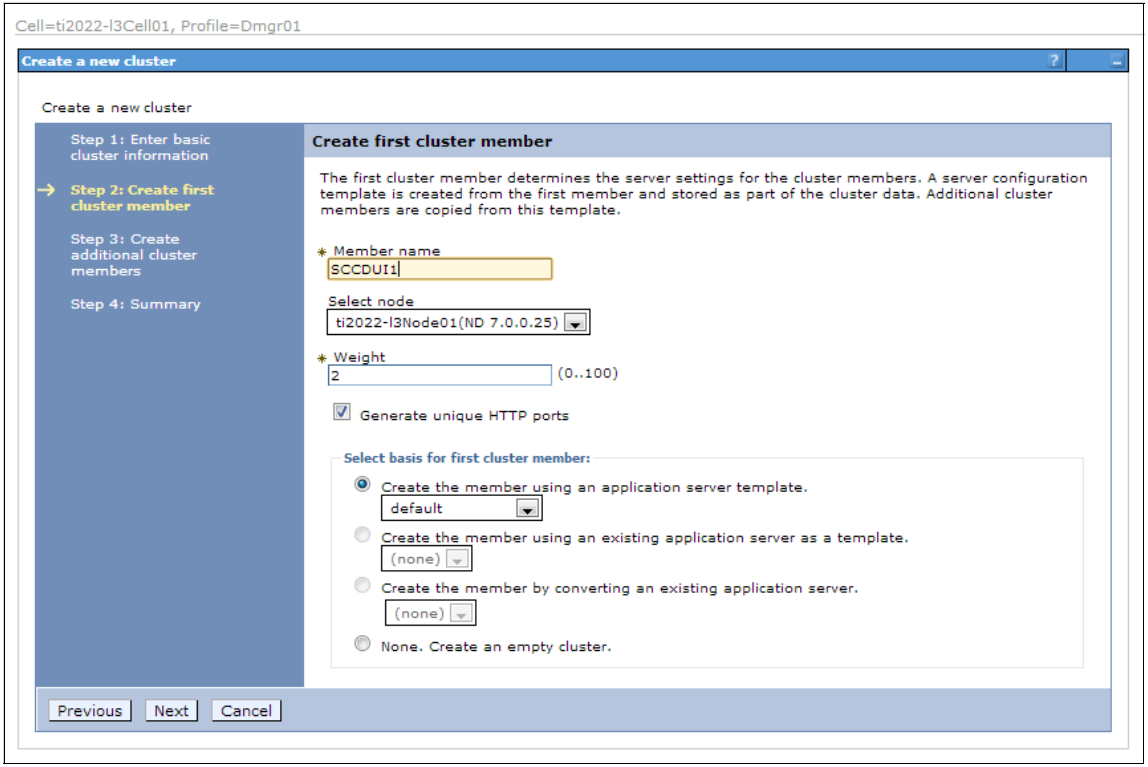

Figure 3-11 Cluster member SCCDUI1 creation

6. Select **Next**.
7. Type `SCCDUI2` for Member name as shown in Figure 3-12 on page 59.

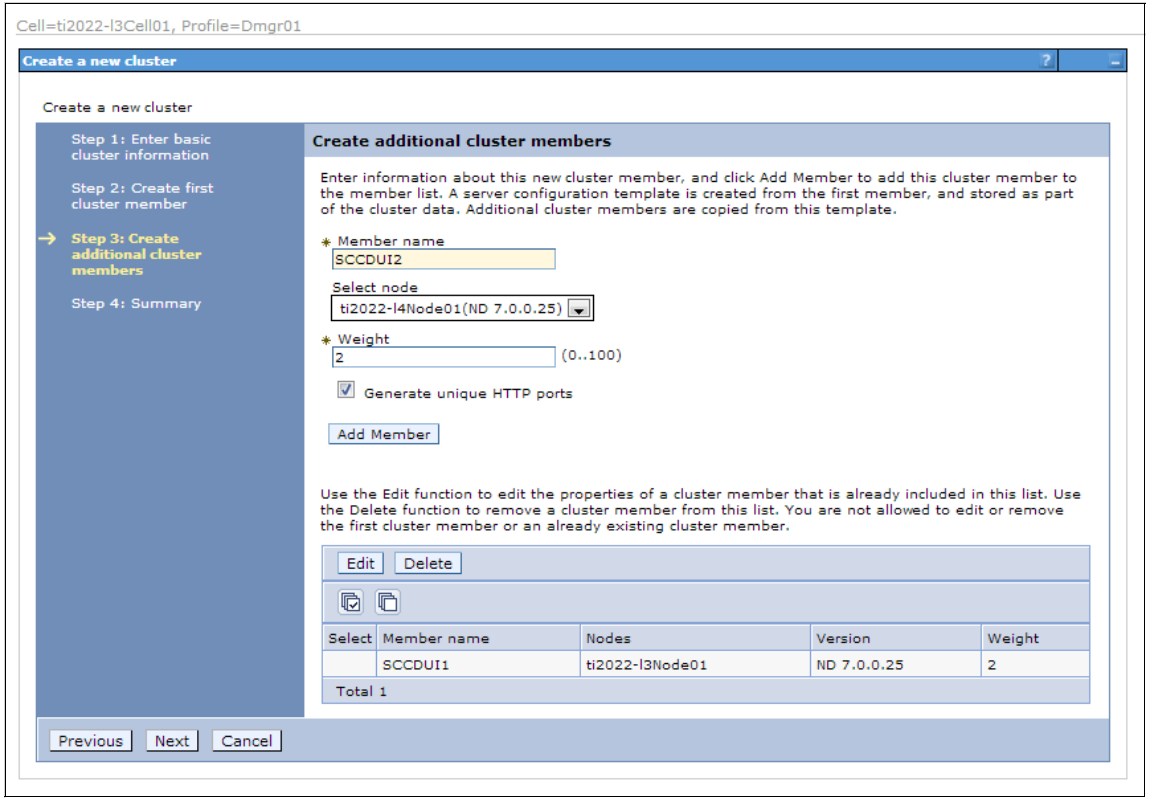

Figure 3-12 Cluster member SCCDUI2 creation

8. Select **Next**.
9. A summary table (Figure 3-13 on page 60) is displayed with cluster information. Review and select **Finish**.

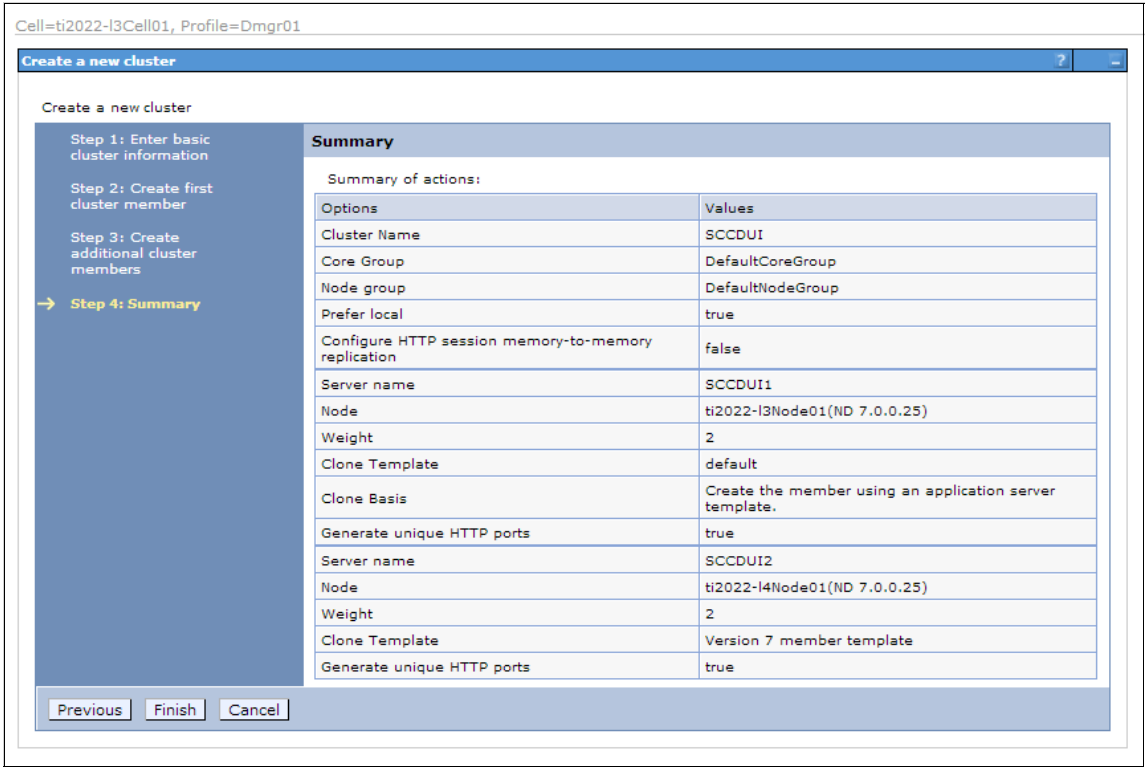

Figure 3-13 Cluster summary

10. Save and synchronize changes.
11. Repeat steps 2 on page 57 through 10 for clusters `SCCDMIF` and `SCCDCRON`.

After performing all the steps, the cluster panel should look as in Figure 3-14 on page 61, and the application server panel should look as in Figure 3-15 on page 61.

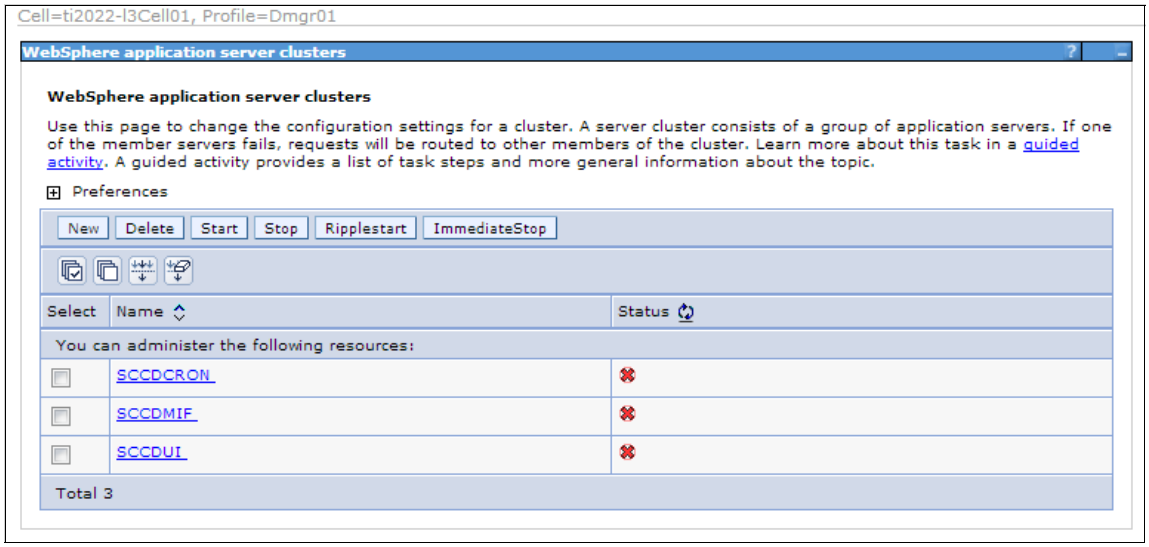

Figure 3-14 Clusters panel

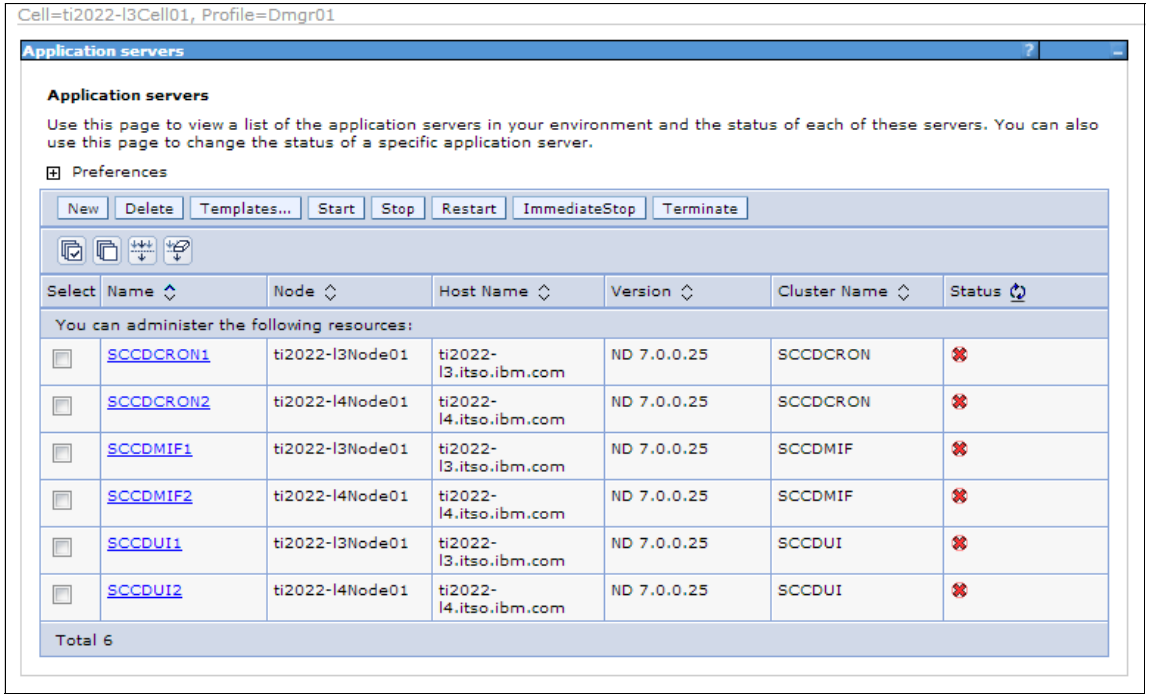

Figure 3-15 Application servers panel

Tip: When utilizing clusters with IBM SmartCloud Control Desk, a good practice is to identify each application server with its name. In order to do that, add the system property `-Dmxe.name=${WAS_SERVER_NAME}` to each application server. System properties can be set on **Servers** → **Server Types** → **WebSphere application servers** → **SERVER_NAME** → **Java and Process Management** → **Process definition** → **Java Virtual Machine** → **Generic JVM Arguments**.

After configuring the clusters, if integrations need to be enabled, refer to 3.8, “Integration framework” on page 107.

3.6 Database

The database is one of the most critical components of the IBM SmartCloud Control Desk application. It provides the option of using IBM DB2, Oracle or Microsoft SQL Server for the deployment. The middleware installer program provides the option of installing a new instance of DB2, or use a preexisting instance of the DB2 database. If you choose Oracle or Microsoft SQL Server, then you must install and configure them manually.

Bringing down the database will disrupt the IBM SmartCloud Control Desk function. It is advised to test your database high availability solution extensively in your environment. Various high availability database configurations are available. It is suggested that you review IBM SmartCloud Control Desk high availability documents to choose the optimum solution for your environment.

Information: For more information, refer to:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/topic/com.ibm.mbs.doc/gp_highavail/t_ctr_configure_database.html

In this book we cover the high availability topology using IBM DB2 High Availability and Disaster Recovery (HADR), DB2 shared disk, DB2 IBM pureScale®, Oracle Real Application Clusters, and Oracle Active Data Guard.

3.6.1 DB2 solutions

DB2 offers various options for designing a highly available solution for IBM SmartCloud Control Desk. The selection of the option will be based on your environment, budget, complexity, and time.

This section describes three options for implementing a high availability solution for IBM SmartCloud Control Desk: Cluster management with DB2 shared disk and DB2 HADR, both with Automatic Client Reroute (ACR); for high availability combined with the performance and scalability of an active cluster, DB2 pureScale can be implemented. Let us now take a closer look at the following details:

- DB2 setup
- HADR setup
- HADR requirements
- HADR considerations
- HADR setup
- DB2 High Availability Instance Configuration Utility
- HADR with a cluster manager
- HADR setup with db2haicu
- DB2 shared disk high availability setup
- DB2 shared disk HA requirements
- DB2 shared disk HA setup
- DB2 pureScale

DB2 setup

It is assumed that the DB2 is already installed and configured for IBM SmartCloud Control Desk application.

For this topology DB2 was installed on two separate servers. The first server performed the role of the primary IBM SmartCloud Control Desk database, the secondary server was the standby database. This data was kept synchronized using shared disk or HADR.

HADR setup

The HADR feature provides a highly available solution for database failure. HADR protects against data loss by replicating data changes from the primary database to the secondary database.

All changes that take place at the primary database are written to the DB2 logs. These logs are shipped to the secondary database server, where the log records are replayed to the local copy of the database. This ensures that the data on the primary and secondary database are in a synchronized state. The secondary server is always in the *rollforward* mode, in the state of near readiness, so the takeover to the standby server is fast.

HADR uses dedicated TCP/IP communication ports and a heartbeat to track the current state of the replication. If the standby database is up to date with the primary database, it is known as a HADR *peer* state.

If the primary database fails, then the HADR *takeover by force* operation converts the standby database to the new primary database. After the old primary database comes back online, you can return both servers to their original roles.

HADR requirements

The following requirements must be in place to set up HADR:

- The operating system version and patch level must be the same on the primary and standby database server. For a short duration during the rolling upgrade they may be different, but use caution.
- The DB2 version, level and bit size (32-bit or 64-bit) must be identical on the primary and standby database server.
- The primary and standby database must have the same name. This means that the two databases cannot be on the same server.
- A reliable TCP/IP interface must be available between the HADR servers.
- The database layout including the bufferpool sizes, tablespace name, size and type, and log space must be identical on the primary and secondary database servers.

HADR considerations

The following parameters should be considered for the HADR setup and adjusted according to the needs:

- AUTORESTART

  Consider setting this **db cfg** `parameter` to `OFF`, when the HADR database is configured with Automatic Client Reroute (ACR). Leave `AUTORESTART` to `ON` for non-HADR environments.

- LOGINDEXBUILD

  This **db cfg** `parameter` should be set to `ON` so that index creation, recreation, or reorganization on the tables are logged on the primary database and replayed on the secondary database.

- HADR_PEER_WINDOW

  `This` is used to ensure data consistency. If the value is set to greater than zero, the HADR database pair continues to behave as though they are in the peer state, for the configured time in case the connection is lost between the two databases.

  The advantage of configuring the peer window is a lower risk of transaction loss during multiple or cascading failures. The disadvantage of configuring the peer window is that transactions on the primary database will take longer or

time out when the primary database is in the peer window waiting for the connection to the standby database or for the peer window to expire.

This parameter must be adjusted according to the needs of your environment.

- HADR_TIMEOUT

  This **db cfg** `parameter` specifies the time in seconds that the DB2 HADR database waits for response from the other database before it considers the communication to have failed and closes the connection.

  This parameter must be adjusted according to the needs of your environment.

- SYNCMODE

  This **db cfg** parameter specifies one of the three synchronization modes. The synchronization modes indicate how log writing is managed between the primary and secondary servers. These modes apply only when the HADR is in the peer state. The valid values are:

  – SYNC (Synchronous)

    This mode provides the greatest protection against transaction loss, and using it results in the longest transaction response time among the three modes.

  – NEARSYNC (Near synchronous)

    While this mode has a shorter transaction response time than synchronous mode, it also provides slightly less protection against transaction loss.

  – ASYNC (Asynchronous)

    This mode has the highest chance of transaction loss if the primary system fails. It also has the shortest transaction response time among the three modes.

  – SUPERASYNC (Super Async)

    This mode ensures that the transaction can never be blocked or experience elongated response times due to network interruption or congestion, thereby allowing transactions to be processed more quickly.

  In our example we used the value of SYNC for the parameter.

Information: For more information about the HADR synchronization modes, refer to:

http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/c0011724.htm

HADR setup

This section describes how to set up HADR for the IBM SmartCloud Control Desk database. The setup is described using the command line interface.

The installation path and other variables are listed in Table 3-3.

Table 3-3 DB2 HADR variables

Variables	Description	Value
DB2_HOME	DB2 instance home	/home/db2inst1
DB2_INSTANCE	DB2 instance name	db2inst1
DB2_DBNAME	DB2 database name	maxdb75
db2hadrhost1	DB2 HADR primary hostname	ti2022-l5.itso.ibm.com
db2hadrhost2	DB2 HADR secondary hostname	ti2022-l6.itso.ibm.com
db2hadrlocalsvc	DB2 HADR local service	55001
db2hadrremotesvc	DB2 HADR remote service	55002

To set up HADR using the command line interface, complete the following steps:

1. Set the required database configuration parameters.

   If archive logging is not configured already, then update the LOGRETAIN and LOGARCHMETH1 parameters by running the following commands:

   ```
   db2 update db cfg for DB2_DBNAME using LOGRETAIN recovery
   db2 update db cfg for DB2_DBNAME using LOGARCHMETH1 LOGRETAIN
   ```

   Set the LOGINDEXBUILD parameter so that the index creation and reorganization operations are logged by running the following command:

   ```
   db2 update db cfg for DB2_DBNAME using LOGINDEXBUILD ON
   ```

2. Back up the database on the primary node by running the following command. The database backup should be an offline backup, which means no user connections are allowed on the database.

   ```
   db2 backup database DB2_DBNAME to BACKUP_PATH
   ```

3. Transfer the backup image to the secondary node.

4. Restore the database on the secondary server by running the following command. The standby database must be in the Rollforward pending mode.

   ```
   db2 restore database DB2_DBNAME from BACKUP_PATH taken at
   BACKUP_TIMESTAMP replace history file
   ```

Tip: Check the database `Rollforward pending` status issuing the **db2 get db cfg for DB2_DBNAME |grep "Rollforward pending"** command.

5. Update the database configuration parameters on the primary database server; see Example 3-20.

Example 3-20 DB2 HADR db cfg update commands for the primary

```
db2 “update db cfg for DB2_DBNAME using HADR_LOCAL_HOST db2hadrhost1”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_HOST db2hadrhost2”
db2 “update db cfg for DB2_DBNAME using HADR_LOCAL_SVC db2hadrlocalsvc”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_SVC
db2hadrremotesvc”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_INST DB2_INSTANCE”
db2 “update db cfg for DB2_DBNAME using HADR_TIMEOUT 120”
db2 “update db cfg for DB2_DBNAME using HADR_SYNCMODE SYNC”
db2 “update db cfg for DB2_DBNAME using HADR_PEER_WINDOW 120”
```

6. Run the **db2 "get db cfg for DB2_DBNAME" |grep HADR** command.

   Example 3-21 lists the db cfg parameter configuration for HADR on the primary database.

Example 3-21 HADR configuration for the primary database

```
db2inst1@ti2022-l5:~> db2 "get db cfg for maxdb75" |grep HADR
 HADR database role                                      = PRIMARY
 HADR local host name                  (HADR_LOCAL_HOST) = ti2022-l5
 HADR local service name                (HADR_LOCAL_SVC) = 55001
 HADR remote host name                (HADR_REMOTE_HOST) = ti2022-l6
 HADR remote service name              (HADR_REMOTE_SVC) = 55002
 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) = 120
```

7. Update the database configuration parameters on the secondary database server; Example 3-22.

Example 3-22 DB2 HADR db cfg update commands for the secondary

```
db2 “update db cfg for DB2_DBNAME using HADR_LOCAL_HOST db2hadrhost2”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_HOST db2hadrhost1”
db2 “update db cfg for DB2_DBNAME using HADR_LOCAL_SVC
db2hadrremotesvc”
```

```
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_SVC
db2hadrlocalsvc”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_INST DB2_INSTANCE”
db2 “update db cfg for DB2_DBNAME using HADR_TIMEOUT 120”
db2 “update db cfg for DB2_DBNAME using HADR_SYNCMODE SYNC”
db2 “update db cfg for DB2_DBNAME using HADR_PEER_WINDOW 120”
```

8. Run the **db2 "get db cfg for DB2_DBNAME" |grep HADR** command. Example 3-23 lists the db cfg parameter configuration for HADR on the secondary database.

Example 3-23 HADR configuration for the secondary database

```
db2inst1@ti2022-l6:~> db2 "get db cfg for maxdb75" |grep HADR
 HADR database role                                      = STANDBY
 HADR local host name                  (HADR_LOCAL_HOST) = ti2022-l6
 HADR local service name                (HADR_LOCAL_SVC) = 55002
 HADR remote host name                (HADR_REMOTE_HOST) = ti2022-l5
 HADR remote service name              (HADR_REMOTE_SVC) = 55001
 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) = 120
```

Note: Adjust the HADR_TIMEOUT, SYNCMODE and HADR_PEER_WINDOW values as per your requirements.

9. From DB2 version 9.7 Fixpack 5, the secondary database can be configured with read-only access, which allows the application and users to run queries and reports against the database. This step is optional.

   Example 3-24 shows the commands to set the read-only access on the secondary database.

Example 3-24 Commands to set the read-only access on the secondary database

```
db2set -i DB2_INSTANCE DB2_STANDBY_ISO=UR
db2set -i DB2_INSTANCE DB2_HADR_ROS=ON
```

10. Start HADR on the standby node by running the following commands.

```
db2 deactivate database DB2_DBNAME
db2 start hadr on database DB2_DBNAME as standby
```

11. Start HADR on the primary node by running the following command:

```
db2 start hadr on database DB2_DBNAME as primary
```

12. Verify HADR status by running the following command.

    ```
    db2pd -d DB2_DBNAME -hadr
    ```

    Example 3-25 displays the HADR status.

Example 3-25 HADR status output

```
db2inst1@ti2022-l5:/maximo/db2backup> db2pd -d maxdb75 -hadr

Database Partition 0 -- Database MAXDB75 -- Active -- Up 0 days 04:00:40 -- Date
10/31/2012 17:47:27

HADR Information:
Role    State                SyncMode   HeartBeatsMissed   LogGapRunAvg (bytes)
Primary Peer                 Sync     0                  0

ConnectStatus ConnectTime                           Timeout
Connected     Wed Oct 31 13:46:51 2012 (1351705611) 120

PeerWindowEnd                          PeerWindow
Wed Oct 31 17:49:25 2012 (1351720165) 120

LocalHost                                LocalService
ti2022-l5                                55001

RemoteHost                               RemoteService      RemoteInstance
ti2022-l6                                55002              db2inst1

PrimaryFile  PrimaryPg  PrimaryLSN
S0000058.LOG 536        0x00000000AD55850A

StandByFile  StandByPg  StandByLSN
S0000058.LOG 536        0x00000000AD55850A
```

DB2 High Availability Instance Configuration Utility

DB2 High Availability Instance Configuration Utility (db2haicu) can be used to configure and administer a highly available database for IBM SmartCloud Control Desk in a clustered environment. Information about the database instance, cluster environment, and cluster manager can be collected by querying the system. This utility is used to configure the cluster manager.

In this section, we describe how to configure HADR with IBM Tivoli System Automation for Multiplatforms (SA MP) to enable automating HADR takeover. Combining HADR with a cluster manager strengthens high availability for the IBM SmartCloud Control Desk database.

HADR does not monitor the status of the primary database for any outages. The standby database keeps waiting for logs to be transferred even though the primary is no longer active. The database administrator has to manually monitor the HADR status and run the appropriate takeover commands in case of a failure. In this situation a cluster manager such as System Automation for Multiplatforms can automate the detection and failover to a secondary server.

The cluster manager monitors the health of the network interface, hardware, and software processes, and detects and displays any failure. In case of failure the cluster manager can transfer the service and all the resources to the secondary server.

HADR with a cluster manager

When a primary database server outage occurs, the cluster manager monitors and detects the failure. In that situation the resources from the primary node are transferred to the secondary node. The secondary node now becomes the new primary node. When the failed server comes back online, it can be started as the standby, reversing the roles. Alternatively, the resources can be transferred back to the old primary and set the roles the same as prior to the outage.

HADR setup with db2haicu

The following prerequisites must be completed before configuring db2haicu with HADR:

- The System Automation for Multiplatforms package should be selected during DB2 installation.
- Update `/etc/hosts` files on primary and secondary nodes to reflect the hostname and complete domain hostname along with the IP address.

  Example 3-26 shows the example of the host file entry.

  Example 3-26 /etc/hosts file entry example

  ```
  9.12.5.168      ti2022-l5.itso.ibm.com ti2022-l5
  ```

- The primary and secondary server should be able to ping each other using the hostnames and IP addresses.
- The hostname resolution should be successful on the primary and secondary server.
- The service IP address must be available to configure the cluster manager.

Before using the `db2haicu` utility, the primary and secondary nodes must be prepared. Run the following command as `root` on both servers. This command needs to be run once per node.

```
preprpnode db2hadrhost1 db2hadrhost2
```

Once the nodes are prepared, log on to the secondary database server and issue the **db2haicu** command as DB2 instance administrator. The following section lists the setup tasks for **db2haicu**.

1. From the secondary database server, issue the **db2haicu** command; see Example 3-27.

Example 3-27 The db2haicu command

```
db2inst1@ti2022-l6:~> db2haicu
Welcome to the DB2 High Availability Instance Configuration Utility
(db2haicu).

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

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

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

db2haicu is collecting information on your current setup. This step
may take some time as db2haicu will need to activate all databases
for the instance to discover all paths ...
When you use db2haicu to configure your clustered environment, you
create cluster domains. For more information, see the topic
'Creating a cluster domain with db2haicu' in the DB2 Information
Center. db2haicu is searching the current machine for an existing
active cluster domain ...
db2haicu did not find a cluster domain on this machine. db2haicu
will now query the system for information about cluster nodes to
create a new cluster domain ...

db2haicu did not find a cluster domain on this machine. To continue
configuring your clustered environment for high availability, you
must create a cluster domain; otherwise, db2haicu will exit.

Create a domain and continue? [1]
1. Yes
```

```
2. No
```

Type 1 and press Enter to create a cluster domain.

2. Enter a unique name for the domain and the number of nodes contained in the domain. For this example we selected the domain name sccd_hadr_domain and two nodes. Next enter the hostnames for the two DB2 HADR server nodes. Example 3-28 lists the domain names and nodes.

Example 3-28 Cluster domain creation using db2haicu

```
Create a unique name for the new domain:
sccd_hadr_domain
Nodes must now be added to the new domain.
How many cluster nodes will the domain sccd_hadr_domain contain?
2
Enter the host name of a machine to add to the domain:
ti2022-l5
Enter the host name of a machine to add to the domain:
ti2022-l6
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 sccd_hadr_domain in the cluster ...
Creating domain sccd_hadr_domain in the cluster was successful.
You can now configure a quorum device for the domain. For more
information, see the topic "Quorum devices" in the DB2 Information
Center. If you do not configure a quorum device for the domain, then
a human operator will have to manually intervene if subsets of
machines in the cluster lose connectivity.
```

3. A quorum must be configured for the cluster domain. The supported quorum type for this solution is the network quorum, which must be a ping-able IP address (gateway router in the example) that is used to decide which node in the cluster will act as the active node during failure, and which node will be offline. Example 3-29 on page 72 shows the quorum creation. Enter 1 and press Enter to create the quorum.

Example 3-29 Network quorum creation

```
Configure a quorum device for the domain called sccd_hadr_domain?
[1]
```

```
1. Yes
2. No
1
The following is a list of supported quorum device types:
  1. Network Quorum
Enter the number corresponding to the quorum device type to be used:
[1]
1
Specify the network address of the quorum device:
9.12.4.1
Configuring quorum device for domain sccd_hadr_domain ...
Configuring quorum device for domain sccd_hadr_domain was
successful.
The cluster manager found 2 network interface cards on the machines
in the domain. You can use db2haicu to create networks for these
network interface cards. For more information, see the topic
'Creating networks with db2haicu' in the DB2 Information Center.
```

4. After the quorum configuration, define the public and private networks of your system to `db2haicu`. This step is important for the cluster to detect network failure. All network interfaces are automatically discovered by the `db2haicu` tool. Example 3-30 shows the definition of a public network.

Example 3-30 Public network definition

```
Create networks for these network interface cards? [1]
1. Yes
2. No
1
Enter the name of the network for the network interface card: eth0
on cluster node: ti2022-l5.itso.ibm.com
1. Create a new public network for this network interface card.
2. Create a new private network for this network interface card.
Enter selection:
1
Are you sure you want to add the network interface card eth0 on
cluster node ti2022-l5.itso.ibm.com to the network
db2_public_network_0? [1]
1. Yes
2. No
1
Adding network interface card eth0 on cluster node
ti2022-l5.itso.ibm.com to the network db2_public_network_0 ...
```

```
Adding network interface card eth0 on cluster node
ti2022-l5.itso.ibm.com to the network db2_public_network_0 was
successful.
Enter the name of the network for the network interface card: eth0
on cluster node: ti2022-l6.itso.ibm.com
1. db2_public_network_0
2. Create a new public network for this network interface card.
3. Create a new private network for this network interface card.
Enter selection:
1
Are you sure you want to add the network interface card eth0 on
cluster node ti2022-l6.itso.ibm.com to the network
db2_public_network_0? [1]
1. Yes
2. No
1
Adding network interface card eth0 on cluster node
ti2022-l6.itso.ibm.com to the network db2_public_network_0 ...
Adding network interface card eth0 on cluster node
ti2022-l6.itso.ibm.com to the network db2_public_network_0 was
successful.
Retrieving high availability configuration parameter for instance
db2inst1 ...
Retrieving high availability configuration parameter for instance
db2inst1 was successful.
Adding DB2 database partition 0 to the cluster ...
Adding DB2 database partition 0 to the cluster was successful.
```

5. After the network definition, **db2haicu** prompts for the cluster manager software being used for the current setup. Example 3-31 lists the selection of cluster manager software, in this case SA MP.

Example 3-31 Cluster manager software selection

```
The cluster manager name configuration parameter (high availability
configuration parameter) is not set. For more information, see the
topic
"cluster_mgr - Cluster manager name configuration parameter" in the
DB2
Information Center. Do you want to set the high availability
configuration parameter?
The following are valid settings for the high availability
configuration
parameter:
1.TSA
```

```
2.Vendor
Enter a value for the high availability configuration parameter: [1]
1
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.
```

6. After the DB2 secondary instance resource has been added to the cluster domain, confirm automation for the HADR database. Example 3-32 displays the validation of the HADR configuration.

Example 3-32 HADR configuration validation

```
Do you want to validate and automate HADR failover for the HADR
database MAXDB75? [1]
1. Yes
2. No
1
Adding HADR database MAXDB75 to the domain ...
The HADR database MAXDB75 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 MAXDB75. Run db2haicu on the primary for the
HADR database MAXDB75 to configure the database for automated
failover.
All cluster configurations have been completed successfully.
db2haicu exiting ...
```

7. After the secondary instance has been configured, the `db2haicu` configuration has to be run on the primary instance. Run the **`db2haicu`** command again on the primary node as DB2 instance administrator. The first step is to select cluster manager software for the setup. Example 3-33 shows the `db2haicu` setup on the primary node.

Example 3-33 db2haicu configuration on the primary node

```
db2inst1@ti2022-l5:~> db2haicu
Welcome to the DB2 High Availability Instance Configuration Utility
(db2haicu).

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

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

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

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

Retrieving high availability configuration parameter for instance
db2inst1 ...
The cluster manager name configuration parameter (high availability
configuration parameter) is not set. For more information, see the
topic "cluster_mgr - Cluster manager name configuration parameter"
in the DB2 Information Center. Do you want to set the high
availability configuration parameter?
The following are valid settings for the high availability
configuration parameter:
  1.TSA
  2.Vendor
Enter a value for the high availability configuration parameter: [1]
1
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.
```

8. **db2haicu** will then proceed to add the DB2 single partition resource for the primary database to the cluster. Next it will prompt you for confirmation of automating a HADR failover. Example 3-34 on page 77 shows the addition of the HADR primary database to the domain.

Example 3-34 Add HADR primary database to the domain

```
Do you want to validate and automate HADR failover for the HADR
database MAXDB75? [1]
1. Yes
2. No
1
Adding HADR database MAXDB75 to the domain ...
Adding HADR database MAXDB75 to the domain was successful.
```

9. Once the HADR database resource has been added to the cluster, **db2haicu** will prompt you to create a virtual IP address. Example 3-35 shows the addition of the service IP configuration for the cluster.

Example 3-35 Service IP configuration with db2haicu

```
Enter the virtual IP address:
9.12.4.135
Enter the subnet mask for the virtual IP address 9.12.4.135:
[255.255.255.0]
255.255.240.0
Select the network for the virtual IP 9.12.4.135:
1. db2_public_network_0
Enter selection:
1
Adding virtual IP address 9.12.4.135 to the domain ...
Adding virtual IP address 9.12.4.135 to the domain was successful.
All cluster configurations have been completed successfully.
db2haicu exiting ...
```

Ensure that the service IP address and subnet mask values are correct. All invalid inputs will be rejected. The configuration for the cluster has been completed. As `root`, issue the **lssam** command to see the resources created.

Figure 3-16 on page 78 lists the output of the **lssam** command. The resource group should be listed online along with the status for various other resources.

```
ti2022-l5:~ # lssam
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_MAXDB75-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs
                |- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:ti2022-l5
                '- Offline IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:ti2022-l6
        '- Online IBM.ServiceIP:db2ip_9_12_4_135-rs
                |- Online IBM.ServiceIP:db2ip_9_12_4_135-rs:ti2022-l5
                '- Offline IBM.ServiceIP:db2ip_9_12_4_135-rs:ti2022-l6
Online IBM.ResourceGroup:db2_db2inst1_ti2022-l5_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_ti2022-l5_0-rs
                '- Online IBM.Application:db2_db2inst1_ti2022-l5_0-rs:ti2022-l5
Online IBM.ResourceGroup:db2_db2inst1_ti2022-l6_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_ti2022-l6_0-rs
                '- Online IBM.Application:db2_db2inst1_ti2022-l6_0-rs:ti2022-l6
Online IBM.Equivalency:db2_db2inst1_db2inst1_MAXDB75-rg_group-equ
        |- Online IBM.PeerNode:ti2022-l5:ti2022-l5
        '- Online IBM.PeerNode:ti2022-l6:ti2022-l6
Online IBM.Equivalency:db2_db2inst1_ti2022-l5_0-rg_group-equ
        '- Online IBM.PeerNode:ti2022-l5:ti2022-l5
Online IBM.Equivalency:db2_db2inst1_ti2022-l6_0-rg_group-equ
        '- Online IBM.PeerNode:ti2022-l6:ti2022-l6
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:ti2022-l5
        '- Online IBM.NetworkInterface:eth0:ti2022-l6
```

Figure 3-16 lssam output

10. Set up the ACR feature on both DB2 database catalogs. The client reroute feature allows a DB2 client application to recover from a lost database connection in case of a network failure. In the high availability configuration the service IP address is used as alternate server for the DB2 database catalog. Example 3-36 displays the ACR setup. This command must be run on both DB2 nodes.

Example 3-36 Automatic client reroute configuration

```
db2inst1@ti2022-l5:~> db2 "update alternate server for database maxdb75 using
hostname 9.12.4.135 port 60000"
DB20000I  The UPDATE ALTERNATE SERVER FOR DATABASE command completed
successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
db2inst1@ti2022-l5:~> db2 "list db directory"
 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                       = MAXDB75
 Database name                        = MAXDB75
 Local database directory             = /maximo
 Database release level               = d.00
```

```
 Comment                             =
 Directory entry type                = Indirect
 Catalog database partition number   = 0
 Alternate server hostname           = 9.12.4.135
 Alternate server port number        = 60000

db2inst1@ti2022-l6:~> db2 "update alternate server for database maxdb75 using
hostname 9.12.4.135 port 60000"
DB20000I  The UPDATE ALTERNATE SERVER FOR DATABASE command completed
successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
db2inst1@ti2022-l6:~> db2 "list db directory"
 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                      = MAXDB75
 Database name                       = MAXDB75
 Local database directory            = /maximo
 Database release level              = d.00
 Comment                             =
 Directory entry type                = Indirect
 Catalog database partition number   = 0
 Alternate server hostname           = 9.12.4.135
 Alternate server port number        = 60000
```

DB2 shared disk high availability setup

In this section we describe the configuration of an IBM SmartCloud Control Desk database failover solution using DB2 shared disk storage. This configuration is based on the DB2 HA feature along with the `db2haicu` utility.

This configuration assumes that there is a shared disk configured and available for use between the active and passive DB2 servers. Figure 3-17 on page 80 displays the typical two-node setup using a shared disk.

In case the active node fails, all the DB2 resources on the shared disk are failed over to the passive node. The cluster manager will automatically mount the shared disk on the passive node and restart the? DB2 instance. At that time, the second node becomes the primary database server.

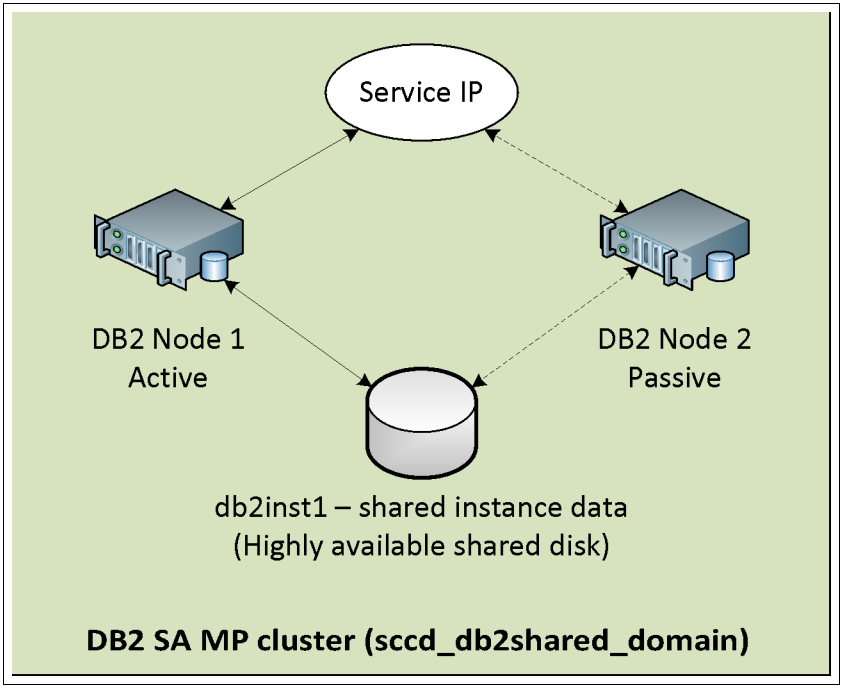

Figure 3-17 DB2 HA with shared disk

DB2 shared disk HA requirements

The following requirements must be in place for the DB2 shared disk HA setup:

- Two DB2 server nodes (active and passive) must be available.
- The shared disk storage should be available to both nodes. The disk is mounted to the primary node only.
- Mount all the required file systems only on the active node prior to the HA configuration. These mount points should also be mountable from the passive node. Ensure that the mount points are created using the `noauto` option to prevent them from being automatically mounted.

  There are some restrictions on what types of file systems can be made highly available. Only file systems that are local can be made highly available, for example:

  - jfs2
  - ext2
  - ext3
  - zfs

These file systems cannot be made highly available by default:

- Shared file systems such as NFS
- Clustered file systems such as GPFS, CFS
- Any file system mounted on the root (/) directory
- Any virtual file system such as `/proc`

Important: The mount points for the shared disks must be defined to the operating system being run on the active and the passive nodes (`/etc/fstab` file for Linux and `/etc/filesystems` for AIX). Consult your system administrator for details about other operating systems.

- The DB2 instance owner name, owner ID, group name, and group ID should be the same on all the nodes. In addition, it is required that the DB2 instance owner password be the same on both nodes.
- The DB2 installation should be performed on both nodes. The DB2 instance and database should be created on the shared disk mounted on the active node. Ensure that the `/etc/services` files on both the nodes match the DB2 entries.

DB2 shared disk HA setup

After a DB2 instance has been created on the primary node, the db2haicu utility will be used to automate HA failover.

The installation path and other variables are listed in Table 3-4.

Table 3-4 DB2 HADR variables

Variables	Description	Value
DB2_SD_HOME	DB2 shared disk instance home	`/sharedhome/db2inst1`
DB2_INSTANCE	DB2 instance name	`db2inst1`
DB2_DBNAME	DB2 database name	`maxdb75`
db2sdhost1	DB2 shared disk primary hostname	`ti2022-i7.itso.ibm.com`
db2sdhost2	DB2 shared disk secondary hostname	`ti2022-l8.itso.ibm.com`

The installation steps are as follows:

1. Before using **db2haicu**, the active and passive nodes must be prepared with the proper security environment. As `root`, run the **preprpnode** command once on both nodes:

   preprpnode db2sdhost1 db2sdhost2

2. From the active DB2 server, run the **db2haicu** command as the instance owner. Select option **1** to create a cluster domain. Create a unique name and add the number of nodes (2 in our example). Example 3-37 displays the cluster domain creation.

   Example 3-37 Cluster domain creation using db2haicu for shared disk

```
db2inst1@ti2022-i7:~> db2haicu
Welcome to the DB2 High Availability Instance Configuration Utility
(db2haicu).

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

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

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

db2haicu is collecting information on your current setup. This step
may take some time as db2haicu will need to activate all databases
for the instance to discover all paths ...
When you use db2haicu to configure your clustered environment, you
create cluster domains. For more information, see the topic
'Creating a cluster domain with db2haicu' in the DB2 Information
Center. db2haicu is searching the current machine for an existing
active cluster domain ...
db2haicu did not find a cluster domain on this machine. db2haicu
will now query the system for information about cluster nodes to
create a new cluster domain ...

db2haicu did not find a cluster domain on this machine. To continue
configuring your clustered environment for high availability, you
must create a cluster domain; otherwise, db2haicu will exit.
```

```
Create a domain and continue? [1]
1. Yes
2. No
1
Create a unique name for the new domain:
sccd_db2shared_domain
Nodes must now be added to the new domain.
How many cluster nodes will the domain sccd_db2shared_domain
contain?
2
```

3. Enter the hostnames of the active and passive nodes and confirm the domain creation. Example 3-38 lists the domain creation.

Example 3-38 Domain creation using db2haicu using shared disk

```
Enter the host name of a machine to add to the domain:
ti2022-i7
Enter the host name of a machine to add to the domain:
ti2022-l8
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 sccd_db2shared_domain in the cluster ...
Creating domain sccd_db2shared_domain in the cluster was successful.
```

4. A quorum must be configured for the cluster domain. The supported quorum type for this solution is network quorum. A network quorum must be a ping-able IP address (gateway router in the example) that is used to decide which node in the cluster will act as the active node during failure. Example 3-39 shows quorum creation. Type 1 and press Enter to configure the quorum device.

Example 3-39 Quorum device creation

```
Quorum devices" in the DB2 Information Center. If you do not
configure a quorum device for the domain, then a human operator will
have to manually intervene if subsets of machines in the cluster
lose connectivity.
```

```
Configure a quorum device for the domain called
sccd_db2shared_domain? [1]
1. Yes
2. No
1
The following is a list of supported quorum device types:
  1. Network Quorum
Enter the number corresponding to the quorum device type to be used:
[1]
1
Specify the network address of the quorum device:
9.12.4.1
Configuring quorum device for domain sccd_db2shared_domain ...
Configuring quorum device for domain sccd_db2shared_domain was
successful.
The cluster manager found 2 network interface cards on the machines
in the domain. You can use db2haicu to create networks for these
network interface cards. For more information, see the topic
'Creating networks with db2haicu' in the DB2 Information Center.
```

5. After the quorum configuration, define the public network of your system to **db2haicu**. This step is important for the cluster to detect network failure. Example 3-40 shows an example of a public network setup.

Example 3-40 Public network setup

```
Create networks for these network interface cards? [1]
1. Yes
2. No
1
Enter the name of the network for the network interface card: eth0
on cluster node: ti2022-i7.itso.ibm.com
1. Create a new public network for this network interface card.
2. Create a new private network for this network interface card.
Enter selection:
1
Are you sure you want to add the network interface card eth0 on
cluster node ti2022-i7.itso.ibm.com to the network
db2_public_network_0? [1]
1. Yes
2. No
1
Adding network interface card eth0 on cluster node
ti2022-i7.itso.ibm.com to the network db2_public_network_0 ...
```

```
Adding network interface card eth0 on cluster node
ti2022-i7.itso.ibm.com to the network db2_public_network_0 was
successful.
Enter the name of the network for the network interface card: eth0
on cluster node: ti2022-l8.itso.ibm.com
1. db2_public_network_0
2. Create a new public network for this network interface card.
3. Create a new private network for this network interface card.
Enter selection:
1
Are you sure you want to add the network interface card eth0 on
cluster node ti2022-l8.itso.ibm.com to the network
db2_public_network_0? [1]
1. Yes
2. No
1
Adding network interface card eth0 on cluster node
ti2022-l8.itso.ibm.com to the network db2_public_network_0 ...
Adding network interface card eth0 on cluster node
ti2022-l8.itso.ibm.com to the network db2_public_network_0 was
successful.
```

6. After the network definition, **db2haicu** prompts for the cluster manager software being used for the current setup. Example 3-41 displays the cluster manager selection. For our example we selected SA MP.

Example 3-41 TSA cluster manager selection

```
The cluster manager name configuration parameter (high availability
configuration parameter) is not set. For more information, see the
topic "cluster_mgr - Cluster manager name configuration parameter"
in the DB2 Information Center. Do you want to set the high
availability configuration parameter?
The following are valid settings for the high availability
configuration parameter:
  1.TSA
  2.Vendor
Enter a value for the high availability configuration parameter: [1]
1
Setting a high availability configuration parameter for instance
db2inst1 to TSA.
```

7. The next step is to configure the failover policy for the instance `db2inst1`. The failover policy determines the nodes on which the cluster manager will restart

the database manager if the database manager goes offline. In our example we selected option 3. Example 3-42 displays the failover policy selection.

Example 3-42 Failover policy selection

```
Now you need to configure the failover policy for the instance
db2inst1. The failover policy determines the machines on which the
cluster manager will restart the database manager if the database
manager is brought offline unexpectedly.

The following are the available failover policies:
  1. Local Restart -- during failover, the database manager will
restart in place on the local machine
  2. Round Robin -- during failover, the database manager will
restart on any machine in the cluster domain
  3. Active/Passive -- during failover, the database manager will
restart on a specific machine
  4. M+N -- during failover, the database partitions on one machine
will failover to any other machine in the cluster domain (used with
DPF instances)
  5. Custom -- during failover, the database manager will restart on
a machine from a user-specified list
Enter your selection:
3
```

8. Next **db2haicu** prompts you to designate any noncritical mount points. In our example we chose to designate two such points. You may choose to add any other noncritical mount points that you are sure that you never want to failover. The list should include any mount points listed in `/etc/fstab` on Linux or `/etc/filesystem` on AIX, *except* for the DB2 shared ones. Example 3-43 displays the noncritical mount point selection.

Example 3-43 Noncritical mount point selection

```
You can identify mount points that are noncritical for failover. For
more information, see the topic 'Identifying mount points that are
noncritical for failover' in the DB2 Information Center.  Are there
any mount points that you want to designate as noncritical? [2]
1. Yes
2. No
1
Enter the full path of the mount to be made non-critical:
/
Adding path / to the non-critical path list ...
Adding path / to the non-critical path list was successful.
Do you want to add more paths to the non-critical path list? [1]
```

```
1. Yes
2. No
1
Enter the full path of the mount to be made non-critical:
/dev
Adding path /dev to the non-critical path list ...
Adding path /dev to the non-critical path list was successful.
Do you want to add more paths to the non-critical path list? [1]
1. Yes
2. No
2
```

9. Next specify the hostnames for the active and passive nodes. The **db2haicu** utility will automatically add the DB2 nodes to the specified cluster manager. Example 3-44 displays the selection of the hostnames for active and passive nodes.

Example 3-44 Active and passive node name selection

```
Active/Passive failover policy was chosen. You need to specify the
host names of an active/passive pair.
Enter the host name for the active cluster node:
ti2022-i7
Enter the host name for the passive cluster node:
ti2022-l8
Adding DB2 database partition 0 to the cluster ...
Adding DB2 database partition 0 to the cluster was successful.
```

10. Once the database resource has been added to the cluster, **db2haicu** will prompt you to create a service IP address. Example 3-45 lists the setup of the service IP address.

Example 3-45 Service IP address setup for shared disk HA for DB2

```
Do you want to configure a virtual IP address for the DB2 partition:
0? [2]
1. Yes
2. No
1
Enter the virtual IP address:
9.12.4.167
Enter the subnet mask for the virtual IP address 9.12.4.167:
[255.255.255.0]
255.255.240.0
Select the network for the virtual IP 9.12.4.167:
1. db2_public_network_0
```

```
Enter selection:
1
Adding virtual IP address 9.12.4.167 to the domain ...
Adding virtual IP address 9.12.4.167 to the domain was successful.
All cluster configurations have been completed successfully.
db2haicu exiting ...
```

Ensure that the service IP address and subnet mask values are correct. All invalid inputs will be rejected. The configuration for the cluster has been completed.

11. Set up the ACR feature on both DB2 database catalogs. The client reroute feature allows a DB2 client application to recover from a lost database connection in case of a failure. In the HA configuration the service IP address is used as alternate server for the DB2 database catalog. This command should be run on the active database node. Example 3-46 displays the example of an automatic client reroute.

Example 3-46 Automatic client reroute

```
db2inst1@ti2022-i7:~> db2 "update alternate server for database
maxdb75 using hostname 9.12.4.167 port 60000"
DB20000I  The UPDATE ALTERNATE SERVER FOR DATABASE command completed
successfully.
DB21056W  Directory changes may not be effective until the directory
cache is
refreshed.
db2inst1@ti2022-i7:~> db2 "list db directory"

 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                       = MAXDB75
 Database name                        = MAXDB75
 Local database directory             = /sharedhome/maximo
 Database release level               = d.00
 Comment                              =
 Directory entry type                 = Indirect
 Catalog database partition number    = 0
 Alternate server hostname            = 9.12.4.167
 Alternate server port number         = 60000
```

12. Run **lssam** as `root` from the active server to see the status of the cluster and the new resource groups created during this process. Figure 3-18 displays the **lssam** output.

```
ti2022-i7:~ # lssam
Online IBM.ResourceGroup:db2_db2inst1_0-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_0-rs
                |- Online IBM.Application:db2_db2inst1_0-rs:ti2022-i7
                '- Offline IBM.Application:db2_db2inst1_0-rs:ti2022-l8
        |- Online IBM.Application:db2mnt-sharedhome-rs
                |- Online IBM.Application:db2mnt-sharedhome-rs:ti2022-i7
                '- Offline IBM.Application:db2mnt-sharedhome-rs:ti2022-l8
        '- Online IBM.ServiceIP:db2ip_9_12_4_167-rs
                |- Online IBM.ServiceIP:db2ip_9_12_4_167-rs:ti2022-i7
                '- Offline IBM.ServiceIP:db2ip_9_12_4_167-rs:ti2022-l8
Online IBM.Equivalency:db2_db2inst1_0-rg_group-equ
        |- Online IBM.PeerNode:ti2022-i7:ti2022-i7
        '- Online IBM.PeerNode:ti2022-l8:ti2022-l8
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:ti2022-i7
        '- Online IBM.NetworkInterface:eth0:ti2022-l8
```

Figure 3-18 lssam output after the configuration

DB2 pureScale

The DB2 pureScale feature incorporates several design features to deliver fault tolerance that not only can keep your instance available, but also minimizes the effect of component failures on the rest of the database system. DB2 pureScale works as an active cluster, which is accessible through a single IP and helps to achieve seamless failover. This solution gives your environment the performance benefits of a load-balanced database and the reliability of a highly available system.

You can implement DB2 pureScale to use these features and assist with your high availability configuration. However, you must complete some IBM SmartCloud Control Desk post-install configuration steps to use DB2 pureScale with your product.

For more information about DB2 pureScale, visit the DB2 pureScale Information Center at:

http://pic.dhe.ibm.com/infocenter/db2luw/v9r8/index.jsp

3.6.2 Oracle

If your environment uses Oracle as a database platform, high availability options are available that are compatible with IBM SmartCloud Control Desk. Oracle Real Application Clusters and Oracle Active Data Guard are two of Oracle's high

availability solutions. The information presented here is only a general overview, and more research into the Oracle HA capabilities should be done.

Oracle Real Application Clusters

Oracle Real Application Clusters (RAC) offers a highly available solution and an alternative to the Oracle Database installation. It facilitates database clustering by using a shared disk and shared cache approach, which improves scalability and performance.

You can create an Oracle RAC database across multiple nodes; however, you must perform specific configuration tasks to ensure that Oracle RAC operates smoothly with IBM SmartCloud Control Desk.

More information about Oracle RAC with IBM SmartCloud Control Desk can be found at:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/topic/com.ibm.mbs.doc/gp_highavail/c_oracle_rac.html

Oracle Active Data Guard

Oracle Active Data Guard is Oracle's standby database replication technology. Active Data Guard allows for two or more databases to synchronize through a log shipping mechanism and can act as a high availability solution for the IBM SmartCloud Control Desk database. The standby database can also be opened in a read-only mode which could allow for queries to the database for reporting functionality.

Active Data Guard provides several protection modes for log shipping and synchronization. It is important to research and determine which configuration works best for your organization. For more information about Active Data Guard, consult Oracle's website at:

http://www.oracle.com/ha

3.7 IBM SmartCloud Control Desk

In this section we describe the steps to enable high availability for the IBM SmartCloud Control Desk. These steps can also be used to configure other IBM Maximo products that are based on the Tivoli Process Automation Engine.

For this book we used the variables shown in Table 3-5. These values are not mandatory for all installations and might vary in other environments.

Table 3-5 Variables

Name	Description	Value
SCCD_HOME	IBM SmartCloud Control Desk installation path	`/opt/IBM/SMP`
SCCD_APP	IBM SmartCloud Control Desk application path	`SCCD_HOME/maximo/applications/maximo`
SCCD_DEPLOY	IBM SmartCloud Control Desk deployment path	`SCCD_HOME/maximo/deployment`
ATTACHMENTS_PATH	Attachments path	`/doclinks`
MIF_FILES_PATH	Maximo Integration Framework (MIF) files path	`/opt/sccd_files/mif`
OBJSRCHIDX_PATH	Object search index path	`/opt/sccd_files/objsearchindexes`

Let us now take a closer look at the following configuration steps:

- Using DB2 High Availability
- Split deployment files
- Ear file deployment on WebSphere Application Server
- Cron tasks configuration
- User Interface property setting
- Attachments configuration
- Object search indexes configuration

3.7.1 Using DB2 High Availability

If the IBM SmartCloud Control Desk application is already installed, then to take advantage of DB2 HA, the `maximo.properties` file needs to be modified. Update the hostname or IP address of the primary database server to the service IP address used for setup using **`db2haicu`**. Example 3-47 on page 92 shows the database connection information in the `maximo.properties` file.

Example 3-47 Database connection string in maximo.properties

```
mxe.db.url=jdbc:db2://9.12.4.135:60000/maxdb75
```

After modifying the IP address, rebuild and redeploy all the ear files. For a new IBM SmartCloud Control Desk installation, enter the service IP address as the database hostname and IP address instead of the hostname and IP address of the primary database server.

3.7.2 Split deployment files

To divide workload among WebSphere Application Server clusters and enable a highly available environment, the ear files must be split. The following steps describe how to configure different deployment files for each cluster.

1. Configure a base `SCCD_APP/properties/maximo.properties` file; Example 3-48.

   Example 3-48 Base maximo.properties example

   ```
   mxe.db.driver=com.ibm.db2.jcc.DB2Driver
   mxe.db.url=jdbc:db2://9.12.4.135:60000/maxdb75
   mxe.db.user=maximo
   mxe.db.password=admin
   mxe.db.schemaowner=maximo
   maximo.min.required.db.version=7100
   mxe.rmi.port=0
   mxe.registry.port=13400
   mxe.name=MXServer
   mxe.encrypted=false
   ```

2. Copy `maximo.properties` to `maximo_UI.properties`, `maximo_MIF.properties`, `maximo_CRON.properties`, and `maximo_ORIG.properties`; Example 3-49.

   Example 3-49 maximo.properties files

   ```
   ti2022-l10:SCCD_APP/properties # ls -l maximo*.properties
   -rw-r--r-- 1 root root 565 Nov  2 20:00 maximo.properties
   -rw-r--r-- 1 root root 538 Nov  4 13:32 maximo_CRON.properties
   -rw-r--r-- 1 root root 538 Nov  4 13:32 maximo_MIF.properties
   -rw-r--r-- 1 root root 318 Oct 25 18:01 maximo_ORIG.properties
   -rw-r--r-- 1 root root 565 Nov  4 13:31 maximo_UI.properties
   ti2022-l10:SCCD_APP/properties #
   ```

3. Add the following properties to `maximo_UI.properties`; Example 3-50.

Example 3-50 UI properties

```
mxe.crontask.donotrun=ALL
mxe.report.birt.disablequeuemanager=1
```

4. Add the following property to `maximo_MIF.properties`:

```
mxe.report.birt.disablequeuemanager=1
```

5. Add the following property to `maximo_CRON.properties`:

```
mxe.report.birt.disablequeuemanager=0
```

6. Go to the `SCCD_APP/mboweb/webmodule/WEB-INF` directory and copy `web.xml` to `web_UI_CRON.xml`, `web_MIF.xml`, and `web_ORIG.xml`; Example 3-51.

Example 3-51 web.xml files

```
ti2022-l10:SCCD_APP/mboweb/webmodule/WEB-INF # ls -l web*.xml
-rw-r--r-- 1 root root 4485 Oct 30 14:55 web.xml
-rw-r--r-- 1 root root 4479 Oct 26 16:53 web_MIF.xml
-rw-r--r-- 1 root root 4485 Oct 26 16:51 web_ORIG.xml
-rw-r--r-- 1 root root 4485 Oct 26 16:51 web_UI_CRON.xml
-rw-r--r-- 1 root root  852 Jan 25  2011 weblogic.xml
```

7. Update the `web_MIF.xml` file and comment out the BIRT servlet configurations, as shown in Example 3-52. The example only shows an excerpt of the complete XML file; the "..." placeholder represents content that has been left out intentionally.

Example 3-52 web_MIF.xml example

```
...
<!-- BIRT REPORT SERVLETS BEGIN
        <servlet>
                <description>Starts Report Engine</description>
                <display-name>Report Engine Startup Servlet</display-name>
                <servlet-name>ReportEngineStartupServlet</servlet-name>
                <servlet-class>
com.ibm.tivoli.maximo.report.birt.servlet.ReportEngineStartupServlet</servlet-class>
                <load-on-startup>1</load-on-startup>
        </servlet>
        <servlet>
                <description>Report Tool Servlet</description>
                <display-name>Report Tool Servlet</display-name>
                <servlet-name>ReportToolServlet</servlet-name>
                <servlet-class>
```

```
com.ibm.tivoli.maximo.report.birt.servlet.tool.ReportToolServlet</servlet-class>
        </servlet>
 BIRT REPORT SERVLETS END -->
...
<!-- BIRT REPORT SERVLET MAPPINGS BEGIN
        <servlet-mapping>
                <servlet-name>ReportToolServlet</servlet-name>
                <url-pattern>/reporttool/*</url-pattern>
        </servlet-mapping>
 BIRT REPORT SERVLET MAPPINGS END -->
...
<!--
    <security-constraint>
        <web-resource-collection>
            <web-resource-name>MAXIMO Report Tool</web-resource-name>
            <description>pages accessible by authorised users</description>
            <url-pattern>/reporttool/*</url-pattern>
            <http-method>GET</http-method>
            <http-method>POST</http-method>
        </web-resource-collection>
        <auth-constraint>
            <description>Roles that have access to MAXIMO Report Tool</description>
            <role-name>maximouser</role-name>
        </auth-constraint>
        <user-data-constraint>
            <description>data transmission gaurantee</description>
            <transport-guarantee>NONE</transport-guarantee>
        </user-data-constraint>
    </security-constraint>
-->
```

8. Go to `SCCD_APP/META-INF` and copy `deployment-application.xml` to `deployment-application_UI.xml`, `deployment-application_MIF.xml`, `deployment-application_CRON.xml`, and `deployment-application_CRON.xml`; Example 3-53.

Example 3-53 deployment-application.xml files

```
ti2022-l10:SCCD_APP/META-INF # ls -l deployment-application*.xml
-rw-r--r-- 1 root root 1398 Oct 24 18:20 deployment-application.xml
-rw-r--r-- 1 root root 1407 Oct 25 16:47
deployment-application_CRON.xml
-rw-r--r-- 1 root root 1405 Oct 25 16:47 deployment-application_MIF.xml
-rw-r--r-- 1 root root 1398 Oct 25 16:45
deployment-application_ORIG.xml
```

```
-rw-r--r-- 1 root root 1398 Oct 25 16:46 deployment-application_UI.xml
```

9. Change the `description` and `display-name` tags on `deployment-application.xml` as shown in Example 3-54. The example only shows an excerpt of the complete XML file; the "..." placeholder represents content that has been left out intentionally.

Example 3-54 deployment-application_UI.xml example

```
...
<description>SCCDUI</description>
<display-name>SCCDUI</display-name>
...
```

10. Repeat step 9 for `deployment-application_MIF.xml`, using `SCCDMIF` as value, and `deployment-application_CRON.xml`, using `SCCDCRON` as value.

11. Comment the `maximouiweb.war` module on `deployment-application_MIF.xml` and `deployment-application_CRON.xml` as shown in Example 3-55. The example only shows an excerpt of the complete XML file; the "..." placeholder represents content that has been left out intentionally.

Example 3-55 deployment-application_MIF.xml example

```
...
<!-- List of Web application modules -->
<!--
<module id="WebModule_1077124925258">
   <web>
      <web-uri>maximouiweb.war</web-uri>
      <context-root>/maximo</context-root>
   </web>
</module>
-->
...
```

12. Go to the `SCCD_DEPLOY` directory and copy the `buildmaximoear.xml` file to `buildmaximoear_UI.xml`, `buildmaximoear_MIF.xml`, `buildmaximoear_CRON.xml` and `buildmaximoear_ORIG.xml`.

Example 3-56 buildmaximoear.xml files

```
ti2022-l10:SCCD_DEPLOY # ls -l buildmaximoear*.xml
-rw-r--r-- 1 root root 14120 Oct 30 14:55 buildmaximoear-build.xml
-rw-r--r-- 1 root root 14226 Oct 30 14:55 buildmaximoear.xml
-rw-r--r-- 1 root root 14209 Oct 25 18:02 buildmaximoear_CRON.xml
-rw-r--r-- 1 root root 14208 Oct 25 18:02 buildmaximoear_MIF.xml
-rw-r--r-- 1 root root 14180 Oct 25 17:07 buildmaximoear_ORIG.xml
```

```
-rw-r--r-- 1 root root 14226 Oct 25 18:03 buildmaximoear_UI.xml
```

13. Update `buildmaximoear_UI.xml` to use the `deployment-application_UI.xml` file as shown in Example 3-57. The example only shows an excerpt of the complete XML file; the "..." placeholder represents content that has been left out intentionally.

Example 3-57 buildmaximoear_UI.xml example

```
...
<property name="maximo.appxmlfile"
value="${maximo.basedir}/META-INF/deployment-application_UI.xml"/>
...
```

14. Repeat step 13 for `buildmaximoear_MIF.xml`, using `deployment-application_MIF.xml` as value, and `buildmaximoear_CRON.xml`, using `deployment-application_CRON.xml` as value.

15. Insert an `exclude` ant task in the `propertiesBuild` target of the `buildmaximoear_UI.xml` file as shown in Example 3-58. The example only shows an excerpt of the complete XML file; the "..." placeholder represents content that has been left out intentionally.

Example 3-58 buildmaximoear_UI.xml example

```
...
<target name="propertiesBuild"
      depends="init"
      description="Builds the MAXIMO properties Archive File
(properties.jar) file">

   <echo>properties.jar
file=${maximo.deploydir.temp}/${maximo.propertiesjarfile}</echo>
   <copy todir="${maximo.deploydir.temp}/properties" >
      <fileset dir="${maximo.basedir}/properties">
         <include name="**/*.*"/>
         <exclude name="maximo_*.properties"/>
         <exclude name="version/*.*"/>
      </fileset>
   </copy>
...
```

16. Repeat step 15 for `buildmaximoear_MIF.xml` and `buildmaximoear_CRON.xml`.

17. Update an `exclude` ant task in the `earBuild` target of the `buildmaximoear_UI.xml` file as shown in Example 3-59 on page 97. The

example only shows an excerpt of the complete XML file; the "..." placeholder represents content that has been left out intentionally.

Example 3-59 buildmaximoear_UI.xml example

```
...
<target name="earBuild"
   ...
   <ear destfile="${maximo.deploydir}/${maximo.earfile}"
      ...
      <!-- WEB Application Modules files -->
      <fileset dir="${maximo.basedir}">
         ...
         <exclude name="META-INF/application.xml"/>
         <exclude name="META-INF/deployment-application_*.xml"/>
...
```

18. Repeat step 17 on page 96 for `buildmaximoear_MIF.xml` and `buildmaximoear_CRON.xml`.
19. Comment out targets `maximouiWarBuild` and `buildDojo` on the `buildmaximoear_MIF.xml` and `buildmaximoear_CRON.xml` files as shown in Example 3-60. The example only shows an excerpt of the complete XML file; the "..." placeholder represents content that has been left out intentionally.

Example 3-60 buildmaximoear_MIF.xml example

```
...
<!--
<target name="maximouiWarBuild"
      depends="init"
      description="Builds the MAXIMO UI Web Application Archive File
(maximoui.war) file">
   ...
</target>
-->
...
<!--
<target name="buildDojo">
   <echo>Building Dojo layer files</echo>
   <ant antfile="builddojo.xml"
dir="${basedir}/../applications/maximo" inheritAll="false"
target="all" />
</target>
-->
...
```

20. Remove the `maximouiWarBuild` and `buildDojo` targets from `earBuild` target dependencies in the `buildmaximoear_MIF.xml` and `buildmaximoear_CRON.xml` files as shown in Example 3-61. The example only shows an excerpt of the complete XML file; the "..." placeholder represents content that has been left out intentionally.

Example 3-61 buildmaximoear_MIF.xml example

```
...
<target name="earBuild"
      depends="init, propertiesBuild, businessObjectsBuild,
commonWebBuild, mboEJBClientBuild, mboWarBuild, mboEJBBuild,
meaWarBuild, maxrestWarBuild, mboJavaBuild"
      description="Builds the MAXIMO Enterprise Archive File
(maximo.ear) file">
...
```

21. Go to the `SCCD_DEPLOY` directory and copy the `buildmaximoear.sh` file to `buildmaximoear_UI.sh`, `buildmaximoear_MIF.sh`, `buildmaximoear_CRON.sh`, and `buildmaximoear_ORIG.sh`, Example 3-62.

Example 3-62 buildmaximoear.sh files

```
ti2022-l10:SCCD_DEPLOY # ls -l buildmaximoear*.sh
-rwxr--r-- 1 root root 1214 Jan 25  2011 buildmaximoear.sh
-rwxr--r-- 1 root root 1552 Nov  5 11:38 buildmaximoear_CRON.sh
-rwxr--r-- 1 root root 1545 Oct 26 16:57 buildmaximoear_MIF.sh
-rwxr--r-- 1 root root 1214 Oct 25 16:55 buildmaximoear_ORIG.sh
-rwxr--r-- 1 root root 1547 Nov  5 11:37 buildmaximoear_UI.sh
```

22. Update the `buildmaximoear_UI.sh` file to replace the custom configuration files before building the ear file as shown in Example 3-63. The example only shows an excerpt of the complete script file; the "..." placeholder represents content that has been left out intentionally.

Example 3-63 buildmaximoear_UI.sh example

```
...
export MAXIMO_HOME=./../applications/maximo
# ----------------------------------------------------------

# Changes IBM SmartCloud Control Desk default EAR build definition and properties
file
export BASE_DIR=./../applications/maximo
cp buildmaximoear_UI.xml buildmaximoear.xml
cp $BASE_DIR/properties/maximo_UI.properties $BASE_DIR/properties/maximo.properties
cp $BASE_DIR/mboweb/webmodule/WEB-INF/web_UI_CRON.xml \
```

```
      $BASE_DIR/mboweb/webmodule/WEB-INF/web.xml

export BUILD_DIR=./default
export EAR_FILENAME=sccdui.ear
export MAXIMO_PROPERTIES=maximo.properties
...
```

23. Update the `buildmaximoear_MIF.sh` file to replace the custom configuration files before building the ear file as shown in Example 3-64. The example only shows an excerpt of the complete script file; the "..." placeholder represents content that has been left out intentionally.

Example 3-64 buildmaximoear_MIF.sh example

```
...
export MAXIMO_HOME=./../applications/maximo
# ------------------------------------------------------------

# Changes IBM SmartCloud Control Desk default EAR build definition and properties
file
export BASE_DIR=./../applications/maximo
cp buildmaximoear_MIF.xml buildmaximoear.xml
cp $BASE_DIR/properties/maximo_MIF.properties $BASE_DIR/properties/maximo.properties
cp $BASE_DIR/mboweb/webmodule/WEB-INF/web_MIF.xml \
      $BASE_DIR/mboweb/webmodule/WEB-INF/web.xml

export BUILD_DIR=./default
export EAR_FILENAME=sccdmif.ear
export MAXIMO_PROPERTIES=maximo.properties
...
```

24. Update the `buildmaximoear_CRON.sh` file to replace the custom configuration files before building the `ear` file as shown in Example 3-65. The example only shows an excerpt of the complete script file; the "..." placeholder represents content that has been left out intentionally.

Example 3-65 buildmaximoear_CRON.sh example

```
...
export MAXIMO_HOME=./../applications/maximo
# ------------------------------------------------------------

# Changes IBM SmartCloud Control Desk default EAR build definition and properties
file
export BASE_DIR=./../applications/maximo
cp buildmaximoear_CRON.xml buildmaximoear.xml
```

```
cp $BASE_DIR/properties/maximo_CRON.properties $BASE_DIR/properties/maximo.properties
cp $BASE_DIR/mboweb/webmodule/WEB-INF/web_UI_CRON.xml \
      $BASE_DIR/mboweb/webmodule/WEB-INF/web.xml

export BUILD_DIR=./default
export EAR_FILENAME=sccdcron.ear
export MAXIMO_PROPERTIES=maximo.properties
...
```

After following these steps, generate new `ear` files running the three custom built scripts.

Example 3-66 Built scripts

```
buildmaximoear_UI.sh
buildmaximoear_MIF.sh
buildmaximoear_CRON.sh
```

3.7.3 Ear file deployment on WebSphere Application Server

After generating the new `.ear` files, it is time to deploy them on WebSphere Application Server. The following steps describe how to deploy the .ear file to its specific clusters.

1. Log in to Integrated Solutions Console and navigate to **Applications** → **Application Types** → **WebSphere enterprise applications**.
2. Select **Install**.
3. Select the `sccdui.ear` file and select **Next**.
4. Select **Next**.
5. Select **Next**.
6. Map the `SCCDUI` cluster and all web servers to all `.ear` modules as shown in Figure 3-19 on page 101.

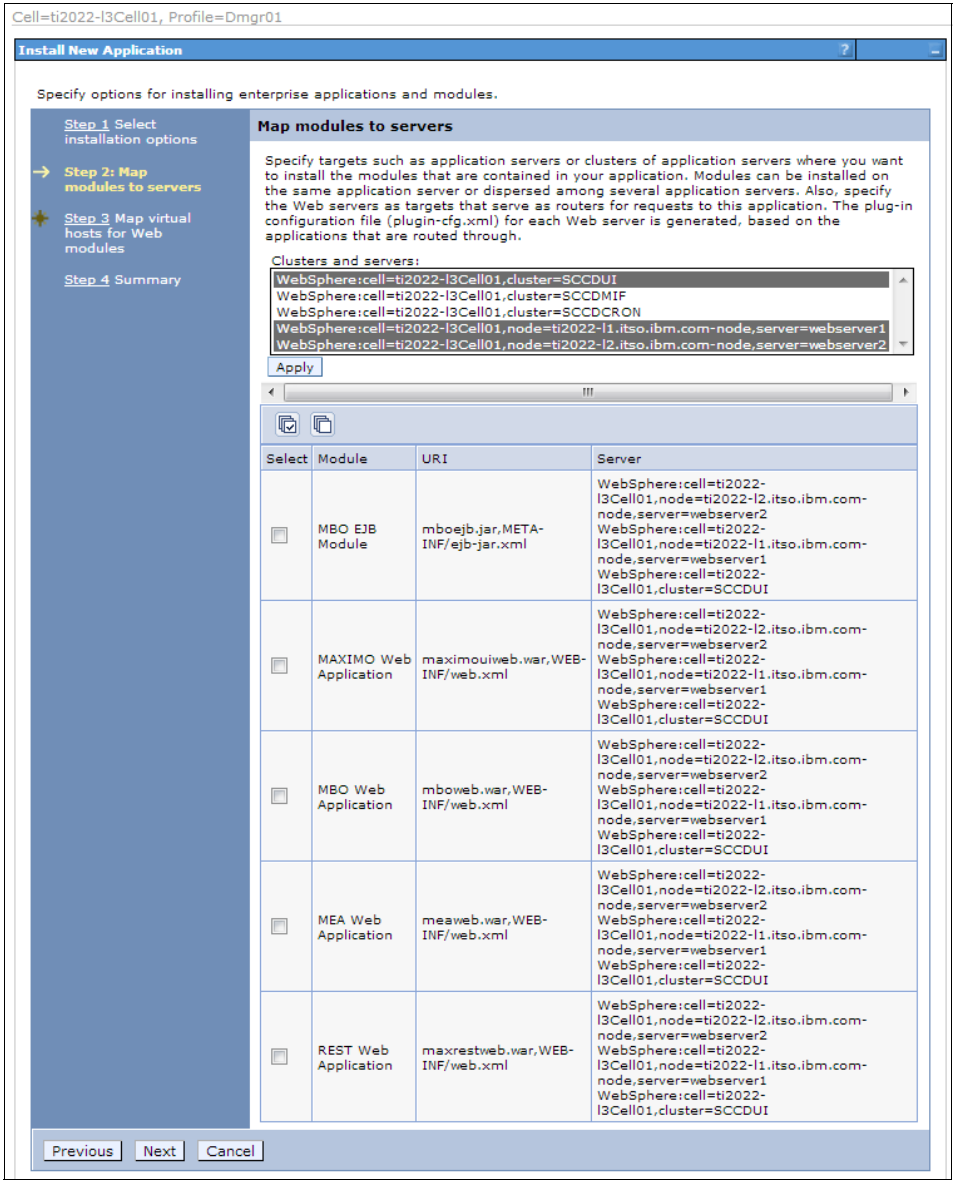

Figure 3-19 sccdui.ear deployment

7. Select **Next**.
8. If there is a custom virtual host configuration, map it to the appropriate modules and select **Next**.
9. Click **Finish**.

10. Save and synchronize changes.
11. Repeat steps 2 on page 100 through 10 on page 102 for `sccdmif.ear`, using cluster `SCCDMIF`, and `sccdcron.ear`, using the `SCCDCRON` cluster.

> **More information:** For more information about deploying IBM SmartCloud Control Desk `.ear` files, refer to:
>
> http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mam.inswas.doc%2Finstall%2Fc_ccmdb_deployccmdbearfiles.html

After installing the `.ear` files, start the clusters.

3.7.4 Cron tasks configuration

After splitting the `.ear` files, make sure that the JMS cron tasks are running on the MIF cluster and other cron tasks are running on the CRON cluster. The following steps describe how to divide this workload.

1. Connect to the IBM SmartCloud Control Desk database and run the query shown in Example 3-67 to get all JMS cron tasks configured in the environment.

Example 3-67 JMS cron tasks SQL

```
select
   crontaskname || '.' || instancename
from
   crontaskinstance
where
   crontaskname = 'JMSQSEQCONSUMER'
```

2. Group all results in a comma-separated list as shown:

   `JMSQSEQCONSUMER.SEQQIN,JMSQSEQCONSUMER.SEQQOUT`

3. Log in to IBM SmartCloud Control Service Desk and navigate to **System Configuration** → **Platform Configuration** → **System Properties**.
4. On the `Instance Properties` section, select **New Row**.
5. Type the following values:
   a. Property Name

      `mxe.crontask.donotrun`

b. Server

   SCCDCRON1

c. Value

   The list built based on the SQL shown in Example 3-67

> **Tip:** Depending on how many cron tasks are defined, the size of the attribute MAXPROPVALUE.PROPVALUE may need to be increased.

6. Select **Save**.
7. Repeat steps 4 on page 102 through 6 for server SCCDCRON.
8. Connect to the IBM SmartCloud Control Desk database and run the query shown in Example 3-68 to get all non-JMS cron tasks configured in the environment.

Example 3-68 Non-JMS cron tasks SQL

```
select
   crontaskname || '.' || instancename
from
   crontaskinstance
where
   crontaskname <> 'JMSQSEQCONSUMER'
```

9. Group all results in a comma-separated list as shown in Example 3-69.

Example 3-69 Non JMS cron task list

```
AssetTopoCacheCron.AssetTopoCacheCron01,AsyncImmediateJobCron.AsyncI
mmediate,AsyncJobCleanupCron.AsyncJobCleanup, ...
```

10. Log in to IBM SmartCloud Control Service Desk and navigate to **System Configuration** → **Platform Configuration** → **System Properties**.
11. In the Instance Properties section, select **New Row**.
12. Type the following values:

a. Property Name

   mxe.crontask.donotrun

b. Server

   SCCDMIF1

c. Value

   The list built based on the SQL shown in Example 3-68.

Tip: Depending on how many cron tasks are defined, the size of the attribute `MAXPROPVALUE.PROPVALUE` may need to be increased.

13. Select **Save**.
14. Repeat steps 8 through 13 for server `SCCDMIF2`.

3.7.5 User Interface property setting

To enable proper failover user session handling, a property must be enabled on IBM SmartCloud Control Desk. The following steps describe how to enable this property:

1. Log in to IBM SmartCloud Control Desk and navigate to **System Configuration → Platform Configuration → System Properties**.
2. Search for **`mxe.webclient.lostconnectionwarningonly`** and set its value to .
3. Select the property check box and perform a **Live Refresh**.

3.7.6 Attachments configuration

To allow attachments to be uploaded and downloaded from any node in a clustered topology, the path must be shared. All the configurations described in this section assume that the `ATTACHMENTS_PATH` is already shared on the same path across all nodes.

1. Log in to the IHS server and go to the `IHS_ROOT/conf` directory.
2. Open the `httpd.conf` file and search for the `Alias` section. Add the alias shown in Example 3-70. The example only shows an excerpt of the complete configuration file; the "..." placeholder represents content that has been left out intentionally.

Example 3-70 httpd.conf alias setup example

```
...
# Aliases: Add here as many aliases as you need (with no limit)...
...
Alias /icons/ "IHS_ROOT/icons/"
<Directory "IHS_ROOT/icons">
   ...
</Directory>
Alias ATTACHMENTS_PATH/ "ATTACHMENTS_PATH/"
<Directory "ATTACHMENTS_PATH/">
    Options Indexes MultiViews
```

```
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
...
```

3. Restart IHS.
4. Repeat steps 1 on page 104 through 3 for all IHS servers.
5. Log in to IBM SmartCloud Control Desk and navigate to **System Configuration** → **Platform Configuration** → **System Properties**.
6. Search for **mxe.doclink.doctypes.defpath** and set its value to `ATTACHMENTS_PATH.`
7. Search for **mxe.doclink.doctypes.topLevelPaths** and set its value to `ATTACHMENTS_PATH.`
8. Restart the IBM SmartCloud Control Desk application servers.
9. Navigate to any application with attachments, for example **Service Desk** → **Incidents**.
10. Select **Select Action** → **Attachment Library/Folders** → **Manage Folders**.
11. Update all existing `Default File Path` values to `ATTACHMENTS_PATH`; see Figure 3-20 on page 106.

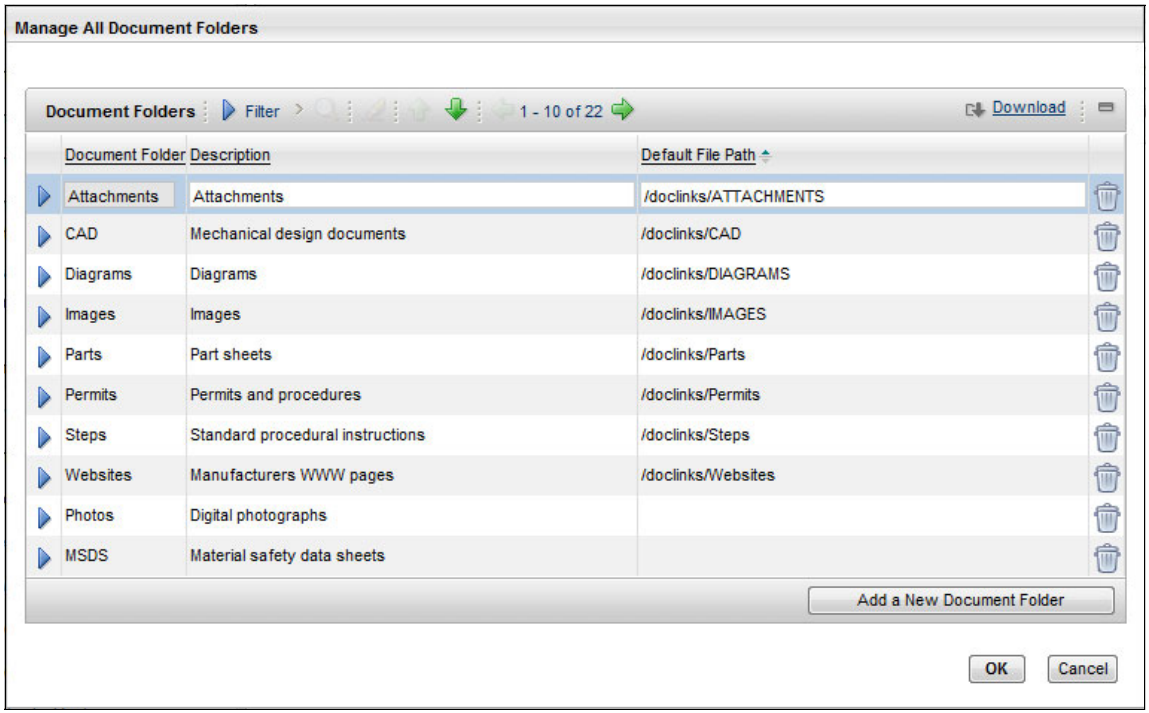

Figure 3-20 Attachments path configuration

3.7.7 Object search indexes configuration

The Global Search application is used to search through Service Request, Incident, Problem and Solution records for specific texts at the same time. This feature utilizes an index to find the proper search results in a timely fashion. All the configurations described in this section assume that the OBJSRCHIDX_PATH is already shared on the same path across all nodes.

1. Log in to IBM SmartCloud Control Desk and navigate to **System Configuration → Platform Configuration → System Properties**.
2. Search for **LUCENEOBJINDEX** and set its value to `OBJSRCHIDX_PATH`.
3. Select the property check box and perform a **Live Refresh**.

More information: For more information about the the Global Search application, refer to:

http://pic.dhe.ibm.com/infocenter/tivihelp/v50r1/index.jsp?topic=%2Fcom.ibm.tusc.doc%2Fglobal_search%2Ft_gsearch_intro.html

3.8 Integration framework

Integration is an important part of IBM SmartCloud Control Desk, because it allows inter-operation with other systems and data loads. The Maximo Integration Framework (MIF) makes use of Java Message Service (JMS) resources and file system-based error management to function properly. This section describes the configuration necessary to make this feature highly available.

For this book we used the variables in Table 3-5. These values are not mandatory for all installations and might vary in other environments.

Table 3-6 Variables

Name	Description	Value
MIF_FILES_PATH	Maximo Integration Framework (MIF) files path	`/opt/sccd_files/mif`
MQ_QM_PATH	WebSphere MQ queue manager installation path	`/opt/mq_files`
MQ_QM_DATA	WebSphere MQ queue manager data path	`MQ_QM_PATH/data`
MQ_QM_LOGS	WebSphere MQ queue manager logs path	`MQ_QM_PATH/logs`
mqhost1	WebSphere MQ primary server hostname	`ti2022-l11.itso.ibm.com`
mqhost2	WebSphere MQ secondary server hostname	`ti2022-l9.itso.ibm.com`
WAS_DMGR_PATH	WebSphere Application Server deployment manager profile installation path	`/opt/was_dmgr_files/profiles/Dmgr01`

3.8.1 Maximo Integration Framework configuration

The Maximo Integration Framework (MIF) web services are based on Web Service Definition Language (WSDL) and XML Schema Definition (XSD). These definitions must be visible to all MIF servers in the topology. All the configurations described in this section assume that the `MIF_FILES_PATH` is already shared on the same path across all nodes.

1. Log in to IBM SmartCloud Control Desk and navigate to **System Configuration** → **Platform Configuration** → **System Properties**.

2. Search for **mxe.int.globaldir** and set its value to MIF_FILES_PATH.

Select the property check box and perform a **Live Refresh**.

3.8.2 Java Message Service resources configuration

When defining JMS resources, there are two options for hosting such resources:

- WebSphere Application Server Service Integration Bus (SIB), described in “Service integration bus configuration” on page 108.
- WebSphere MQ, described in “WebSphere MQ configuration” on page 123.

For local high availability topology, the SIB solution is faster to set up and requires less resources, since its runtime is embedded in WebSphere Application Server. The WebSphere MQ solution is more robust for active-passive and hybrid-active topologies and requires less configuration efforts when changing the environment.

Service integration bus configuration

The integration bus must be a highly available component to avoid a single point of failure. The following steps will cover intjmsbus creation across the SCCDMIF cluster.

1. Create the WAS_DMGR_PATH/jdbc directory.
2. Transfer DB2 JDBC drivers located under DB2_INSTALL_PATH/java to WAS_DMGR_PATH/jdbc. After transferring drivers, it should list the files shown in Example 3-71.

Example 3-71 DB2 JDBC drivers listing

```
ti2022-l3:WAS_DMGR_PATH/jdbc # ls -l
total 10820
-r--r--r-- 1 root root 2968087 Oct 31 17:32 Common.jar
-r--r--r-- 1 root root  878575 Oct 31 17:32 db2java.zip
-r--r--r-- 1 root root 3502759 Oct 31 17:32 db2jcc.jar
-r--r--r-- 1 root root 3700005 Oct 31 17:32 db2jcc4.jar
-r--r--r-- 1 root root    1015 Oct 31 17:32 db2jcc_license_cu.jar
```

Tip: The DB2_INSTALL_PATH for DB2 V9.7 default value for Linux is /opt/ibm/db2/V9.7.

3. Log into the Integrated Solutions Console and navigate to **Environment** → **WebSphere variables**.
4. Select **Cell scope** and then **New**.

5. Type `WAS_DMGR_PATH` for Name and the deployment manager path for Value as shown in Figure 3-21.

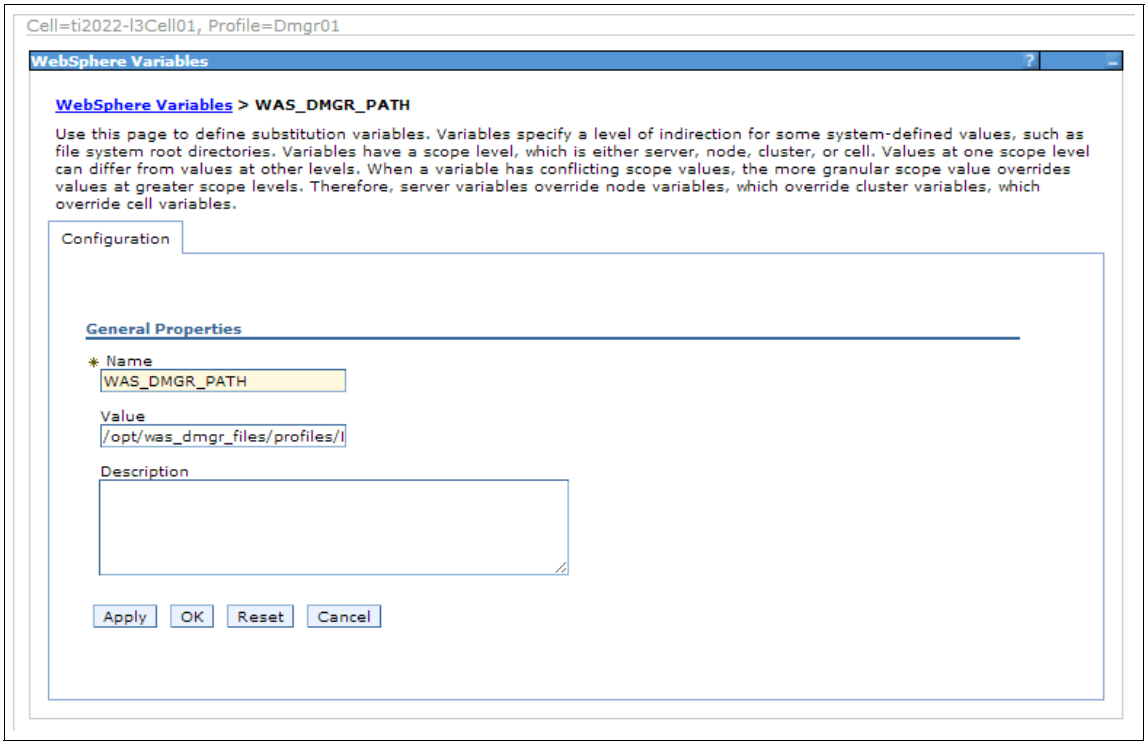

Figure 3-21 WAS_DMGR_PATH variable setup

6. Select **OK**.
7. Save and synchronize changes.
8. Navigate to **Resources** → **JDBC** → **JDBC providers**.
9. Select **Cell scope** and select **New**.
10. Select `DB2` as Database Type, `DB2 Using IBM JCC Driver` as Provider Type and `Connection pool data source` as Implementation type, as shown in Figure 3-22 on page 110.

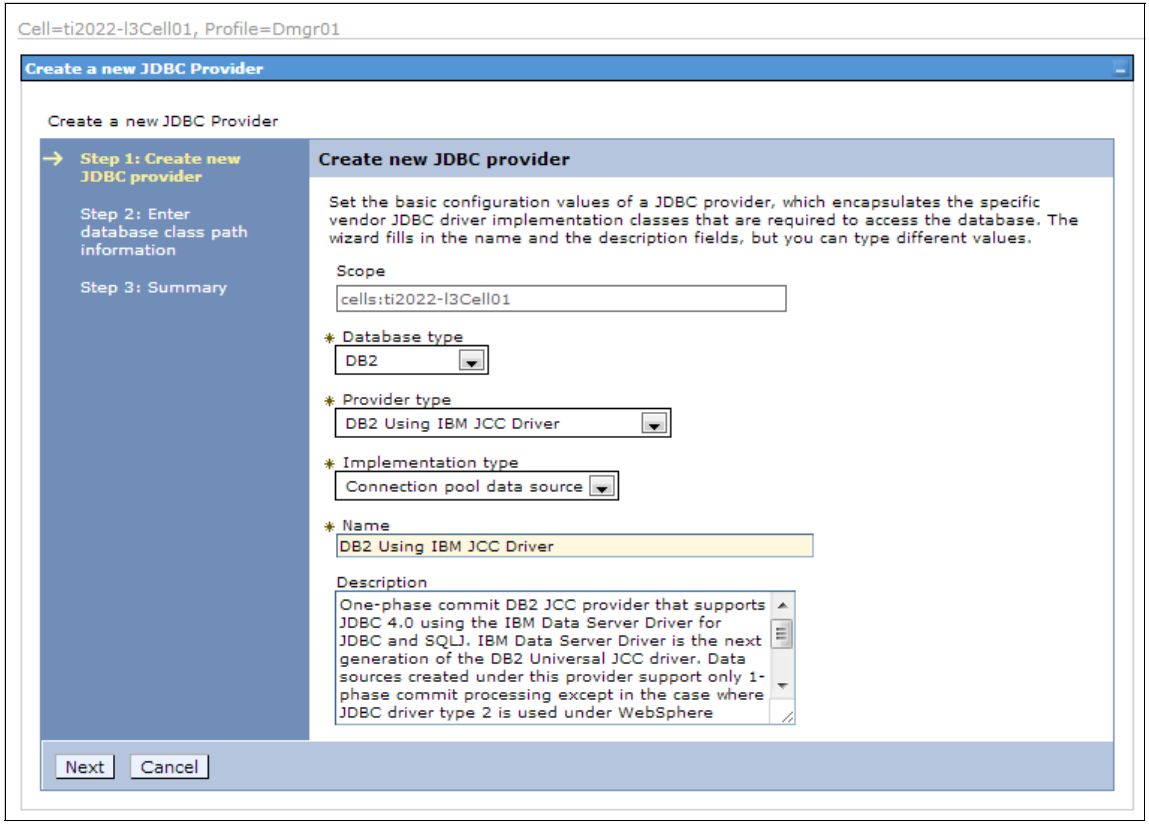

Figure 3-22 JDBC provider creation

11. Select **Next**.
12. Type `${WAS_DMGR_PATH}/jdbc` as both paths as shown in Figure 3-23 on page 111.

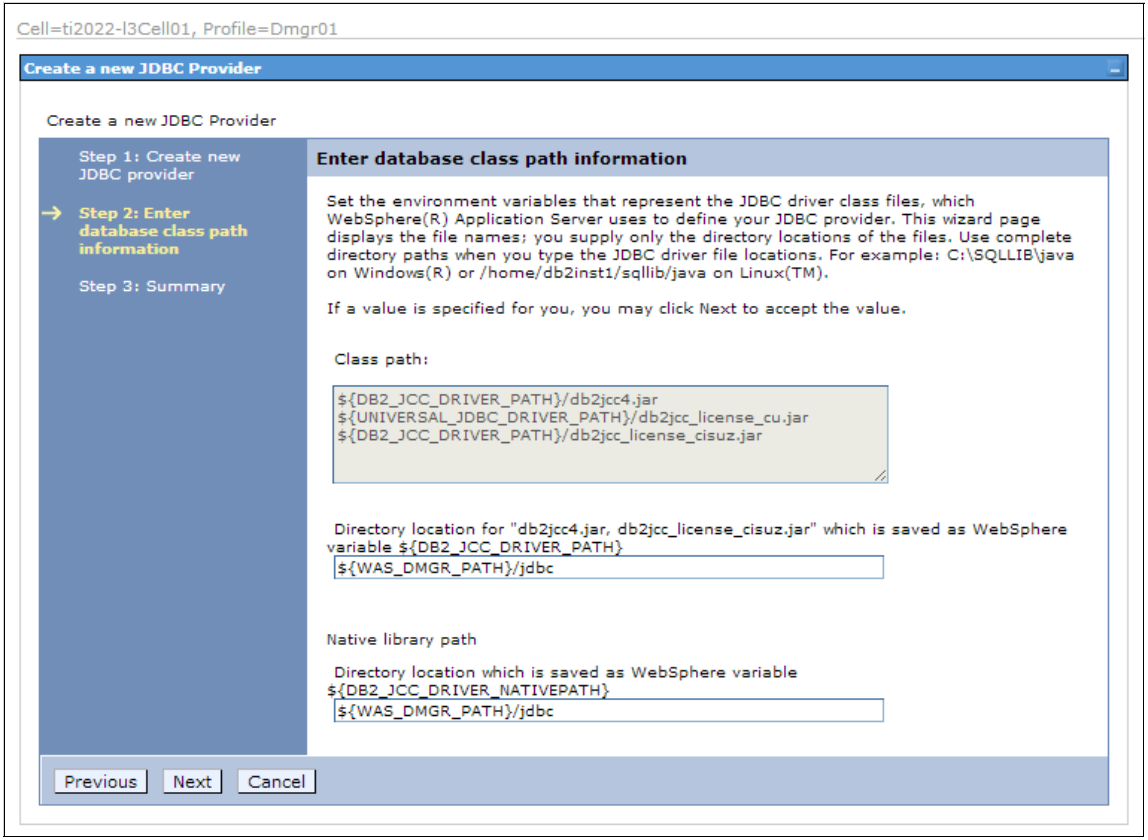

Figure 3-23 JDBC provider driver path setup

13. Select **Next**.
14. A summary table (Figure 3-24 on page 112) will be displayed with the provider information. Review and select **Finish**.

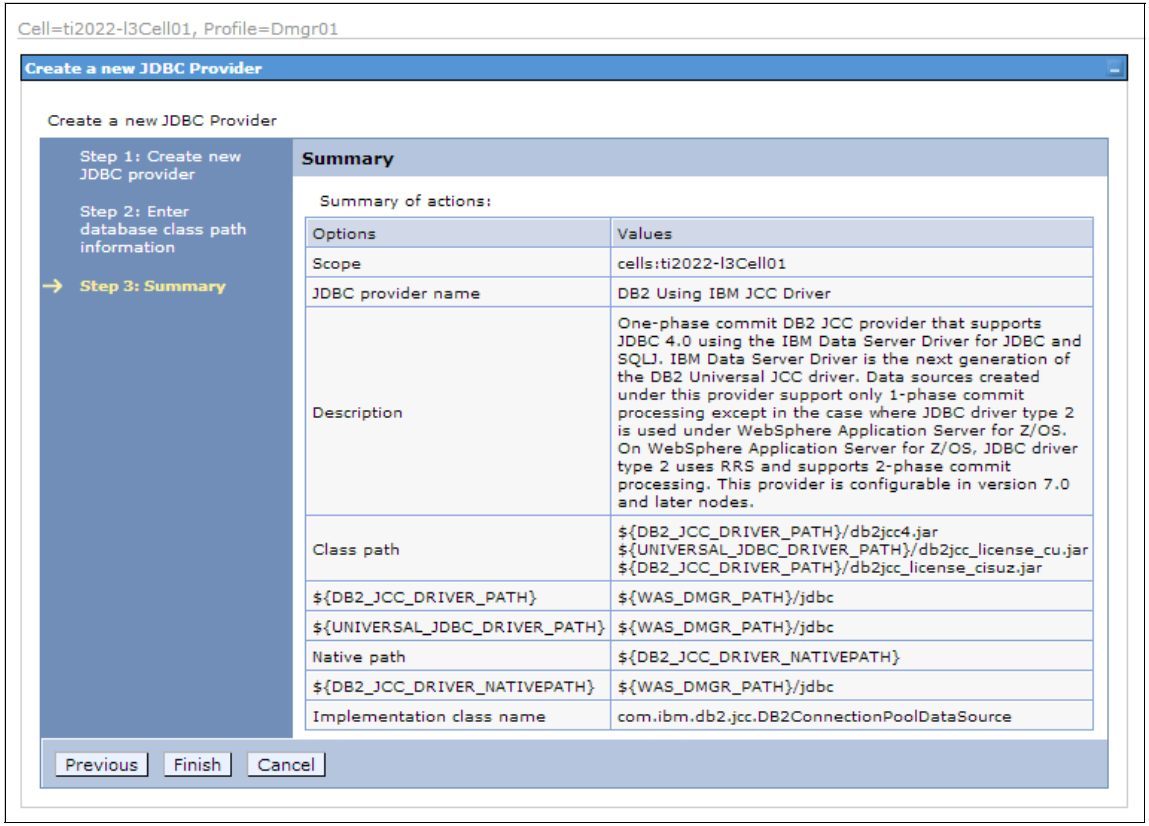

Figure 3-24 JDBC provider creation summary

15. Save and synchronize changes.
16. Navigate to **Security** → **Global Security** → **Java Authentication and Authorization Service** → **J2C authentication data**.
17. Select **New**.
18. Type `maximo` for alias, user id and password that will be used to store Java Message Service (JMS) integration data and select **OK**.
19. Save and synchronize changes.

Important: The user ID configured must have create schema and select, insert, delete, and update privileges.

20. Navigate to **Resources** → **JDBC** → **Data sources**.
21. Select **Cell scope** and then **New**.

22. Type `MAXDB75` as Data source name, `jdbc/MAXDB75` as JNDI name, and select **Next**.

23. Select `DB2 Universal JDBC Driver Provider` as an existing JDBC provider and select **Next**.

24. Type `MAXDB75` as Database name, database `IP/hostname address` as Server name, database `port` as Port number, uncheck the “Use this data source in container managed persistence (CMP)” option (Figure 3-25), and select **Next**.

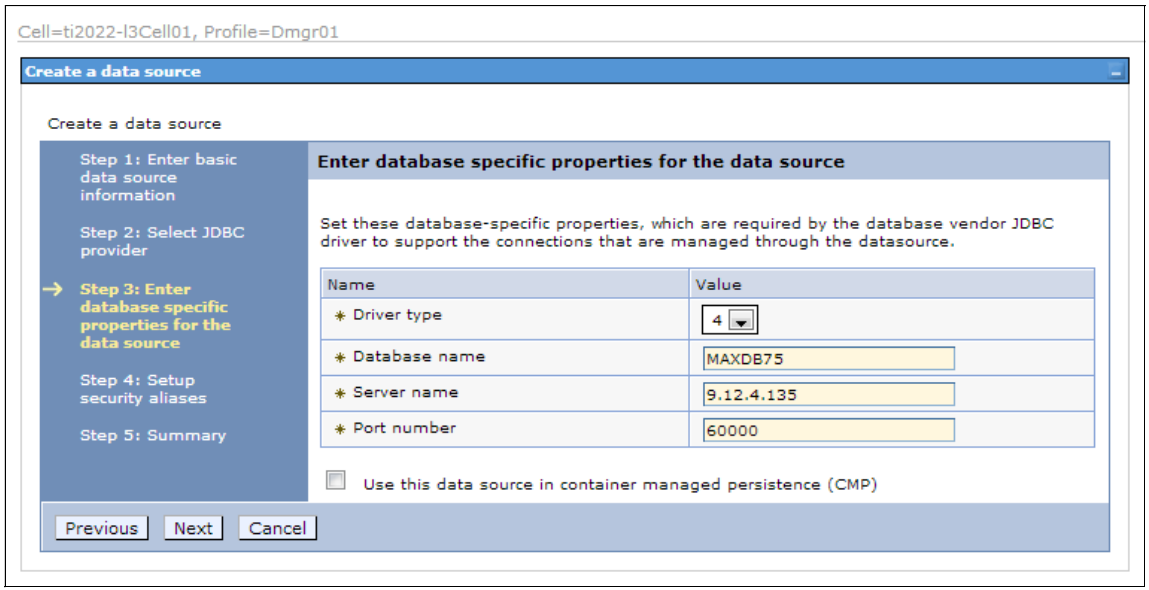

Figure 3-25 JDBC data source configuration

Attention: Database name, address and port number might vary from environment to environment. Check database information before creating the data source

25. Select the `maximo` alias created in Figure 3-26 on page 114 for ”Component-managed authentication alias” and select **Next**.

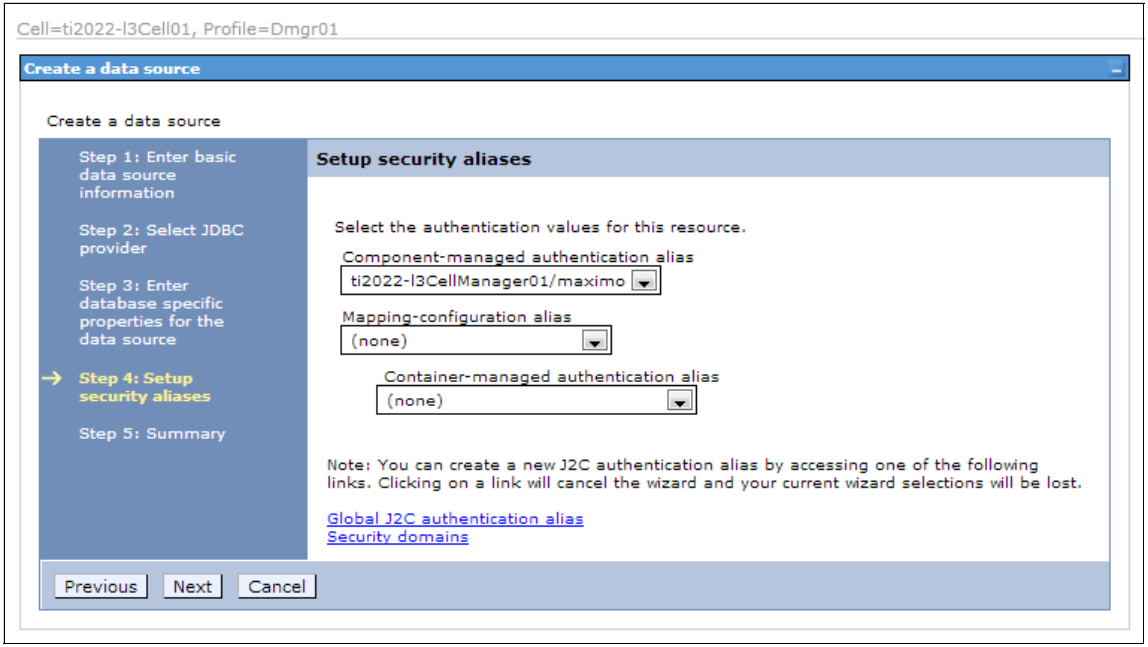

Figure 3-26 JDBC data source security alias

26. A summary table (Figure 3-27 on page 115) will be displayed with the data source information. Review and select **Finish**.

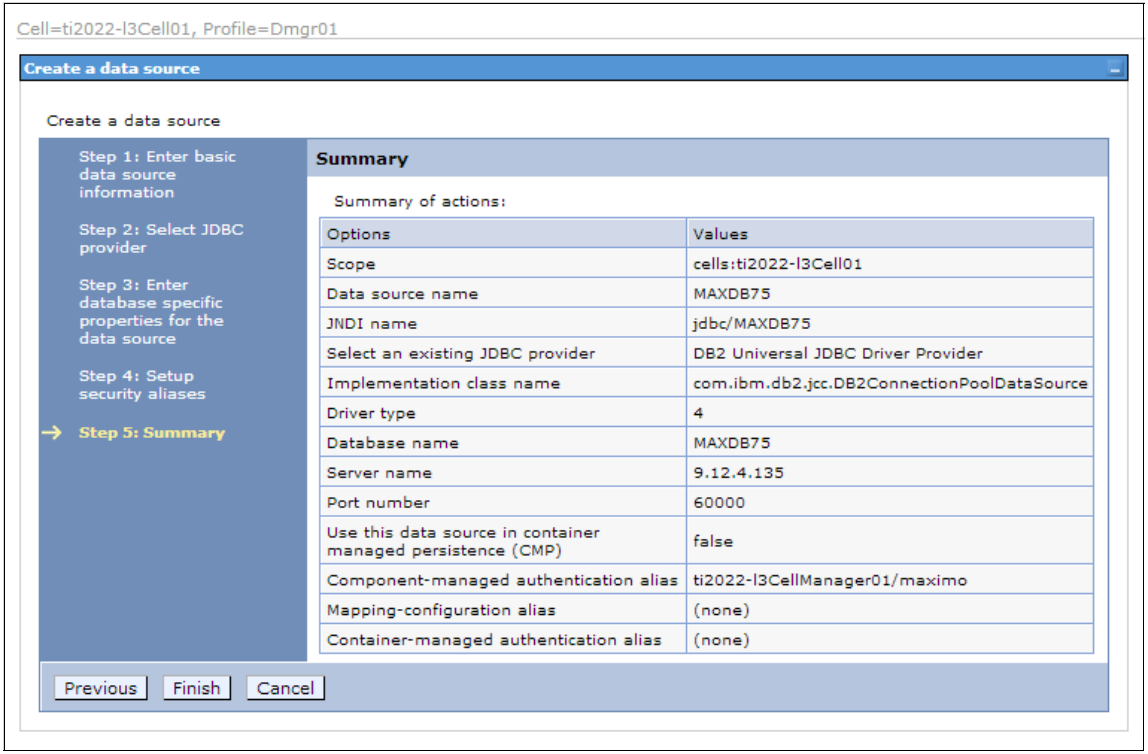

Figure 3-27 JDBC data source creation summary

27. Save and synchronize changes.
28. Navigate to **Service integration** → **Buses**.
29. Select **New**.
30. Type `intjmsbus` as the name for the new bus and uncheck “Bus Security”.
31. Select **Next**.
32. Select **Finish**.
33. Select the `intjmsbus`.
34. Select **Bus members**.
35. Select **Add**.
36. Select the cluster `SCCDMIF` as shown in Figure 3-28 on page 116.

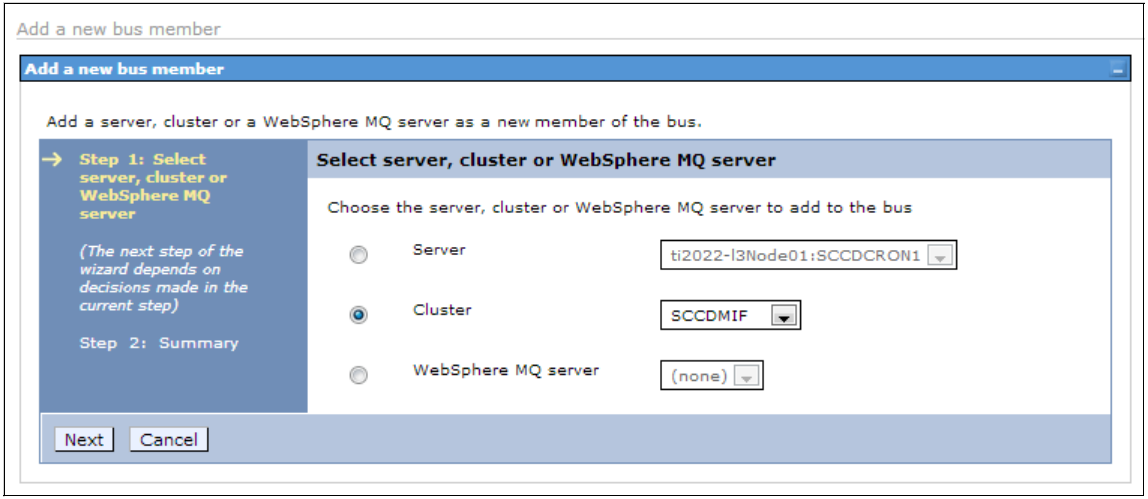

Figure 3-28 Bus member creation

37. Select **Next**.
38. Select the **High Availability** policy as shown in Figure 3-29 on page 117.

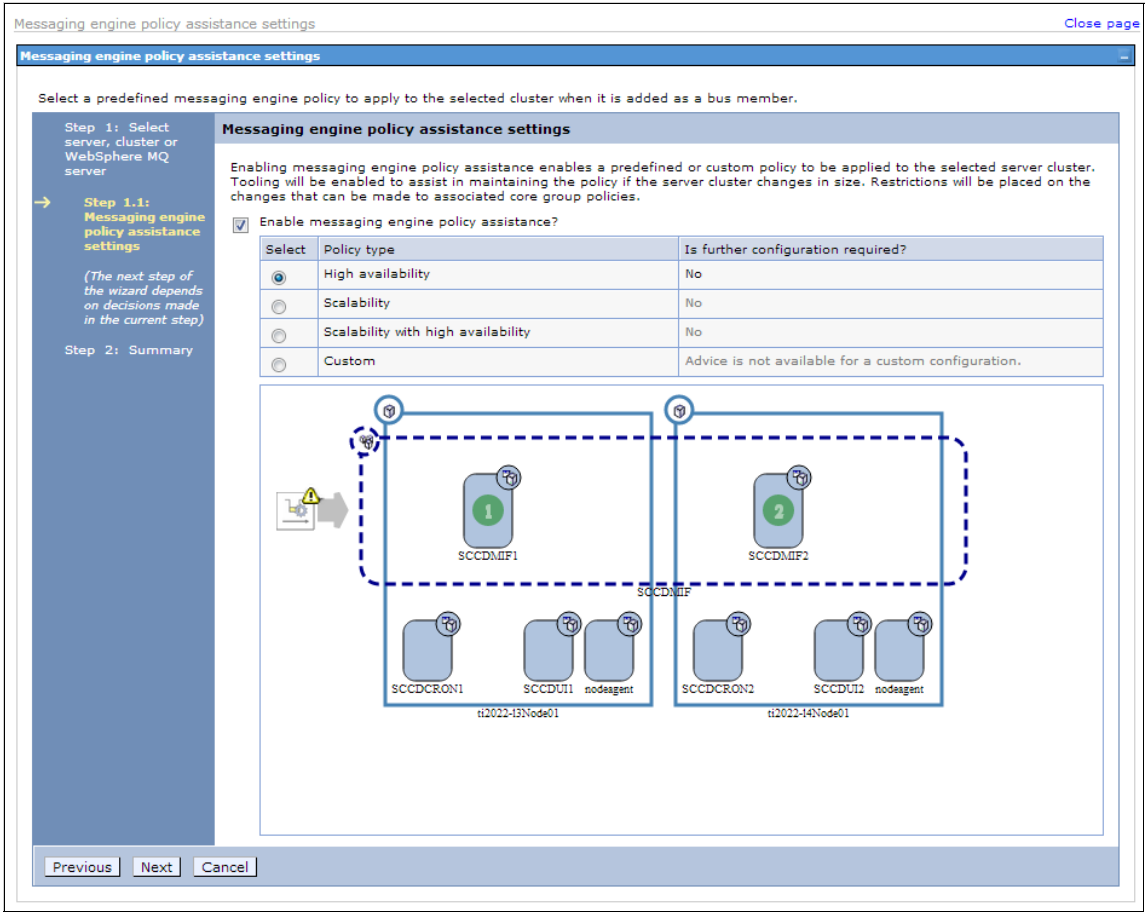

Figure 3-29 Bus member policy

39. Select **Next**.
40. Select **Data Store** and then **Next**.
41. Select **SCCDMIF.000-intjmsbus**.
42. Type `jdbc/MAXDB75` as the data source JNDI name, select the `maximo` alias as authentication alias and then select **Next** as shown in Figure 3-30 on page 118.

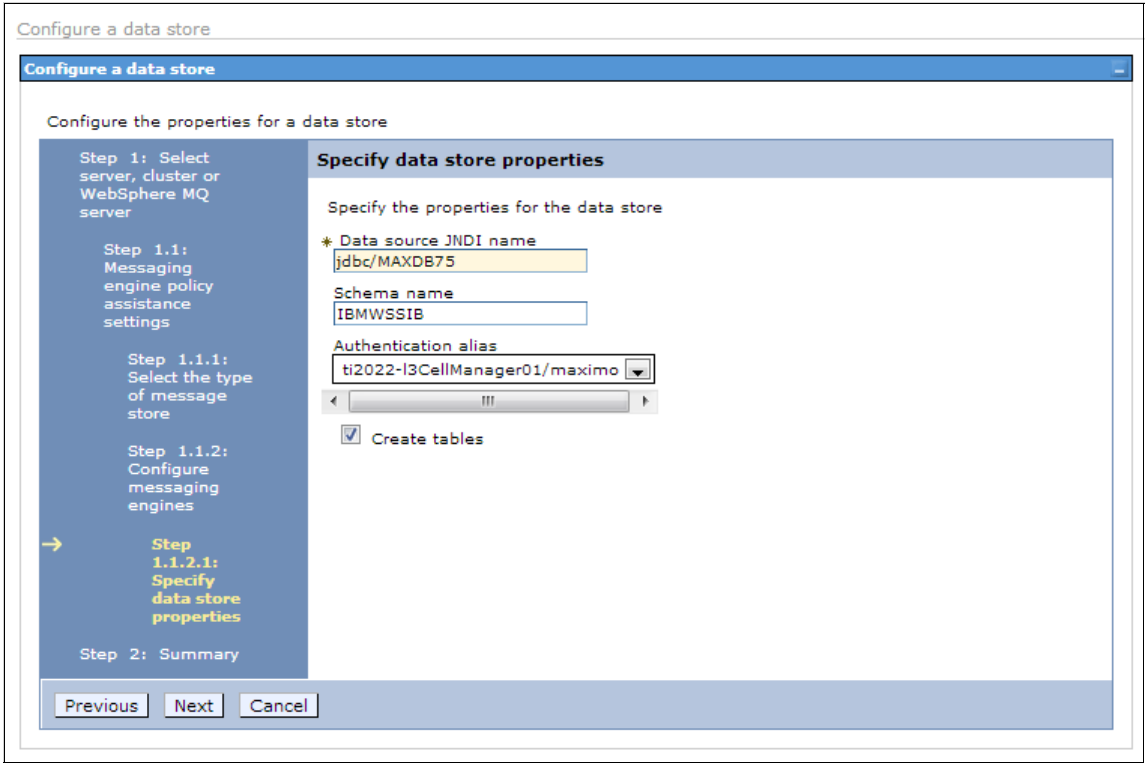

Figure 3-30 Bus member data source properties

43. Select **Next**.
44. Select **Next**.
45. A summary will be displayed with the bus member information. Review and select **Finish**.
46. Save and synchronize changes.
47. Navigate to **Service integration** → **Buses** → **intjmsbus** → **Destinations**.
48. Select **New**.
49. Select `Queue` and select **Next**.
50. Type `cqinbd` as Identifier and select **Next**.
51. Select `Cluster=SCCDMIF` as bus member and select **Next**.
52. A summary will be displayed with the destination information. Review and select **Finish**.
53. Repeat steps 48 through 52 for destinations `cqinerrbd`, `sqinbd` and `sqoutbd`.

54. Save and synchronize changes.
55. Navigate to **Resources** → **JMS** → **Queues**.
56. Select **Cell scope** and then **New**.
57. Select `Default messaging provider` and then **OK**.
58. Type `cqin` as the name and `jms/maximo/int/queues/cqin` as the JNDI name.
59. Select `intjmsbus` as bus name, `cqinbd` as queue name and thenselect **OK** as shown in Figure 3-31.

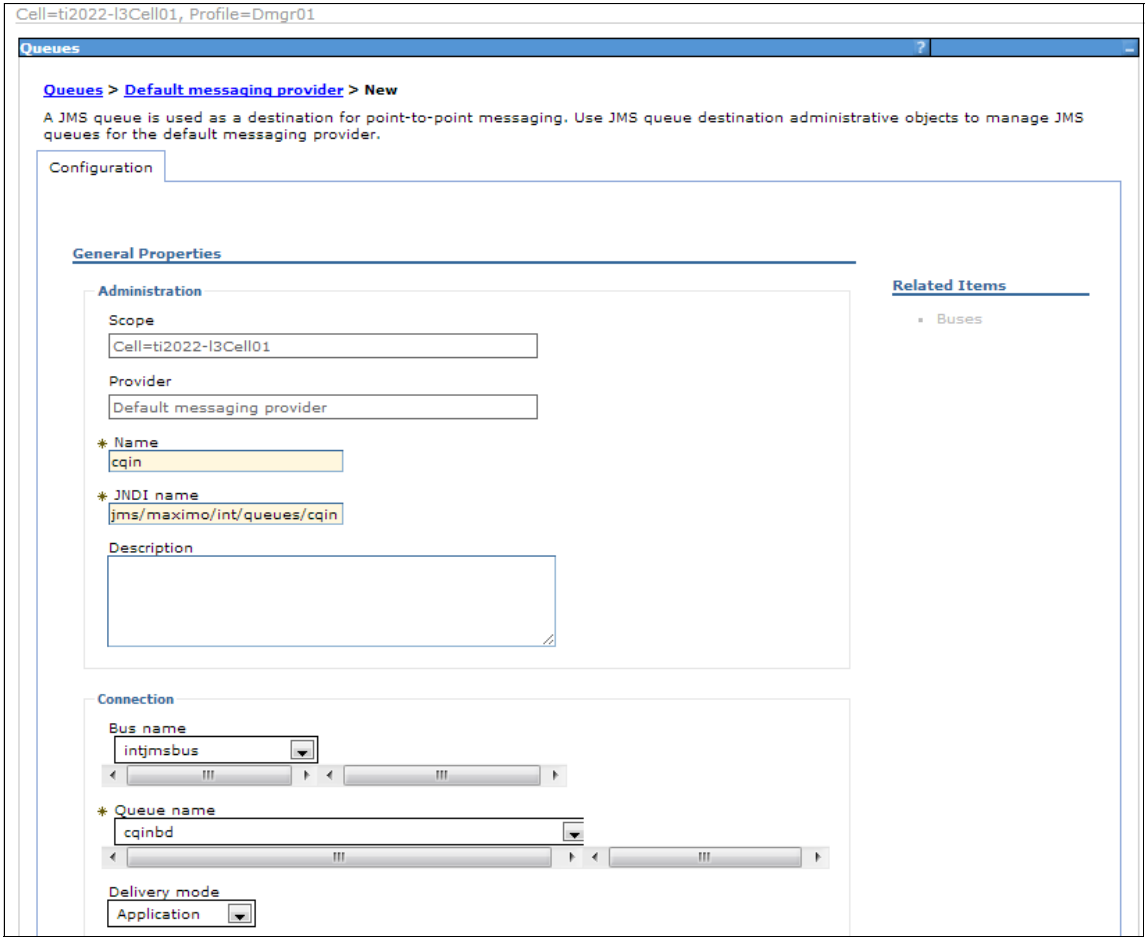

Figure 3-31 JMS queue creation

60. Repeat steps 56 on page 119 through 59 for queues `cqinerr`, `sqin` and `sqout`.
61. Save and synchronize changes.

62. After performing queue creations, the Queues panel should list the queues shown in Figure 3-32.

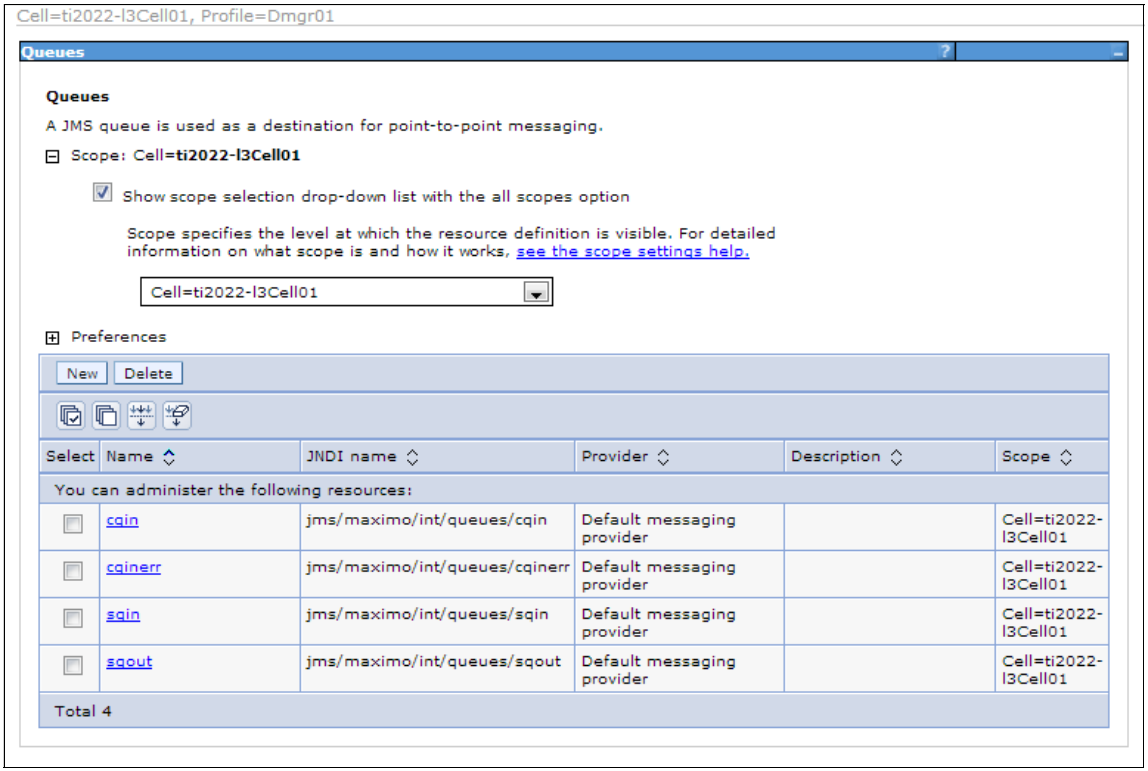

Figure 3-32 JMS Queues panel

63. Navigate to **Resources** → **JMS** → **Queue connection factories**.
64. Select **Cell scope** and then **New**.
65. Select `Default messaging provider` and then **OK**.
66. Type `intconfact` as the name and `jms/maximo/int/cf/intcf` as the JNDI name.
67. Select `intjmsbus` as bus name and then **OK** as shown in Figure 3-33 on page 121.

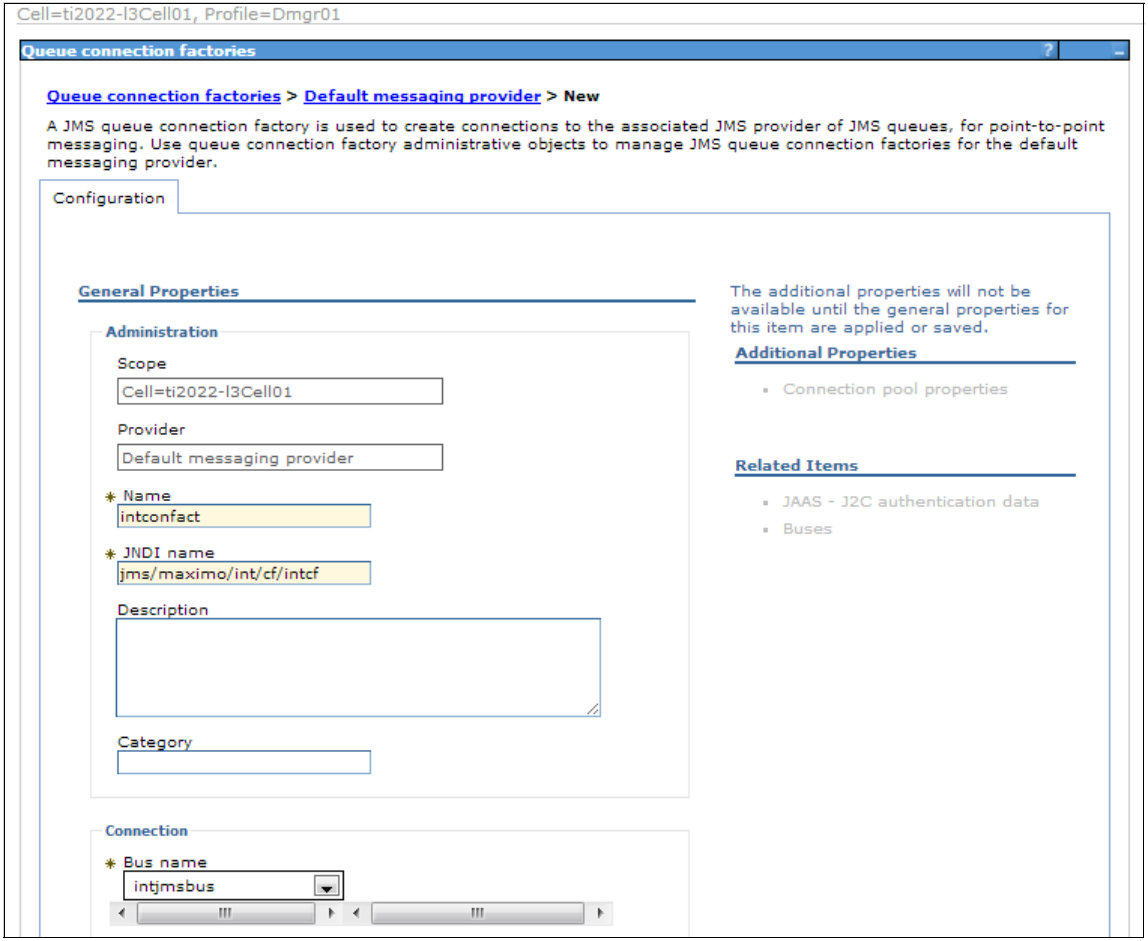

Figure 3-33 Queue connection factory creation

68. Save and synchronize changes.
69. Navigate to **Resources** → **JMS** → **Activation specifications**.
70. Select **Cell scope** and then **New**.
71. Select `Default messaging provider` and then **OK**.
72. Type `intjmsact` as the name, `intjmsact` as the JNDI name and `jms/maximo/int/queues/cqin` as the destination JNDI name.
73. Select `intjmsbus` as bus name and then select **OK** as shown in Figure 3-34 on page 122.

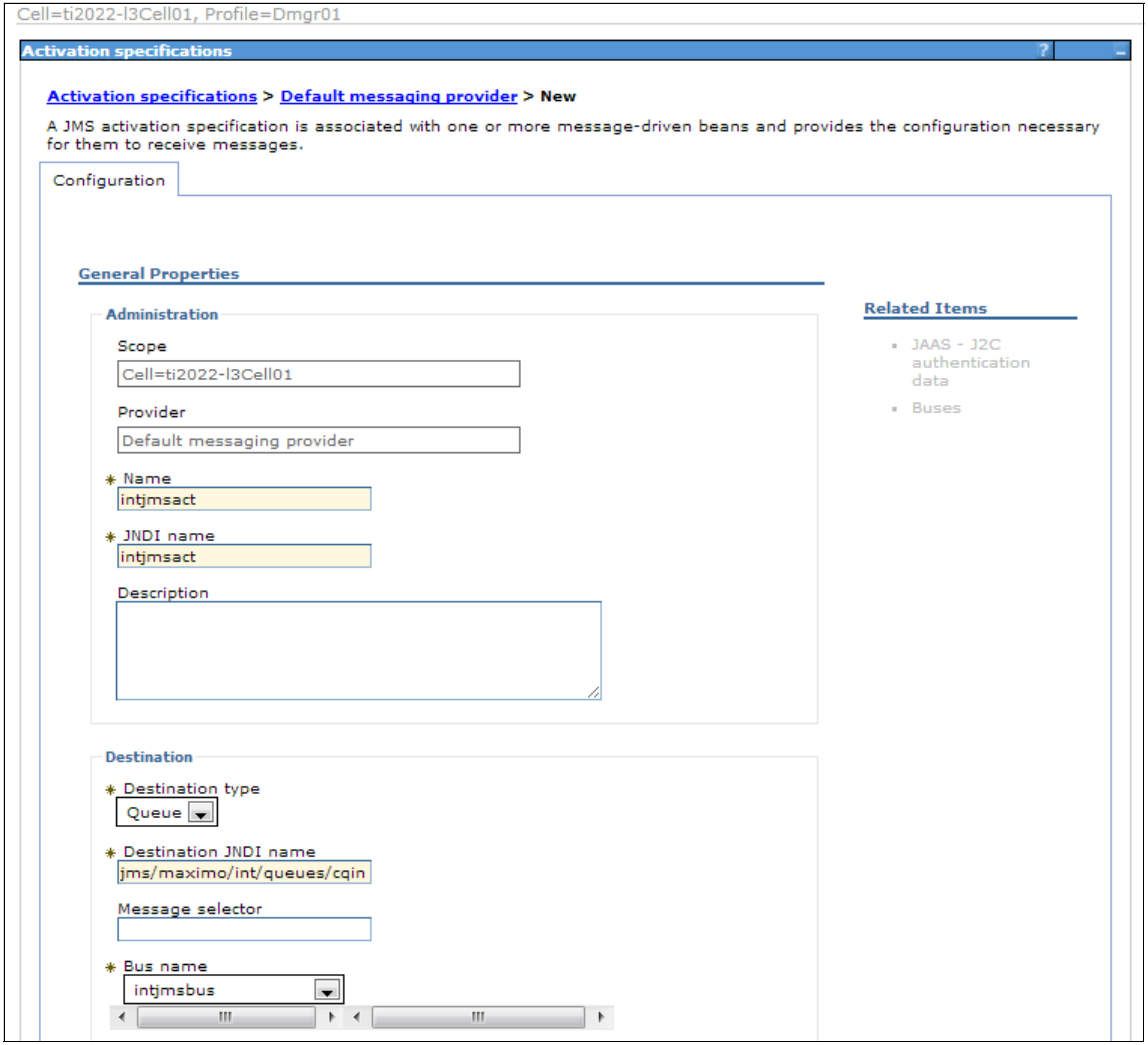

Figure 3-34 intjmsact activation specification creation

74. Select **Cell scope** and then **New**.
75. Select `Default messaging provider` and select **OK**.
76. Type `intjmsacterr` as the name, `intjmsacterr` as the JNDI name and `jms/maximo/int/queues/cqinerr` as the destination JNDI name.
77. Select `intjmsbus` as bus name and thenselect **OK** as shown in Figure 3-35 on page 123.

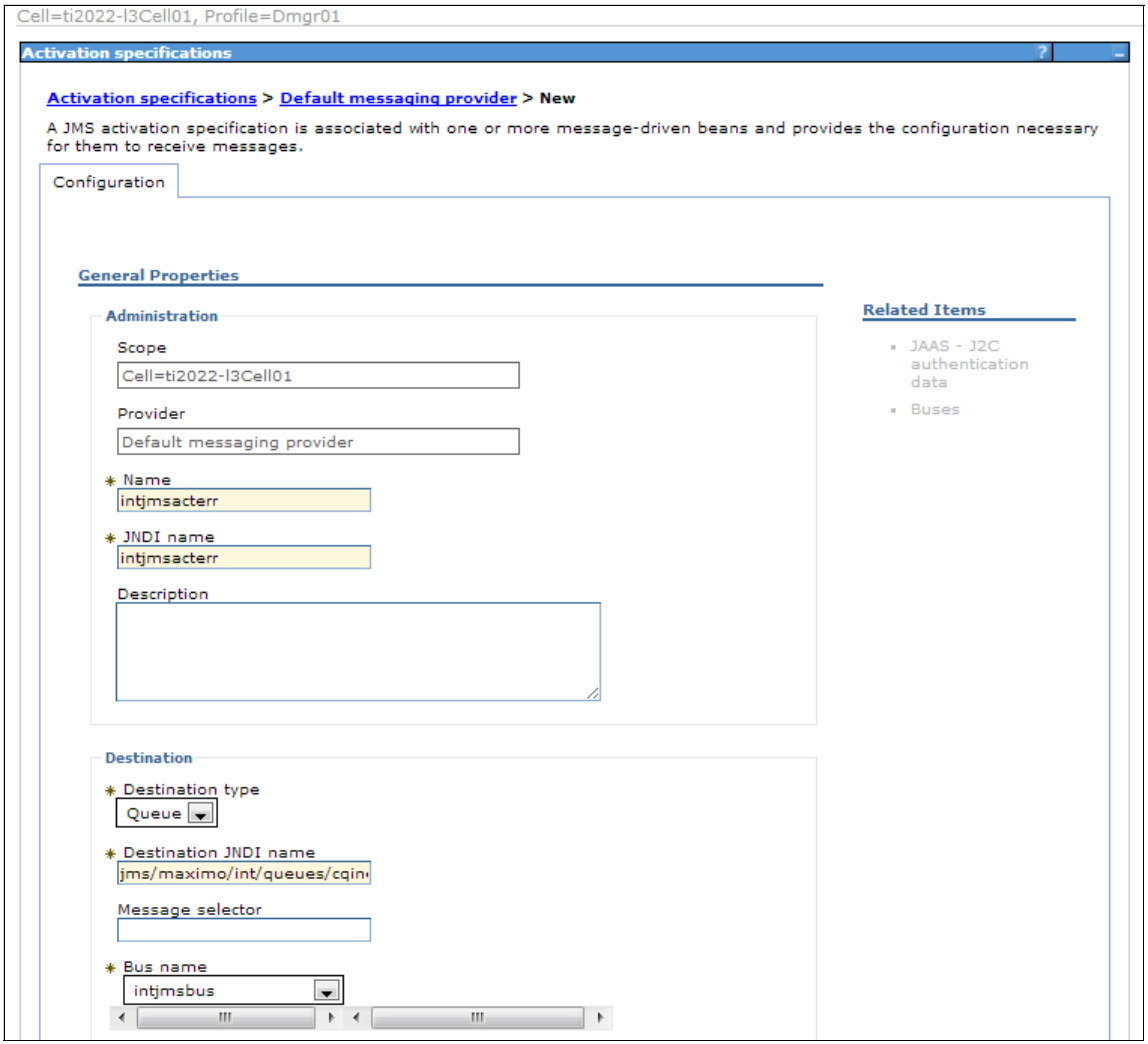

Figure 3-35 intjmsacterr activation specification creation

78. Save and synchronize changes.

WebSphere MQ configuration

The WebSphere MQ queue manager must be a highly available component to avoid a single point of failure.

For this section, the following assumptions are made:

- WebSphere MQ Server is already installed on both `mqhost1` and `mqhost2` servers.

Tip: For more information about WebSphere MQ installation, refer to the "Quick Beginnings" sections of the information center at:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp

- User mqm and group mqm have the same uid and gid, respectively, on both servers.
- The MQ_QM_PATH is *shared* and *mounted* on the same path on both servers.

Tip: For more information about WebSphere MQ multi-instance queue manager configuration, refer to:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=%2Fcom.ibm.mq.amqzag.doc%2Ffa70150_.htm

- WebSphere MQ Client is installed on all WebSphere Application Server nodes.

Multi-instance queue manager creation

The following steps cover the SCCDMIF multi-instance queue manager creation on mqhost1 and mqhost2.

1. Log in as mqm user on mqhost1.
2. Create the SCCDMIF queue manager using the **ctrmqm** command as shown in Example 3-72.

Example 3-72 SCCDMIF queue manager creation

```
mqm@ti2022-l11:~> crtmqm -md MQ_QM_DATA -ld MQ_QM_LOGS -u DLQ SCCDMIF
WebSphere MQ queue manager created.
Directory 'MQ_QM_DATA/SCCDMIF' created.
Creating or replacing default objects for SCCDMIF.
Default objects statistics : 65 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

3. Start the SCCDMIF queue manager using the **strmqm** command with the **-x** parameter to allow a *standby* queue manager, as shown in Example 3-73.

Example 3-73 SCCDMIF queue manger startup

```
mqm@ti2022-l11:~> strmqm -x SCCDMIF
WebSphere MQ queue manager 'SCCDMIF' starting.
5 log records accessed on queue manager 'SCCDMIF' during the log
replay phase.
```

```
Log replay for queue manager 'SCCDMIF' complete.
Transaction manager state recovered for queue manager 'SCCDMIF'.
WebSphere MQ queue manager 'SCCDMIF' started.
```

4. Check the status of the queue manager using the **dspmq** command with the **-x -o all** parameters to show details and multi-instance status as shown in Example 3-74.

Example 3-74 SCCDMIF status

```
mqm@ti2022-l11:~> dspmq -x -o all
QMNAME(SCCDMIF)       STATUS(Running) DEFAULT(no) STANDBY(Permitted)
    INSTANCE(ti2022-l11) MODE(Active)
ti2022-l11:~ #
```

5. Create a file named `mqsc_sccdmif.in` that will be used to define the queue manager listener and queues as shown in Example 3-75.

Example 3-75 Example mqsc_sccdmif.in file

```
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)

DEFINE QLOCAL(DLQ) MAXDEPTH(500000)
DEFINE QLOCAL(CQINBD) MAXDEPTH(50000)
DEFINE QLOCAL(CQINERRBD) MAXDEPTH(50000)
DEFINE QLOCAL(SQINBD) MAXDEPTH(50000)
DEFINE QLOCAL(SQOUTBD) MAXDEPTH(50000)

START LISTENER(LISTENER.TCP)
START CHANNEL(SYSTEM.ADMIN.SVRCONN)
```

Reminder: The `MAXDEPTH` values used are an example; these values can be changed to fit environment characteristics.

6. Run the `mqsc_sccdmif.in` file using the **runmqsc** command as shown in Example 3-76.

Example 3-76 Running mqsc_sccdmif.in file with the runmqsc command

```
mqm@ti2022-l11:~> runmqsc SCCDMIF < mqsc_sccdmif.in
5724-H72 (C) Copyright IBM Corp. 1994, 2009.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager SCCDMIF.

     1 : DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) PORT(1414)
CONTROL(QMGR)
```

```
AMQ8626: WebSphere MQ listener created.
     2 : DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)
AMQ8014: WebSphere MQ channel created.
       :
     3 : DEFINE QLOCAL(DLQ) MAXDEPTH(500000)
AMQ8006: WebSphere MQ queue created.
     4 : DEFINE QLOCAL(CQINBD) MAXDEPTH(50000)
AMQ8006: WebSphere MQ queue created.
     5 : DEFINE QLOCAL(CQINERRBD) MAXDEPTH(50000)
AMQ8006: WebSphere MQ queue created.
     6 : DEFINE QLOCAL(SQINBD) MAXDEPTH(50000)
AMQ8006: WebSphere MQ queue created.
     7 : DEFINE QLOCAL(SQOUTBD) MAXDEPTH(50000)
AMQ8150: WebSphere MQ object already exists.
       :
     8 : START LISTENER(LISTENER.TCP)
AMQ8730: Listener already active.
     9 : START CHANNEL(SYSTEM.ADMIN.SVRCONN)
AMQ8018: Start WebSphere MQ channel accepted.
9 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

7. Get the queue manager definition using the **dspmqinf** command with the **-o command** argument as shown in Example 3-77.

Example 3-77 SCCDMIF add command

```
mqm@ti2022-l11:~> dspmqinf -o command SCCDMIF
addmqinf -s QueueManager -v Name=SCCDMIF -v Directory=SCCDMIF -v
Prefix=/var/mqm -v DataPath=MQ_QM_DATA/SCCDMIF
```

8. Log in as `mqm` user on `mqhost2`.
9. Execute the add queue manager command generated in step 7 as shown in Example 3-78.

Example 3-78 Add SCCDMIF queue manager to mqhost2

```
mqm@ti2022-l9:~> addmqinf \
> -s QueueManager \
> -v Name=SCCDMIF \
> -v Directory=SCCDMIF \
> -v Prefix=/var/mqm \
> -v DataPath=MQ_QM_DATA/SCCDMIF
WebSphere MQ configuration information added.
```

10. Start the SCCDMIF queue manager using the **strmqm** command with the **-x** parameter to start as a *standby* queue manager, as shown in Example 3-79.

Example 3-79 Start SCCDMIF queue manager as standby

```
mqm@ti2022-l9:~> strmqm -x SCCDMIF
WebSphere MQ queue manager 'SCCDMIF' starting.
A standby instance of queue manager 'SCCDMIF' has been started.
The active instance is running elsewhere.
```

11. Check the queue manager status again on any of the servers. mqhost2 will now show as *standby,* as shown in Example 3-80.

Example 3-80 SCCDMIF status check

```
mqm@ti2022-l11:~> dspmq -x -o all
QMNAME(SCCDMIF)       STATUS(Running) DEFAULT(no) STANDBY(Permitted)
    INSTANCE(ti2022-l11) MODE(Active)
    INSTANCE(ti2022-l9) MODE(Standby)
```

JMS resources creation

After creating WebSphere MQ multi-instance queue managers, the corresponding JMS resource must be created in WebSphere Application Server.

Tip: For more information about WebSphere Application Server and WebSphere MQ inter-operation, refer to:

http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tmm_ep.html

The following steps describe the JMS resource creation:

1. Log into Integrated Solution Console and navigate to **Resources** → **JMS** → **JMS providers** → **WebSphere MQ messaging provider**.
2. Type /opt/mqm/java/lib as Native library path.

Important: The native library path varies for each operating system. For specific information, refer to:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=%2Fcom.ibm.mq.csqzaw.doc%2Fja10340_.htm

3. Select **OK**.
4. Save and synchronize changes.

5. Navigate to **Resources** → **JMS** → **Queues**.
6. Select **Cell scope** and then **New**.
7. Select `WebSphere MQ messaging provider` and select **OK**.
8. In the Administration section, type `cqin` as the name and `jms/maximo/int/queues/cqin` as the JNDI name.
9. In the WebSphere MQ Queue section, type `CQINBD` as the queue name and `SCCDMIF` as queue manager (Figure 3-36).

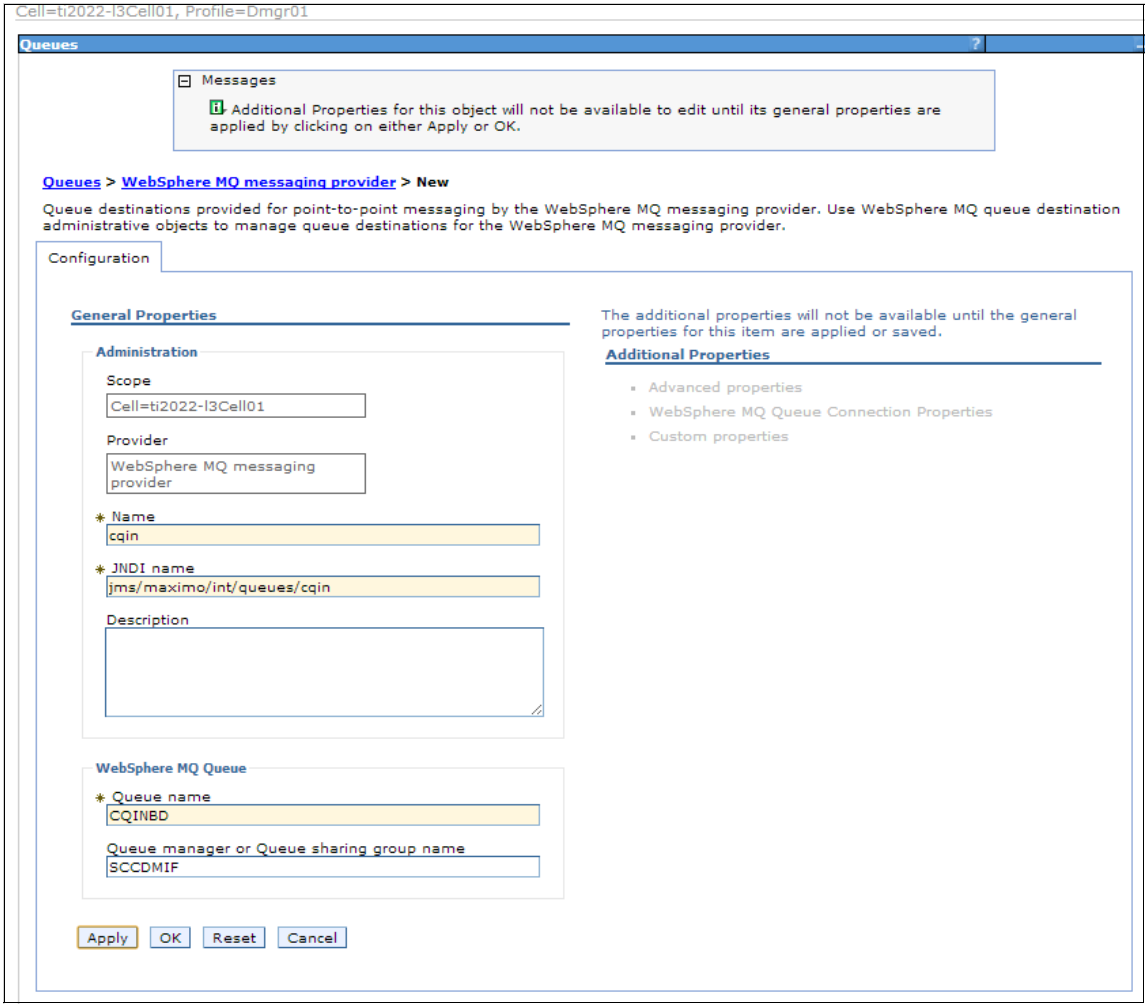

Figure 3-36 JMS MQ queue creation

10. Select **OK**.

11. Repeat steps 6 on page 128 through 10 for the queues `cqinerr`, `sqin`, and `sqout` (Figure 3-37).
12. Save and synchronize changes.

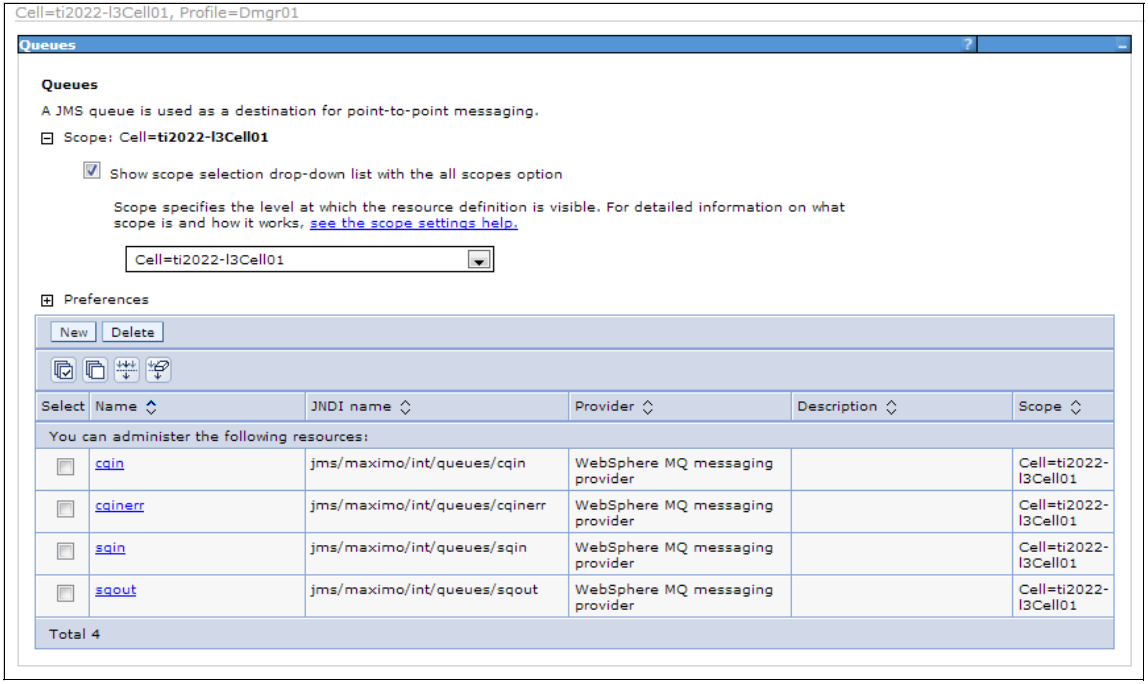

Figure 3-37 JMS MQ queues listing

13. Navigate to **Resources** → **JMS** → **Queue connection factories**.
14. Select **Cell scope** and then **New**.
15. Select `WebSphere MQ messaging provider` and select **OK**.
16. Type `intconfact` as name, `jms/maximo/int/cf/intcf` as JNDI name and select **Next**.
17. Select `Enter all the required information into this wizard` and select **Next**.
18. Type `SCCDMIF` as queue manager.
19. Select `Client` as transport.
20. Type `mqhost1` as hostname, `1414` as port, and `SYSTEM.DEF.SVRCONN` as server connection channel.
21. Select **Next**.
22. Select **Test connection**.

23. If the connection test result is not successful, try replacing `mqhost1` by its IP address. This value will be overridden later by the connection name list parameter.
24. Select **Next**.
25. A summary will be displayed; review and select **Finish**.
26. Select the recently created connection factory **intconfact**.
27. Select **Custom properties**.
28. Select **New**.
29. Type `XMSC_WMQ_CONNECTION_NAME_LIST` as name and `mqhost1(1414),mqhost2(1414)` as value.
30. Select **OK**.
31. Select **New**.
32. Type `CLIENTRECONNECTOPTIONS` as name and `QMGR` as value.
33. Select **OK**.
34. Select **New**.
35. Type `CLIENTRECONNECTTIMEOUT` as name and `900` as value. This value can be changed to meet environment characteristics.
36. Select **OK**.
37. Save and synchronize changes (Figure 3-38).

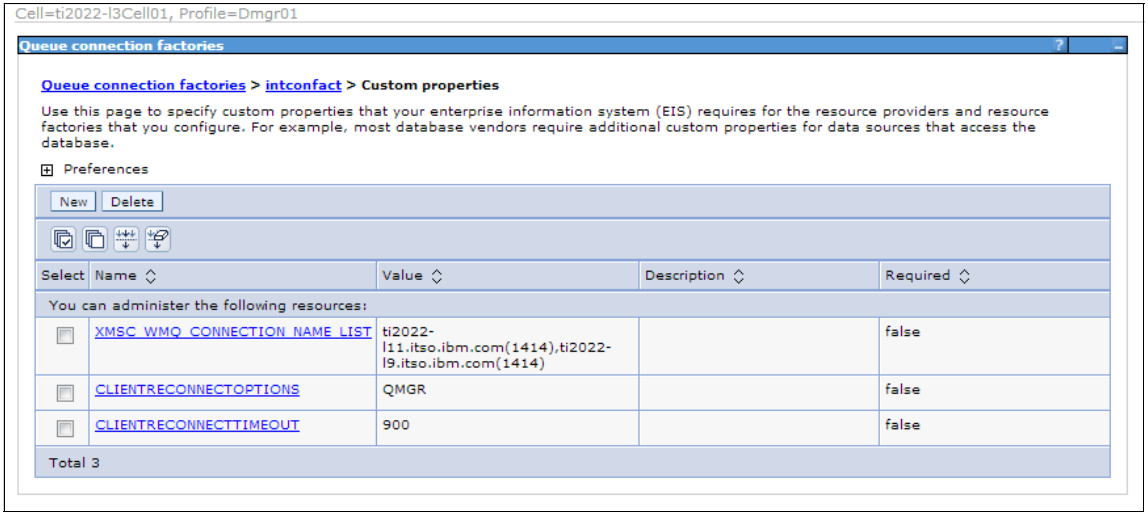

Figure 3-38 JMS queue connection factory custom properties

38. Navigate to **Resources** → **JMS** → **Activation specifications**.
39. Select **Cell Scope** and then **New**.
40. Select `WebSphere MQ messaging provider` and select **OK**.
41. Type `intjmsact` as name and JNDI name.
42. Select **Next**.
43. Type `jms/maximo/int/queues/cqin` as destination JNDI name and select **Next**.
44. Select `Enter all the required information into this wizard` and select **Next**.
45. Type `SCCDMIF` as queue manager.
46. Select `Client` as transport.
47. Type `mqhost1` as hostname, `1414` as port and `SYSTEM.DEF.SVRCONN` as server connection channel.
48. Select **Next**.
49. Select **Test connection**.
50. If the connection test result is not successful, try replacing `mqhost1` by its IP address. This value will be overridden later by the connection name list parameter.
51. Select **Next**.
52. A summary will be displayed; review and select **Finish**.
53. Select the recently created connection factory **intconfact**.
54. Select **Custom properties**.
55. Select **New**.
56. Type `connectionNameList` as name and `mqhost1(1414),mqhost2(1414)` as value.
57. Select **OK**.
58. Repeat steps 38 on page 131 through 57 for activation specification `intjmsacterr` using `jms/maximo/int/queues/cqinerr` as destination JNDI name.
59. Save and synchronize changes.

3.9 Failover testing

An important part of implementing a high availability topology is the failover testing. There are several types of failures you can simulate in your environment, which include system failure and process failure. You can also execute commands to perform a graceful failover between nodes. The examples in this section provide three common scenarios and the results of each. It is important to note how the IBM SmartCloud Control Desk user interface (UI) reacts to such failures.

- System failure
- Process failure
- Graceful failover

Other scenarios: Network failure and hardware failure testing should also be considered but the results and symptoms are very similar to the overall system failure testing. To simulate network failure, the Ethernet or fiber cable can be disconnected. Hardware failures such as storage failure can be simulated by carefully disconnecting them from the system, if possible. Although these scenarios are not specifically covered, they should be considered when testing.

3.9.1 Web server failover

The example here outlines some of the results and symptoms of failover when using IBM HTTP Server with Tivoli System Automation for Multiplatforms (SA MP) as a cluster manager.

System failure

By powering off the active node you can simulate an entire system failing. System Automation for Multiplatforms can detect that the IBM HTTP Server active system is offline and quickly restore services by bringing the passive node online.

1. Run the `lssam` command on one of the nodes to check which node is active.
2. Open a terminal or ssh connection to the passive node to monitor the status when the active system fails.
3. Power off the active system.
4. On the passive system, run the `lssam` command; it should look similar to the output in Figure 3-39 on page 133. This shows how the first node is detected

as offline by System Automation for Multiplatforms and the services are being restored on the second node.

```
ti2022-l2:~ # lssam
Online IBM.ResourceGroup:ihs-rg Nominal=Online
        |- Online IBM.Application:ihs-rs
                |- Failed offline IBM.Application:ihs-rs:ti2022-l1 Node=Offline
                '- Online IBM.Application:ihs-rs:ti2022-l2
        '- Online IBM.ServiceIP:ihs-ip
                |- Failed offline IBM.ServiceIP:ihs-ip:ti2022-l1 Node=Offline
                '- Online IBM.ServiceIP:ihs-ip:ti2022-l2
Online IBM.Equivalency:ihs-ip-equ
        |- Offline IBM.NetworkInterface:eth0:ti2022-l1 Node=Offline
        '- Online IBM.NetworkInterface:eth0:ti2022-l2
```

Figure 3-39 lssam output when a system fails

Process failure

When the IBM HTTP Server process terminates unexpectedly, System Automation for Multiplatforms will detect this and should attempt to bring the process back online on the current node. This failover sequence is often very fast and symptoms in the user interface are minimal.

1. Run the **lssam** command on one of the nodes to check which node is active.
2. Open a terminal or ssh connection on the active node to monitor the status when the process fails.
3. Determine the process ID of the HTTP server. Note that there may be multiple process IDs. On Linux, running **ps -ef | grep httpd** should show the process IDs.
4. Kill the active processes for httpd to simulate process failure. Running **kill -9** and listing all process IDs would work, or using a command such as:

   ```
   for pid in $(ps -ef |grep -v grep |grep httpd |awk '{print $2}'); do
   kill -9 $pid; done
   ```

5. Run the **lssam** command to view the status of the cluster. System Automation for Multiplatforms output should show that the HTTP server process is *pending online* on the same node. Figure 3-40 on page 134 shows the **lssam** output before and after killing the HTTP server processes.

```
ti2022-l1:~ # lssam
Online IBM.ResourceGroup:ihs-rg Nominal=Online
        |- Online IBM.Application:ihs-rs
                |- Online IBM.Application:ihs-rs:ti2022-l1
                '- Offline IBM.Application:ihs-rs:ti2022-l2
        '- Online IBM.ServiceIP:ihs-ip
                |- Online IBM.ServiceIP:ihs-ip:ti2022-l1
                '- Offline IBM.ServiceIP:ihs-ip:ti2022-l2
Online IBM.Equivalency:ihs-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l1
        '- Online IBM.NetworkInterface:eth0:ti2022-l2
ti2022-l1:~ # lssam
Pending online IBM.ResourceGroup:ihs-rg Nominal=Online
        |- Pending online IBM.Application:ihs-rs
                |- Pending online IBM.Application:ihs-rs:ti2022-l1
                '- Offline IBM.Application:ihs-rs:ti2022-l2
        '- Online IBM.ServiceIP:ihs-ip
                |- Online IBM.ServiceIP:ihs-ip:ti2022-l1
                '- Offline IBM.ServiceIP:ihs-ip:ti2022-l2
Online IBM.Equivalency:ihs-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l1
        '- Online IBM.NetworkInterface:eth0:ti2022-l2
ti2022-l1:~ #
```

Figure 3-40 Process failure lssam output

6. The failover sequence should complete and the IBM HTTP server should come back online on the same node. This procedure happens very fast, often without users noticing it.

Graceful failover

Sometimes it may be desirable to change the active node from one system to another. If the current active node requires maintenance or a reboot for example. Forcing a graceful failover can push the active node to the second system and users can continue to browse the application with minimal interruption.

1. Run the **`lssam`** command on one of the nodes to determine which is active.
2. Run the **`rgreq -o move ihs-rg`** command on either node to force a graceful failover.
3. Run **`lssam`** again to see that the resources (including the service IP) are moving to the second node.
4. Services should be restored quickly on the second node.

Symptoms of failover

Although the failover times will differ from one failure type to another, the symptoms in the user interface should be similar for all.

1. From a web browser, connect to the IBM SmartCloud Control Desk through the service IP or hostname.
2. Simulate a failure and try to retrieve records or interact with applications while the cluster manager is performing the failover sequence. When the “User Interface property setting” on page 104 is set, users in the UI should receive a pop-up dialog explaining that communication has been lost from the server. When communication is restored, they should be able to continue using their session. Figure 3-41 shows an example of the dialog the user will receive. When the second node is online and connection is re-established, another dialog indicating the connection has been restored will show. The user can now continue using the application with minimal interruption.

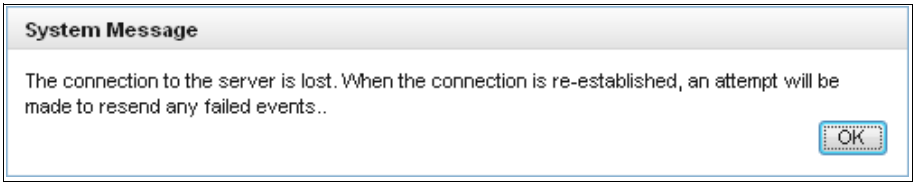

Figure 3-41 Pop-up dialog when the HTTP server fails

- If users try to browse to an application that is not cached in the browser session, they may get a blank page. Refreshing this page when the secondary node is online will often correct the problem.
- If users try to connect to the IBM SmartCloud Control Desk login page during the failover sequence, they may receive an error 503 from the browser. Refreshing when the secondary node is online should correct the problem.

3.9.2 Deployment manager failover

After configuring the WebSphere Application Server Deployment Manager for high availability, it should be tested to ensure the failover procedures happen as expected.

System failure

Simulating an entire system failure will prove that the deployment manager can successfully failover to the second node and resume administrative operations through the same service IP.

1. Run the `lssam` command on one of the nodes to determine which is the active deployment manager node.
2. Open a terminal or ssh connection to the passive node to monitor the status when the active system fails.

3. Power off the active node.
4. Run **lssam** again on the passive node. It should detect that the active node went offline and will bring the system online on the second node. Figure 3-42 shows the **lssam** output when the second node becomes active. Notice the *failed offline* status of the first node. The service IP should be applied on the second node.

```
ti2022-l4:~ # lssam
Online IBM.ResourceGroup:dmgr-rg Nominal=Online
        |- Online IBM.Application:dmgr-jvm
                |- Failed offline IBM.Application:dmgr-jvm:ti2022-l3 Node=Offline
                '- Online IBM.Application:dmgr-jvm:ti2022-l4
        '- Online IBM.ServiceIP:dmgr-ip
                |- Failed offline IBM.ServiceIP:dmgr-ip:ti2022-l3 Node=Offline
                '- Online IBM.ServiceIP:dmgr-ip:ti2022-l4
Failed offline IBM.ResourceGroup:nodeagent-ti2022-l3-rg Nominal=Online
        '- Failed offline IBM.Application:nodeagent-ti2022-l3
                '- Failed offline IBM.Application:nodeagent-ti2022-l3:ti2022-l3 Node=Offline
Online IBM.ResourceGroup:nodeagent-ti2022-l4-rg Nominal=Online
        '- Online IBM.Application:nodeagent-ti2022-l4
                '- Online IBM.Application:nodeagent-ti2022-l4:ti2022-l4
Online IBM.Equivalency:dmgr-ip-equ
        |- Offline IBM.NetworkInterface:eth0:ti2022-l3 Node=Offline
        '- Online IBM.NetworkInterface:eth0:ti2022-l4
```

Figure 3-42 lssam output when a system fails

Process failure

Simulating an unexpected crash of the deployment manager process on the active node will show how System Automation for Multiplatforms reacts. The process should be quickly brought back online on the same node.

1. Run the **lssam** command on one of the deployment manager nodes to determine which node is active.
2. Connect to a terminal or ssh session on the active node to monitor the status
3. Determine the process ID of the deployment manager. On Linux running **ps -ef | grep dmgr** should show the process ID.
4. Kill the dmgr process by running **kill -9** on the process ID.
5. Running **lssam** should show that the process is *pending online* and will be brought online on the same node. Figure 3-43 on page 137 shows an example **lssam** output for a process failure.

```
ti2022-l3:~ # lssam
Pending online IBM.ResourceGroup:dmgr-rg Request=Move Nominal=Online
        |- Pending online IBM.Application:dmgr-jvm
                |- Pending online IBM.Application:dmgr-jvm:ti2022-l3
                '- Offline IBM.Application:dmgr-jvm:ti2022-l4
        '- Online IBM.ServiceIP:dmgr-ip
                |- Online IBM.ServiceIP:dmgr-ip:ti2022-l3
                '- Offline IBM.ServiceIP:dmgr-ip:ti2022-l4
Online IBM.ResourceGroup:nodeagent-ti2022-l3-rg Nominal=Online
        '- Online IBM.Application:nodeagent-ti2022-l3
                '- Online IBM.Application:nodeagent-ti2022-l3:ti2022-l3
Online IBM.ResourceGroup:nodeagent-ti2022-l4-rg Nominal=Online
        '- Online IBM.Application:nodeagent-ti2022-l4
                '- Online IBM.Application:nodeagent-ti2022-l4:ti2022-l4
Online IBM.Equivalency:dmgr-ip-equ
        |- Online IBM.NetworkInterface:eth0:ti2022-l3
        '- Online IBM.NetworkInterface:eth0:ti2022-l4
```

Figure 3-43 lssam output when the dmgr process is killed

Graceful takeover

If you need to move the active node to the second machine, you can perform a graceful takeover.

1. Run the **`lssam`** command to determine which is the active deployment manager node.
2. Run the **`rgreq -o move dmgr-rg`** command to force the deployment manager to move to the second node.
3. Running **`lssam`** again should show that the deployment manager has moved to the second node.

Symptoms of failover

The main symptom of failover for the deployment manager is a loss of connection to the deployment manager console. Administrators who may be logged into the console at the time of failover will receive an error in the web browser indicating that the page cannot be displayed. When the failover sequence completes, administrators should be able to log back into the deployment manager and resume operations.

There may be a brief moment when the application server status does not show properly in the deployment manager console. Synchronizing the nodes should correct this problem.

3.9.3 WebSphere Application Server nodeagent testing

Adding the resource groups for the nodeagents is a simple restart scenario only with Tivoli System Automation for Multiplatforms. The *nominal status* of the

nodeagents at all times is online. If a nodeagent process crashes or the server restarts, System Automation for Multiplatforms can detect and restart this process.

System failure

If the system fails, System Automation for Multiplatforms will show the nodeagent in a *failed offline* status. When the system comes back online, System Automation for Multiplatforms attempts to restart the nodeagent process. Figure 3-44 shows an example of **lssam** output when the nodeagent system fails. The remaining nodeagent stays online unless it fails as well.

```
ti2022-l4:~ # lssam
Online IBM.ResourceGroup:dmgr-rg Nominal=Online
        |- Online IBM.Application:dmgr-jvm
                 |- Failed offline IBM.Application:dmgr-jvm:ti2022-l3 Node=Offline
                 '- Online IBM.Application:dmgr-jvm:ti2022-l4
        '- Online IBM.ServiceIP:dmgr-ip
                 |- Failed offline IBM.ServiceIP:dmgr-ip:ti2022-l3 Node=Offline
                 '- Online IBM.ServiceIP:dmgr-ip:ti2022-l4
Failed offline IBM.ResourceGroup:nodeagent-ti2022-l3-rg Nominal=Online
        '- Failed offline IBM.Application:nodeagent-ti2022-l3
                 '- Failed offline IBM.Application:nodeagent-ti2022-l3:ti2022-l3 Node=Offline
Online IBM.ResourceGroup:nodeagent-ti2022-l4-rg Nominal=Online
        '- Online IBM.Application:nodeagent-ti2022-l4
                 '- Online IBM.Application:nodeagent-ti2022-l4:ti2022-l4
Online IBM.Equivalency:dmgr-ip-equ
        |- Offline IBM.NetworkInterface:eth0:ti2022-l3 Node=Offline
        '- Online IBM.NetworkInterface:eth0:ti2022-l4
```

Figure 3-44 lssam output for the nodeagent system failure

Process failure

If the nodeagent process on either server terminates unexpectedly, System Automation for Multiplatforms will attempt to restart the process as soon as it realizes this. The *pending online* status of the failed nodeagent will show until the nodeagent process is back online.

3.9.4 WebSphere Application Server application server failover

IBM SmartCloud Control Desk does not support user interface (UI) session failover. For this reason, when an application server fails, any users connected to this will be redirected to another JVM. When using Integration Framework through the WebSphere Service Integration Bus (SIB) the Messaging Engine (ME) will need to failover to another JVM if the active ME JVM goes down.

System failure

When the WebSphere Application Server system fails, the corresponding applications will no longer show online on the Deployment Manager console. The corresponding nodeagent on the same machine will also fail with system failure, so when the system comes back online the nodeagent may restart automatically, depending on the policy you have configured. It is possible to have the

application server JVMs start with the nodeagent by configuring the monitoring policy on the application server (Figure 3-45 on page 139). To configure the application servers to start with the nodeagents, you can follow this procedure:

1. In the deployment manager console, log in and go to **Servers** → **Server Types** → **WebSphere application servers** → **SERVER_NAME**.
2. In the Server Infrastructure section, click **Java and Process Management** → **Monitoring policy**.
3. In the General Properties section, change the Node restart state to RUNNING.

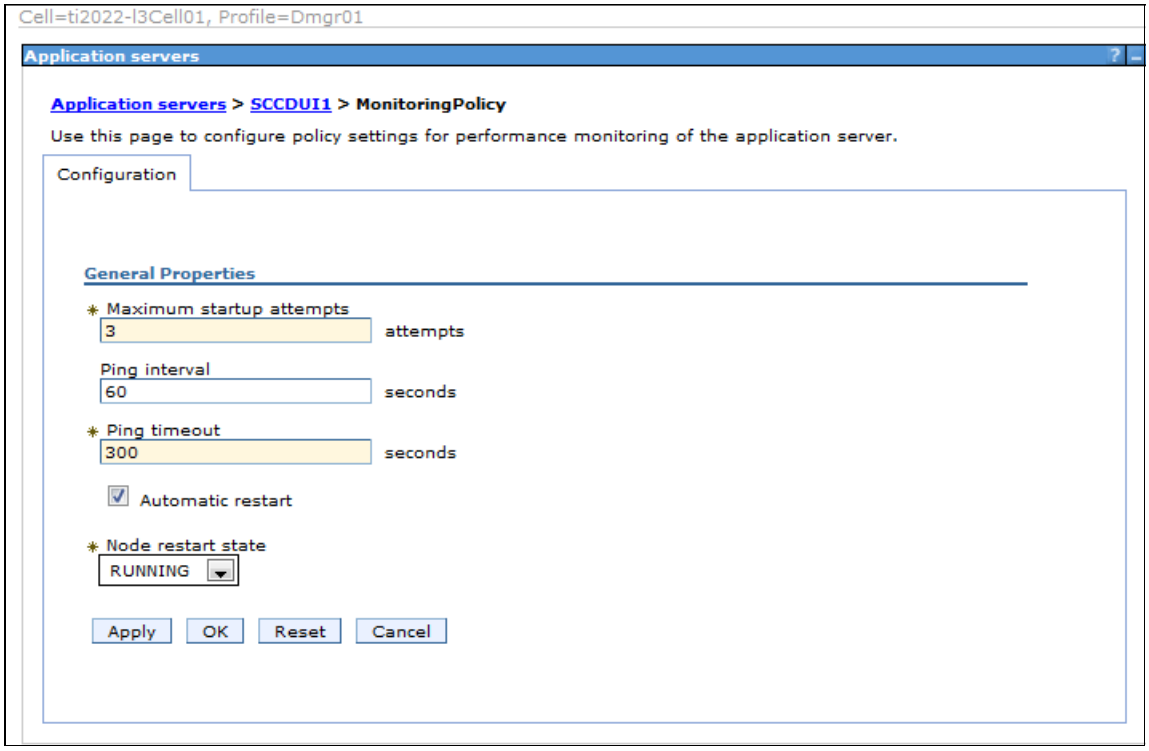

Figure 3-45 MonitoringPolicy for application server startup

Tip: Auto-starting application servers may not be ideal in many environments. Some administrators may have application servers that should not be running all the time, or would like to control the status. Ensure that this configuration is best for your organization before implementing it.

Process failure

By default, the WebSphere Application Server Network Deployment HAManager will attempt to restart any application servers that have crashed or terminated unexpectedly. When a process fails, WebSphere should restart the application server automatically.

Symptoms of failover

When a user session is active on the WebSphere application server, this user session is persistent to that application server. If this application server were to fail or restart (Figure 3-46), the user session would be terminated and the user will be directed to another application server. IBM SmartCloud Control Desk does not support session failover, so the user will have to log back in to the application.

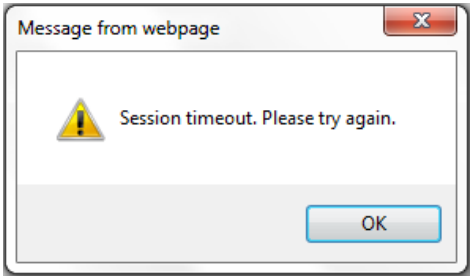

Figure 3-46 Shows an error the user may receive when an application server fails

Another symptom of application server failure is the potential performance decrease because more users will now be using the same application server JVM. If multiple application servers fail, performance could be noticeably impacted.

3.9.5 WebSphere Application Server messaging engine failover

IBM SmartCloud Control Desk relies on the application server messaging engine to process integration transactions. For this reason, when the messaging engine fails, any messages on any JMS queue will not be processed until the messaging engine is started properly.

Determining in which server the messaging engine is started

In order to determine on which application server the messaging engine is started, the `wsadmin` needs to be utilized. The steps to accomplish that are:

1. Log in to Integrated Solutions Console and navigate to **Service integration** → **Buses** → **intjmsbus** → **Messaging engines**.

2. The messaging engine will be displayed. Its name (Figure 3-47) will be used to query its current application server process through the `wsadmin` tool. For this example, the value used is `SCCDMIF.000-intjmsbus`.

Figure 3-47 Messaging engine name

3. Create a script file as shown in Example 3-81. If needed, modify the `meName` variable with the current messaging engine name. For this example, the script file is `WAS_DMGR_PATH/bin/sibMEProcess.py`.

Example 3-81 Messaging engine application server query script example

```
#Messaging engine name
meName = 'SCCDMIF.000-intjmsbus'
meObjName = AdminControl.makeObjectName('WebSphere:type=SIBMessagingEngine,name=' + \
        meName + ',*')
meMBean = AdminControl.queryNames_jmx(meObjName, None)[0]
print meName + ' is running on ' + meMBean.getKeyProperty('process') + ' at ' + \
        meMBean.getKeyProperty('node')
```

4. Run the script with the `wsadmin` tool; the output will show the application server running the messaging engine, as shown in Example 3-82.

Example 3-82 Script output

```
ti2022-l3:WAS_DMGR_PATH/bin # ./wsadmin.sh  \
> -lang jython \
> -user admin \
```

```
> -password admin \
> -f sibMEProcess.py
WASX7209I: Connected to process "dmgr" on node ...
SCCDMIF.000-intjmsbus is running on SCCDMIF2 at ti2022-14Node01
```

System failure

When the WebSphere Application Server system fails, the corresponding application severs will no longer show online in the Integrated Solution Console. If the messaging engine is running in one of the application servers affected, it will need to failover to another application server.

Determine the messaging engine process as described in “Determining in which server the messaging engine is started” on page 140 and shut down its corresponding server. The WebSphere Application Server should failover the messaging engine to another application server in the cluster and provide access to JMS queues normally.

Process failure

By default, the WebSphere Application Server Network Deployment HAManager will attempt to restart any application servers that have crashed or terminated unexpectedly. When a process fails, WebSphere should restart the application server automatically. During its restart, the messaging engine should failover to another application server in the cluster and provide access to JMS queues normally.

Symptoms of failover

Due to database locking mechanisms, when an unexpected process termination occurs, the connections holding the locks are not released. The solution for this situation is tweaking DB2 server operating systemTCP “keep alive” parameters.

During messaging engine startup, it will try to obtain the lock on datastore tables for 15 minutes. If not possible, the messaging engine will be disabled. Make sure your TCP keep alive parameters are set to release these idle connections in less than 15 minutes.

For operating system-related configuration for DB2, refer to:

http://www-01.ibm.com/support/docview.wss?uid=swg21231084

Tip: For more information about how the WebSphere Application Server messaging engine works, refer to:

http://www-01.ibm.com/support/docview.wss?uid=swg27020333

If the application server that is running the JMS cron tasks is affected by a system outage, the cron task will failover to another node in the cron task cluster. This failover takes approximately 5 minutes to complete, so messages will not be consumed until the cron task failover is complete.

3.9.6 Database failover

IBM SmartCloud Control Desk application failover scenarios were tested by simulating database failover. The application availability was tested using DB2 using HADR and shared disk configuration.

Database HADR failover testing

The following failover scenarios were tested with the DB2 HADR setup:

System failure

This scenario was tested by powering down the primary database server. The entire workload was transferred to the secondary database server by the cluster manager. The following steps were executed:

1. Run the **lssam** command as root from either the primary or the secondary database server. Figure 3-48 shows the output of the **lssam** command in the normal operating environment.

```
ti2022-l5:~ # lssam
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_MAXDB75-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs
                |- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:ti2022-l5
                '- Offline IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:ti2022-l6
        '- Online IBM.ServiceIP:db2ip_9_12_4_135-rs
                |- Online IBM.ServiceIP:db2ip_9_12_4_135-rs:ti2022-l5
                '- Offline IBM.ServiceIP:db2ip_9_12_4_135-rs:ti2022-l6
Online IBM.ResourceGroup:db2_db2inst1_ti2022-l5_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_ti2022-l5_0-rs
                '- Online IBM.Application:db2_db2inst1_ti2022-l5_0-rs:ti2022-l5
Online IBM.ResourceGroup:db2_db2inst1_ti2022-l6_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_ti2022-l6_0-rs
                '- Online IBM.Application:db2_db2inst1_ti2022-l6_0-rs:ti2022-l6
Online IBM.Equivalency:db2_db2inst1_db2inst1_MAXDB75-rg_group-equ
        |- Online IBM.PeerNode:ti2022-l5:ti2022-l5
        '- Online IBM.PeerNode:ti2022-l6:ti2022-l6
Online IBM.Equivalency:db2_db2inst1_ti2022-l5_0-rg_group-equ
        '- Online IBM.PeerNode:ti2022-l5:ti2022-l5
Online IBM.Equivalency:db2_db2inst1_ti2022-l6_0-rg_group-equ
        '- Online IBM.PeerNode:ti2022-l6:ti2022-l6
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:ti2022-l5
        '- Online IBM.NetworkInterface:eth0:ti2022-l6
ti2022-l5:~ #
```

Figure 3-48 lssam output for a normal operating environment

2. Log on to the IBM SmartCloud Control Desk application and navigate to one of the application panels.

3. Shut down the primary database server. Power off the server from the console as if the server crashed and powered off. Run **lssam** on the secondary server to see the behavior of the system. Figure 3-49 displays the output of **lssam** in case of a server failure.

```
ti2022-l6:~ # lssam
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_MAXDB75-rg Request=Lock Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs Control=SuspendedPropagated
                |- Failed offline IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:ti2022-l5 Node=Offline
                '- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:ti2022-l6
        '- Online IBM.ServiceIP:db2ip_9_12_4_135-rs Control=SuspendedPropagated
                |- Failed offline IBM.ServiceIP:db2ip_9_12_4_135-rs:ti2022-l5 Node=Offline
                '- Online IBM.ServiceIP:db2ip_9_12_4_135-rs:ti2022-l6
Failed offline IBM.ResourceGroup:db2_db2inst1_ti2022-l5_0-rg Nominal=Online
        '- Failed offline IBM.Application:db2_db2inst1_ti2022-l5_0-rs
                '- Failed offline IBM.Application:db2_db2inst1_ti2022-l5_0-rs:ti2022-l5 Node=Offline
Online IBM.ResourceGroup:db2_db2inst1_ti2022-l6_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_ti2022-l6_0-rs
                '- Online IBM.Application:db2_db2inst1_ti2022-l6_0-rs:ti2022-l6
Online IBM.Equivalency:db2_db2inst1_db2inst1_MAXDB75-rg_group-equ
        |- Offline IBM.PeerNode:ti2022-l5:ti2022-l5 Node=Offline
        '- Online IBM.PeerNode:ti2022-l6:ti2022-l6
Online IBM.Equivalency:db2_db2inst1_ti2022-l5_0-rg_group-equ
        '- Offline IBM.PeerNode:ti2022-l5:ti2022-l5 Node=Offline
Online IBM.Equivalency:db2_db2inst1_ti2022-l6_0-rg_group-equ
        '- Online IBM.PeerNode:ti2022-l6:ti2022-l6
Online IBM.Equivalency:db2_public_network_0
        |- Offline IBM.NetworkInterface:eth0:ti2022-l5 Node=Offline
        '- Online IBM.NetworkInterface:eth0:ti2022-l6
```

Figure 3-49 lssam output in case of a server failure

4. The IBM SmartCloud Control Desk session hangs for a short interval while the cluster manager transfers the workload to the secondary server. In one of the tests, the IBM SmartCloud Control Desk session was lost. In that case relog in to the application and resume work. All the transactions that were not committed would be lost or rolled back.
5. All the resources are now transferred to the secondary server. When the primary server comes back up, the old primary server will be added back to the cluster manager and monitored.

Process failure

This scenario was tested by simulating the DB2 server process failure. The database server instance was shut down while the application was connected. The cluster manager detected that the DB2 server process was down and restarted the process. The following steps were executed:

1. Run the **lssam** command as root from either the primary or the secondary database server. The output should indicate normal operation.
2. Log on to the IBM SmartCloud Control Desk application and navigate to one of the application screens.
3. Issue the **db2_kill** command to abruptly end all the DB2 server processes. Run **lssam** as root user to list the status of the cluster. Figure 3-50 on page 145 shows the **lssam** output during the DB2 server process failure.

```
ti2022-l5:~ # lssam
Pending online IBM.ResourceGroup:db2_db2inst1_db2inst1_MAXDB75-rg Nominal=Online
        |- Pending online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs
                |- Pending online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:ti2022-l5
                '- Offline IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:ti2022-l6
        '- Online IBM.ServiceIP:db2ip_9_12_4_135-rs
                |- Online IBM.ServiceIP:db2ip_9_12_4_135-rs:ti2022-l5
                '- Offline IBM.ServiceIP:db2ip_9_12_4_135-rs:ti2022-l6
Online IBM.ResourceGroup:db2_db2inst1_ti2022-l5_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_ti2022-l5_0-rs
                '- Online IBM.Application:db2_db2inst1_ti2022-l5_0-rs:ti2022-l5
Online IBM.ResourceGroup:db2_db2inst1_ti2022-l6_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_ti2022-l6_0-rs
                '- Online IBM.Application:db2_db2inst1_ti2022-l6_0-rs:ti2022-l6
Online IBM.Equivalency:db2_db2inst1_db2inst1_MAXDB75-rg_group-equ
        |- Online IBM.PeerNode:ti2022-l5:ti2022-l5
        '- Online IBM.PeerNode:ti2022-l6:ti2022-l6
Online IBM.Equivalency:db2_db2inst1_ti2022-l5_0-rg_group-equ
        '- Online IBM.PeerNode:ti2022-l5:ti2022-l5
Online IBM.Equivalency:db2_db2inst1_ti2022-l6_0-rg_group-equ
        '- Online IBM.PeerNode:ti2022-l6:ti2022-l6
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:ti2022-l5
        '- Online IBM.NetworkInterface:eth0:ti2022-l6
ti2022-l5:~ #
```

Figure 3-50 lssam output during the DB2 server process failure

Graceful transfer to secondary server

This scenario was tested by manually transferring the resources to the secondary server. In case of a planned change the application resources can be transferred to the secondary server while the primary server undergoes any maintenance change. The following steps were executed:

1. Run **`lssam`** as root from either the primary or the secondary database server. The output should indicate normal operation.
2. Log on to the IBM SmartCloud Control Desk application and navigate to one of the application panels.
3. Issue the **`rgreq -o move db2_db2inst1_db2inst1_MAXDB75-rg`** command to move the resources over to the secondary server.
4. All the DB2 resources are transferred to the secondary node. The DB2 application or the server can now be taken down for maintenance or changes.

Symptoms of failover

When a database failover occurs, the IBM SmartCloud Control Desk application will appear to hang until the database failover sequence is complete. When service is restored, the user interface may show a brief database error: `The database connection failed and the record was not retrieved. Try the operation again. If you experience repeated failures, check the log files in the home directory or contact your system administrator.`

Sometimes the user may receive a blank panel when using the application during failover. Refreshing the browser page often corrects this problem. If the browser

session cannot be recovered, the user may need to navigate back to the login page and re-authenticate.

DB2 shared disk failover testing

The following failover scenarios should be tested with the DB2 shared disk setup.

System failure

This scenario can be tested by powering down the primary database server. The entire workload should be transferred to the secondary database server by the cluster manager.

1. Run **lssam** as root from either the primary or the secondary database server. Figure 3-51 shows the output of the **lssam** command in the normal operating environment.

```
ti2022-i7:~ # lssam
Online IBM.ResourceGroup:db2_db2inst1_0-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_0-rs
                |- Online IBM.Application:db2_db2inst1_0-rs:ti2022-i7
                '- Offline IBM.Application:db2_db2inst1_0-rs:ti2022-l8
        |- Online IBM.Application:db2mnt-sharedhome-rs
                |- Online IBM.Application:db2mnt-sharedhome-rs:ti2022-i7
                '- Offline IBM.Application:db2mnt-sharedhome-rs:ti2022-l8
        '- Online IBM.ServiceIP:db2ip_9_12_4_167-rs
                |- Online IBM.ServiceIP:db2ip_9_12_4_167-rs:ti2022-i7
                '- Offline IBM.ServiceIP:db2ip_9_12_4_167-rs:ti2022-l8
Online IBM.Equivalency:db2_db2inst1_0-rg_group-equ
        |- Online IBM.PeerNode:ti2022-i7:ti2022-i7
        '- Online IBM.PeerNode:ti2022-l8:ti2022-l8
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:ti2022-i7
        '- Online IBM.NetworkInterface:eth0:ti2022-l8
```

Figure 3-51 lssam output for normal operating environment

2. Log on the IBM SmartCloud Control Desk application and navigate to one of the application panels.
3. Shut down the primary database server. Run **lssam** on the secondary server to see the behavior of the system. Figure 3-52 on page 147 shows the **lssam** command output in case of a server failure.

```
ti2022-l8:~ # lssam
Pending online IBM.ResourceGroup:db2_db2inst1_0-rg Control=MemberInProblemState Nominal=Online
        |- Pending online IBM.Application:db2_db2inst1_0-rs Control=MemberInProblemState
                |- Failed offline IBM.Application:db2_db2inst1_0-rs:ti2022-i7 Node=Offline
                '- Pending online IBM.Application:db2_db2inst1_0-rs:ti2022-l8
        |- Online IBM.Application:db2mnt-sharedhome-rs Control=MemberInProblemState
                |- Failed offline IBM.Application:db2mnt-sharedhome-rs:ti2022-i7 Node=Offline
                '- Online IBM.Application:db2mnt-sharedhome-rs:ti2022-l8
        '- Online IBM.ServiceIP:db2ip_9_12_4_167-rs Control=MemberInProblemState
                |- Failed offline IBM.ServiceIP:db2ip_9_12_4_167-rs:ti2022-i7 Node=Offline
                '- Online IBM.ServiceIP:db2ip_9_12_4_167-rs:ti2022-l8
Online IBM.Equivalency:db2_db2inst1_0-rg_group-equ
        |- Offline IBM.PeerNode:ti2022-i7:ti2022-i7 Node=Offline
        '- Online IBM.PeerNode:ti2022-l8:ti2022-l8
Online IBM.Equivalency:db2_public_network_0
        |- Offline IBM.NetworkInterface:eth0:ti2022-i7 Node=Offline
        '- Online IBM.NetworkInterface:eth0:ti2022-l8
```

Figure 3-52 lssam output in case of a server failure

4. All the resources are now transferred to the secondary server. When the primary server comes back up, the old primary server will be added back to the cluster manager and monitored.

Process failure

This scenario simulates the DB2 server process failure. The cluster manager detects that the DB2 server process is down and restarts the process.

1. Run **lssam** as root from either the primary or the secondary database server. The output should indicate normal operation.
2. Log on to the IBM SmartCloud Control Desk application and navigate to one of the application panels.
3. Issue the **db2_kill** command to abruptly end all the DB2 server processes. Run **lssam** as root user to list the status of the cluster. Figure 3-53 displays the **lssam** output in case of DB2 process failure in the shared disk setup.

```
ti2022-i7:~ # lssam
Pending online IBM.ResourceGroup:db2_db2inst1_0-rg Nominal=Online
        |- Pending online IBM.Application:db2_db2inst1_0-rs
                |- Pending online IBM.Application:db2_db2inst1_0-rs:ti2022-i7
                '- Offline IBM.Application:db2_db2inst1_0-rs:ti2022-l8
        |- Online IBM.Application:db2mnt-sharedhome-rs
                |- Online IBM.Application:db2mnt-sharedhome-rs:ti2022-i7
                '- Offline IBM.Application:db2mnt-sharedhome-rs:ti2022-l8
        '- Online IBM.ServiceIP:db2ip_9_12_4_167-rs
                |- Online IBM.ServiceIP:db2ip_9_12_4_167-rs:ti2022-i7
                '- Offline IBM.ServiceIP:db2ip_9_12_4_167-rs:ti2022-l8
Online IBM.Equivalency:db2_db2inst1_0-rg_group-equ
        |- Online IBM.PeerNode:ti2022-i7:ti2022-i7
        '- Online IBM.PeerNode:ti2022-l8:ti2022-l8
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:ti2022-i7
        '- Online IBM.NetworkInterface:eth0:ti2022-l8
```

Figure 3-53 lssam output during DB2 process failure in the shared disk setup

4. The DB2 process should restart and the application should continue processing as normal.

Graceful failover

This scenario can be tested by manually transferring the resources to the secondary server. In case of a planned change the application resources can be transferred to the secondary server while the primary server undergoes any maintenance change.

1. Run **`lssam`** as `root` from either the primary or the secondary database server. The output should indicate normal operation.
2. Log on to the IBM SmartCloud Control Desk application and navigate to one of the application panels.
3. Issue the **`rgreq -o move db2_db2inst1_db2inst1_0-rg`** command to move the resources over to the secondary server.
4. All the DB2 resources are transferred to the secondary node. The DB2 application or the server can now be taken down for maintenance or changes.

Symptoms of failure

When a database failover occurs, the IBM SmartCloud Control Desk application will appear to hang until the database failover sequence is complete. When service is restored, the user interface may show a brief database error: `The database connection failed and the record was not retrieved. Try the operation again. If you experience repeated failures, check the log files in the home directory or contact your system administrator.`

Sometimes you may receive a blank panel when using the application during failover. Refreshing the browser page often corrects this problem. If the browser session cannot be recovered, you may need to navigate back to the login page and re-authenticate.

3.10 Conclusion

This chapter gave an overview and configuration examples for local high availability. It described how to eliminate single points of failure in an IBM SmartCloud Control Desk environment.

4

Implementing a passive disaster recovery site

In this chapter we provide information about configuring a secondary site for disaster recovery with IBM SmartCloud Control Desk. This is considered an active-passive configuration.

- Introduction
- Disaster recovery plan
- Prerequisites
- Storage replication
- Web server and load balancer
- Application server
- Integration framework
- Database
- IBM SmartCloud Control Desk configuration
- Failover scenarios and testing
- Symptoms of failover

4.1 Introduction

In many organizations, maintaining system availability is critical for business operations. Localized high availability can help with process and hardware failure but what happens when there is an unpredictable event that can compromise the entire data center? For critical systems a disaster recovery plan should be in place. Disaster recovery implies complete site replication to an alternate location so that services can be restored when the primary site goes down. This can be thought of as a type of *insurance policy* for the IBM SmartCloud Control Desk infrastructure.

An active-passive site configuration can provide a company with a contingency plan when an unexpected failure occurs. File system, database, backup/restore procedures can all be implemented to keep the passive site synchronized with the primary. The technologies used depend on the distance, budget, and synchronization state required by the organization. Having a reliable, high-speed network infrastructure and link between the sites is one of the most important elements in the plan. Figure 4-1 on page 151 shows an example high level topology for active-passive disaster recovery using file-based and database replication techniques.

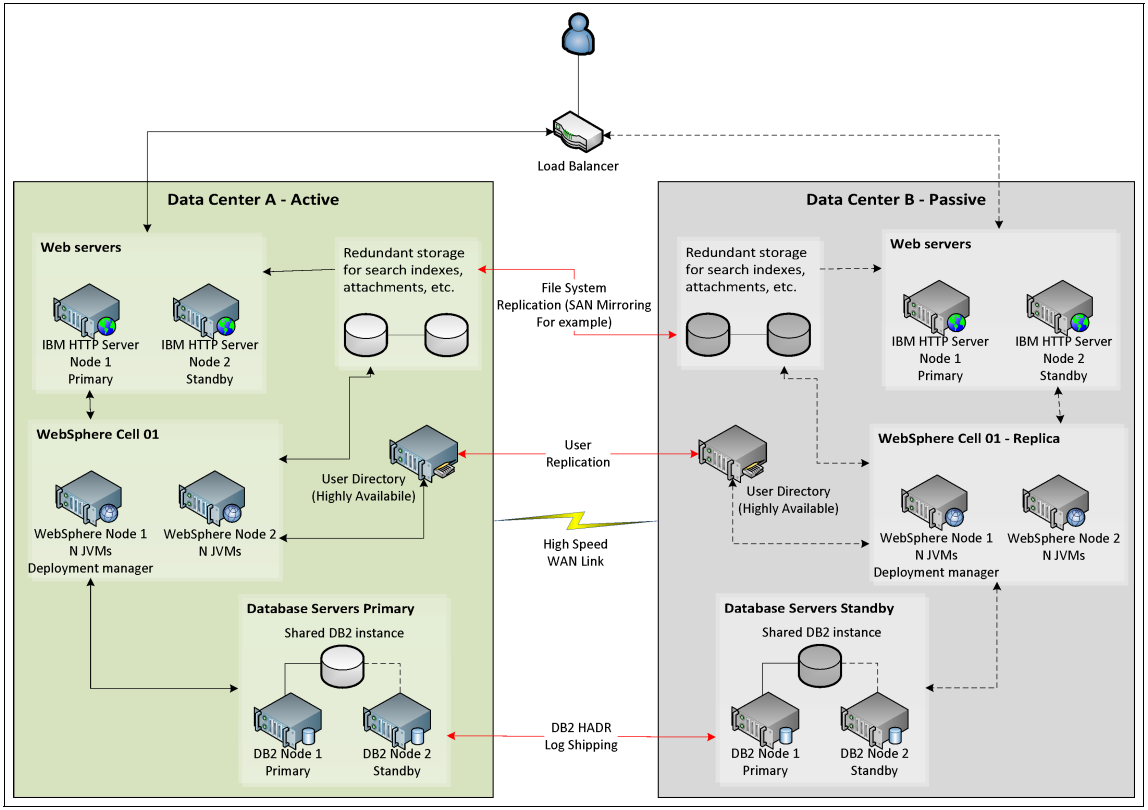

Figure 4-1 Active-passive disaster recovery topology

Two important considerations when designing a disaster recovery topology are the *Recovery Time Objective (RTO)* and *Recovery Point Objective (RPO)*. RTO is the time required to restore the environment on an alternate site when the primary goes down. RPO is the amount of time between replications that the company can afford. If an RPO is 15 minutes for example, it means that the environments are not in a synchronous state and could potentially have up to 15 minutes of data loss.

When selecting a second site for the disaster recovery data center, RTO and RPO are major considerations. If there were an environmental disaster, you would want to make sure your second site was far enough away that it would not be affected. The distance, however, will affect the synchronization state of the sites and could impact the RPO times. Two sites that are very close together could potentially maintain a near synchronous state with a low RPO (assuming a very fast WAN link). This can also be a dangerous situation because it creates a higher probability that both sites will be impacted by a disaster. Spreading the

sites further apart will widen the synchronization gap and increase RPO but decreases the chances of both sites being affected.

Performance and scalability are also important considerations when planning your sites. The passive secondary site should be able to maintain user workload if it were to become the active site. If one site grows to accommodate more users, the disaster recovery site will need to grow as well to ensure performance and system stability is maintained upon failover. Sometimes, organizations decide that a scaled down secondary site is acceptable in a disaster scenario but it is crucial that there are precautions taken to make sure this site does not crash due to excessive load.

4.2 Disaster recovery plan

When implementing a passive site for disaster recovery it is important to understand the types of disaster scenarios your organization could potentially face. In order to successfully and efficiently execute a complete site failover, a well designed and tested disaster recovery plan should be in place.

4.2.1 Failover overview

There are several different scenarios that may require a site switch.

- Testing

  After implementing a second site for disaster recovery it is important to test the site failover to ensure service can be restored properly. The testing should also occur at designated intervals to be sure the environment is ready in case of an actual disaster. Executing this failover procedure will also act as training for the administrators to be ready for a disaster. Documenting the lessons learned after failover tests will help evolve the overall disaster plan.

- Planned takeover

  Sometimes a planned site takeover might be necessary and not just for testing purposes. Situations that may require a planned takeover include:

  - Site maintenance

    There may be situations where an entire site or several essential components of a site need to come down for maintenance. Middleware fixpacks, hardware upgrades and networking changes are a few examples of such scenarios. If this maintenance will affect the availability of the IBM SmartCloud Control Desk application for a long duration, it may be desirable to switch services to your secondary site. Although the site switch will take some time to complete, the downtime may be less than the

duration of the maintenance. This also gives administrators a chance to ensure the disaster recovery plan still works as designed.

Important: Although maintenance and fixpacks are mentioned, this does not include maintenance and upgrades to the IBM SmartCloud Control Desk application itself

- Possibility of a disaster

  Many disaster situations happen unexpectedly. Massive power outages, earthquakes and hardware failure are examples of disasters that may happen without any warning. Other types, such as hurricanes and other weather-related disasters, you often have warning of. When there is the possibility of these types of disasters it may be a good idea to failover to another site that has less of a chance of being affected. For instance, if a hurricane is heading your way, and you have a site far enough away, switching services to this site should be considered.

- Disaster

  This is the main reason for the disaster recovery topology. Unpredictable events can affect your primary site and completely bring down the service. Weather, human error, hardware failures, malicious users are some of the many types of problems that can bring down a site. When a disaster occurs, it is time to execute the disaster recovery plan. Restoring services as quickly as possible on the backup site will cause minimal impact on operations.

4.2.2 Designing a disaster recovery plan

One of the most crucial components of an efficient recovery is a well-designed disaster recovery plan. Your company has invested in another site with all the resources required to take over operations, but without a procedure in place this takeover can take too long or fail completely. Disaster recovery plans should include:

- Names and contact information of all parties involved with the failover procedure. It is a good idea to have backups for everyone in case someone cannot be contacted at that time.
- A detailed step-by-step outline on the order of operations for takeover. For example, the database should be online with a primary role assigned before application servers are started.
- System information required for administrators to restore services.
- Test cases so administrators can verify that the system is functioning properly before allowing users to reconnect.

These were just a few of the common requirements in a disaster recovery plan. Organizations should define a plan that is tailored to their specific needs. Detailed documentation and thorough testing can help with creating a solid plan. Communication of this plan to all administrators is extremely important. Many disaster recovery procedure failures are attributed to a lack of internal planning and coordination amongst all system administrators.

4.3 Prerequisites

There are many prerequisites to implementing a disaster recovery topology. Some of these topics are covered in this IBM Redbooks publication and others are assumptions. It is best to research which solutions are best for your organizational needs.

- Local high availability

  Most disaster recovery topologies are supplemental to a local high availability solution. Often, process and hardware failures can be corrected quickly without the need for failover to a second site. Completely switching sites will have an impact to users that cannot be masked like the high availability solution. Please refer to Chapter 3, “Local high availability topology” on page 25 for more information.

- Load balancer

  A load balancer can be used in front of the web servers to provide balancing across several web servers and also provide a transparent access point for users when a site failover occurs. There are hardware and software solutions for load balancers but care should be taken to ensure this is not a single point of failure. Having a load balancer that can detect when one site is offline can help ease the transition in a disaster scenario. Most load balancing solutions have high availability and disaster recovery options.

- Appropriate licenses

  When implementing an additional site into your environment it is important to check your license agreements to make sure this is covered. Please check with your IBM sales associate to review the license agreements for your organizations. Additional licensing may be required for the new site.

- Networking

  The network link between the two sites is critical for synchronization of the application and data. Network administrators should be involved in the planning process and the network may need to be upgraded when connecting a second site. Redundant network links could help avoid synchronization loss when a single link fails.

Other networking considerations such as DNS and routing can help reduce the recovery time during a failover. For instance, if both sites have similar networking and IP addresses can be rerouted to the second site, this can help eliminate the need for reconfiguration when failover occurs. Full application replication using technologies such as SAN Global Mirroring, will synchronize the hostnames and IP addresses as well as the application configuration. The ability to manipulate the networking and hostname resolution without having to reconfigure the whole application can speed up the process.

- Storage replication

  When implementing a second site there are files that will get stored on the primary and should be replicated to the passive site. Attached documents, global search indexes, integration framework files are examples of such files. If the primary site fails, these files are required on the standby site to continue with full application functionality. An example storage solution is disk-based mirroring as described in 4.4, “Storage replication” on page 156.

  Some organizations choose to replicate the storage for the entire IBM SmartCloud Control Desk environment from the active site to the passive site instead of using the middleware replication mechanisms. This includes:

  - Web server installation and configuration files
  - Application server installation, configuration files, and profiles
  - Database installation and database files
  - Tivoli Process Automation Engine application installation files
  - Any other middleware files used in the topology

  This would allow for the second site to be an exact duplicate of the primary. It is important that a robust storage mirroring solution be in place for this to work effectively. Asynchronous versus synchronous mirroring can affect the RPO of the site failover and should be considered. The distance between sites can affect the synchronization ability of the storage solution.

- A second site

  An obvious prerequisite to a disaster recovery plan is a second site that can take over from the primary when a disaster occurs. Careful consideration should be taken when selecting a location. Sites too close could both be affected in a disaster, but sites too far will not be able to synchronize as quickly and could potentially have data loss.

- Administrators with necessary skills

  When implementing disaster recovery technologies such as database and file system replication, the complexity of the IBM SmartCloud Control Desk topology increases. Administrators who are familiar with these technologies and posses the skills required to configure, maintain and test the infrastructure are critical. Lack of coordination amongst the team can lead to a failed disaster recovery.

- Application installation files on both sites

  It is important that the application installation directory for IBM SmartCloud Control Desk are copied to the secondary site. If the primary site fails and application reconfiguration is required, these files will need to be accessible from the secondary. These directories should also be kept synchronized with each other after any changes. Frequent backups of the application installation directories is advised.

4.4 Storage replication

This section describes the active-passive disaster recovery for IBM SmartCloud Control Desk using the storage or disk mirroring techniques. There are various disk storage systems like IBM Storage Area Network (SAN) or Veritas storage foundation can be used to design a disaster recovery solution.

In a typical disaster recovery topology using mirroring, both the sites are equipped with the exactly same hardware configuration. The data is replicated or mirrored using replication techniques like flashcopy, synchronous mirror or asynchronous mirror. By using these mirroring techniques, a storage disk cluster is setup so that an update performed on the primary site gets mirrored on the secondary site. The storage volumes can be on the remote locations with high speed WAN network link.

IBM SmartCloud Control Desk comprises of at least three middleware components, the web server, application server and database server. The user directory (Tivoli Directory Server or Microsoft Active Directory) and WebSphere MQ are optional components but should be considered as well. It is important to ensure that the mirroring for all components occurs in a synchronized manner. For example, the database update for the attachment should be mirrored at the same time as the creation of the attachment document on the application server. According to best practices, a logical consistency group should be created which comprises of all the middleware components and ensure that these consistency groups are in a synchronized state with the standby groups.

There are two modes for disk mirroring. Depending on your applications need, distance and tolerance for data loss, one of the two modes can be selected.

- Synchronous mirroring

  In synchronous mirroring mode the application write would be committed on the secondary site before the next write operation is permitted. This can affect the performance over the WAN. The distance between two sites can impact the performance of the writes.

- Asynchronous mirroring

In asynchronous mirroring mode the application writes can be configured to be written to the secondary site on the predefined interval. In this mode, all the writes are stacked before they get written to the secondary site. This provides better performance over the WAN. There is potential loss of data if the primary site or the disks go offline before the writes completed on the secondary site.

A mirror relationship is established between the two storage sites. The primary role of this relationship allows for read/write access to the disk drives on the primary site. The secondary role of the mirror relationship prohibits write access to the secondary drive from any other host except the owning controllers. This ensures that the data is in a synchronized state between the two sites. The data is initially copied from the primary site to the secondary site. This process requires an outage window. After the full synchronization is completed, the updates are logged on the mirrored drive and replicated to the secondary site. If the network communication was interrupted between the two sites, then mirror is suspended until the communication is restored and all the updates are transferred to the secondary site.

SAN: For more information about SAN mirroring features, refer to the following IBM Redbooks publications:

- *IBM System Storage DS8000 Copy Services for Open Systems*, SG24-6788
- *IBM XIV Storage System: Copy Services and Migration*, SG24-7759
- *IBM System Storage DS Storage Manager Copy Services Guide*, SG24-7822
- *SAN Volume Controller and Storwize V7000 Replication Family Services*, SG24-7574

In an environment where there is little or no tolerance for data loss, it is important to have frequent backups in addition to designing a disaster recovery location. Various backup techniques such as Flashcopy or backup to the tape management services can be implemented to ensure that the data can be recovered in case the failover to the disaster recovery site fails.

4.5 Web server and load balancer

The web server is the user access point for the IBM SmartCloud Control Desk environment. Users will connect to the web server and the web server will balance the requests across the application servers. If an environment has more than one web server, a load balancer can be added to act as transparent access

point for user traffic. Intelligent load balancers can be used to detect the status of a site or if particular web servers are offline to reroute the users to another location.

In an active-passive disaster recovery topology, the load balancer is most needed when there are multiple web servers. Having only one load balancer on one of the sites can create a single point of failure so it is important to have a load balancer on the disaster recovery site that can take over.

4.5.1 IBM HTTP Server

This configuration and topology example uses IBM HTTP Server as the web server solution for IBM SmartCloud Control Desk. When implementing a passive site for disaster recovery, IBM HTTP Servers should be replicated to the second site. Configuration of the IBM HTTP Servers can be synchronized using backup and restore techniques, SAN mirroring or by simply installing the IBM HTTP Servers in an identical manner on the second site. On a passive disaster recovery site the IBM HTTP Server should be offline so users cannot attempt to access the server.

Installation and configuration

The installation and configuration of IBM HTTP Server should include the Plug-in for WebSphere Application Server. High availability of IBM HTTP Server should also be considered on the second site. If there is an extended outage on the primary site, having IBM HTTP Server as a single point of failure on the backup site may be too risky. The backup site should be configured in a similar manner as defined in 3.4.1, “IBM HTTP Server” on page 32.

Passive site considerations

The IBM HTTP Server configuration should be synchronized with the primary site. If the servers are reconfigured or changed on the primary site, these changes should be replicated to the secondary site. Minor changes might be replicated manually to the second site by system administrators, but they must use caution to ensure there is no error with the configuration that may increase the recovery time when a site switch is required. Using file-based replication techniques from one site to the other can be used to keep the configurations synchronized.

IBM HTTP Server can be used for other IBM SmartCloud Control Desk capabilities, such as serving attachments for the *attached documents* functionality within many of the applications. These attachments are stored on a file system and should be replicated to the second site if they are considered critical. If you are using file-based replication for the HTTP server configuration, these attached documents should be stored on the same paths on the backup

site so configuration changes on the IBM HTTP Server are not required. Even when manually configuring the IBM HTTP Server on the second site, it is a good practice to replicate the attached documents to the same location to keep the configuration consistent.

4.5.2 Load balancer

Many types of load balancers are available that can be used with your IBM SmartCloud Control Desk topology. Hardware and software-based load balancers can help distribute the load across several web servers as a *reverse proxy* or directly to the application servers, if desired.

A couple of common solutions for load balancing include:

- IBM WebSphere Edge Components for software-based

  More information can be found at:

  `http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=%2Fcom.ibm.websphere.edge.doc%2Flb%2Fwelcome_edge.html`

- F5 BIG-IP for hardware-based

  More information can be found at:

  `http://www-304.ibm.com/software/brandcatalog/ismlibrary/marketplace/details?catalog.label=1TW10MA4B`

Passive site considerations

Some organizations use a load balancer to provide a single transparent access point to users. Load balancers can be used to detect the status of web servers or application servers and can contain logic to reroute to other servers if they detect a failure. The load balancer should not route user requests to the passive site when the primary is active.

It is important to note that IBM SmartCloud Control Desk does not support session failover. If an application server fails and users are rerouted to another application server JVM, they will need to log back in to a new session. Load balancers must also be configured for *sticky* sessions. This means, when the load balancer attaches a user to a specific server, this user must remain attached to the same server throughout the duration of the session. This is referred to as *session affinity*.

Additional information: For more detailed configuration information, consult the manufacturer documentation for your load balancer solution

4.6 Application server

The application server is responsible for handling business logic and processing load demanded by the end users. Therefore, it is very important that the secondary site have capacity to handle the same amount of load as the primary site.

4.6.1 WebSphere Application Server

The configuration for the WebSphere Application Server on the secondary site will be the same as in the primary site. The installation method can be the same as depicted in “Installing WebSphere Application Server” on page 43 or using a backup and restore configuration followed by hostname updates. In a WebSphere Application Server multisite topology, it is not advisable to spread a single cell across two sites. For this reason, each site should have its own distinct cell. The backup and restore functionality will replicate the cell to the second site creating a new independent cell.

The backup and restore configuration is recommended because it recreates the exact same primary environment on the secondary site.

Important: If the SIB is being used for integration, the backup and restore configuration method must be used to reflect the same UUIDs for messaging engines and destinations on both environments

For this IBM Redbooks publications it is assumed the following variables shown in Table 4-1. These values are not mandatory for all installations and might vary in other environments.

Table 4-1 Variables

Name	Description	Value
WAS_HOME	WebSphere Application Server installation path	`/opt/IBM/WebSphere/AppServer`
WAS_DMGR_PATH	Deployment manager profile installation path	`/opt/was_dmgr_files/profiles/Dmgr01`
primary_washost1	WebSphere Application Server primary site node 1 hostname	`ti2022-l3.itso.ibm.com`
primary_washost2	WebSphere Application Server primary site node 2 hostname	`ti2022-l4.itso.ibm.com`

Name	Description	Value
secondary_washost1	WebSphere Application Server secondary site node 1 hostname	ti-2021-2.itso.ral.ibm.com
secondary_washost2	WebSphere Application Server secondary site node 2 hostname	ti-2021-7.itso.ral.ibm.com
primary_ihshost1	IBM HTTP Server primary site node 1 hostname	ti2022-l1.itso.ibm.com
primary_ihshost2	IBM HTTP Server primary site node 2 hostname	ti2022-l2.itso.ibm.com
secondary_ihshost1	IBM HTTP Server secondary site node 1 hostname	ti-2021-1.itso.ral.ibm.com
secondary_ihshost2	IBM HTTP Server secondary site node 2 hostname	ti-2021-6.itso.ral.ibm.com

Backup and restore configuration

This section describes the method to back up and restore the configuration from the primary site to a secondary site using WebSphere Application Server tools.

Before starting, make sure all assumptions and prerequisites from “Installing WebSphere Application Server” on page 43 are being met.

1. Login as WebSphere Application Server installation user on `primary_washost1`.
2. Back up the current deployment manager and node profiles using the **`manageprofiles`** command shown in Example 4-1.

Example 4-1 Backup profiles

```
ti2022-l3:WAS_HOME/bin # ./manageprofiles.sh \
> -backupProfile \
> -profileName Dmgr01 \
> -backupFile primary_washost1_Dmgr01_bkp.zip
INSTCONFSUCCESS: Success: The profile backup operation was successful.
ti2022-l3:WAS_HOME/bin # ./manageprofiles.sh \
> -backupProfile \
> -profileName AppSrv01 \
> -backupFile primary_washost1_AppSrv01_bkp.zip
INSTCONFSUCCESS: Success: The profile backup operation was successful.
```

3. Transfer the two backup files to server `secondary_washost1`.
4. Log in as WebSphere Application Server installation user on `secondary_washost1`.

5. Restore the backups taken using the **manageprofiles** command shown in Example 4-2.

Example 4-2 Restore profiles

```
ti-2021-2:WAS_HOME/bin # ./manageprofiles.sh \
> -restoreProfile \
> -backupFile primary_washost1_Dmgr01_bkp.zip
INSTCONFSUCCESS: Success: The profile was successfully restored.
ti-2021-2:WAS_HOME/bin # ./manageprofiles.sh \
> -restoreProfile \
> -backupFile primary_washost1_AppSrv01_bkp.zip
INSTCONFSUCCESS: Success: The profile was successfully restored.
```

6. Log in as WebSphere Application Server installation user on `primary_washost2`.
7. Back up the current node profile using the **manageprofiles** command shown in Example 4-3.

Example 4-3 Backup profile

```
ti2022-l4:WAS_HOME/bin # ./manageprofiles.sh \
> -backupProfile \
> -profileName AppSrv01 \
> -backupFile primary_washost1_AppSrv01_bkp.zip
INSTCONFSUCCESS: Success: The profile backup operation was
successful.
```

8. Transfer the backup file to server `secondary_washost2`.
9. Log in as WebSphere Application Server installation user on `secondary_washost2`.
10. Restore the backups taken using the **manageprofiles** command shown in Example 4-4.

Example 4-4 Restore profile

```
ti-2021-7:WAS_HOME/bin # ./manageprofiles.sh \
> -restoreProfile \
> -backupFile primary_washost2_AppSrv01_bkp.zip
INSTCONFSUCCESS: Success: The profile was successfully restored.
```

11. Log in as WebSphere Application Server installation user on `secondary_washost1`.
12. Update hostnames using the **wsadmin** tool shown in Example 4-5.

Example 4-5 Hostname updates

```
ti-2021-2:WAS_HOME/bin # ./wsadmin.sh -lang jython -conntype NONE
WASX7357I: By request, this scripting client is not connected to any server process.
Certain configuration and application operations will be available in local mode.
WASX7031I: For help, enter: "print Help.help()"
wsadmin>AdminTask.changeHostName('[-nodeName primary_washost1CellManager01 -hostName
secondary_washost1 ]')
''
wsadmin>AdminTask.changeHostName('[-nodeName primary_washost1Node01 -hostName
secondary_washost1 ]')
''
wsadmin>AdminTask.changeHostName('[-nodeName primary_washost2Node01 -hostName
secondary_washost2 ]')
''
wsadmin>AdminTask.changeHostName('[-nodeName primary_ihshost1-node -hostName
secondary_ihshost1 ]')
''
wsadmin>AdminTask.changeHostName('[-nodeName primary_ihshost2-node -hostName
secondary_ihshost2 ]')
''
wsadmin>AdminConfig.save()
''
wsadmin>exit
```

13. Synchronize all nodes using the **syncNode** command.
14. Configure System Automation for Multiplatforms for the secondary site deployment manager as outlined in “Automating deployment manager failover with SA MP” on page 44.
15. Configure System Automation for Multiplatforms for the secondary site nodeagents as outlined in “Automating nodeagent restart with SA MP” on page 52.

Storage mirroring

This section describes an optional topology that utilizes a disk replication system for IBM WebSphere Application Server. There are three types of data that are important to be captured and replicated to the secondary site:

- Installation data associated with the WebSphere product.
- Configuration data associated with the application and the resources needed to run them.
- Run data associated with the specific instance of process and business data.

The data can be divided into three independent but related consistency groups or a single consistency group. In some cases the consistency group should be expanded to include data from the DB2 server. The rationale for including the data for all the servers in a single consistency group is that it is required by all the data to be consistent. In some cases, as the number of nodes grows, it may be important to limit the number of consistency groups.

The rationale for dividing the data into these consistency groups is that the actions that make the data inconsistent are different for each group:

- The install data for each of the servers is included in the same consistency group.
- The configuration data for each of the profiles in the cell is included in the same consistency group.
- The run data for each of the profiles in the environment is included in the same consistency group.

It is important to ensure that the write order is preserved on both sites to maintain the consistency of the data. For more information about the disaster recovery setup of the WebSphere Application Server using disk mirroring refer to:

http://www.ibm.com/developerworks/websphere/library/techarticles/0809_redlin/0809_redlin.html

4.7 Integration framework

The integration framework must be able to recover all transactions without loss when a failure occurs. After configuring the WebSphere Application Server, either Service Integration Bus (SIB) or WebSphere MQ Server can be chosen as the JMS provider.

This section describes the necessary configuration for each of these JMS providers.

4.7.1 Service integration bus configuration

When using the SIB as the JMS provider (Figure 4-2 on page 165), WebSphere Application Server must be installed on the secondary site using the steps described in “Backup and restore configuration” on page 161. If this method is not used, the messaging engine will not be able to recover due to its different UUID.

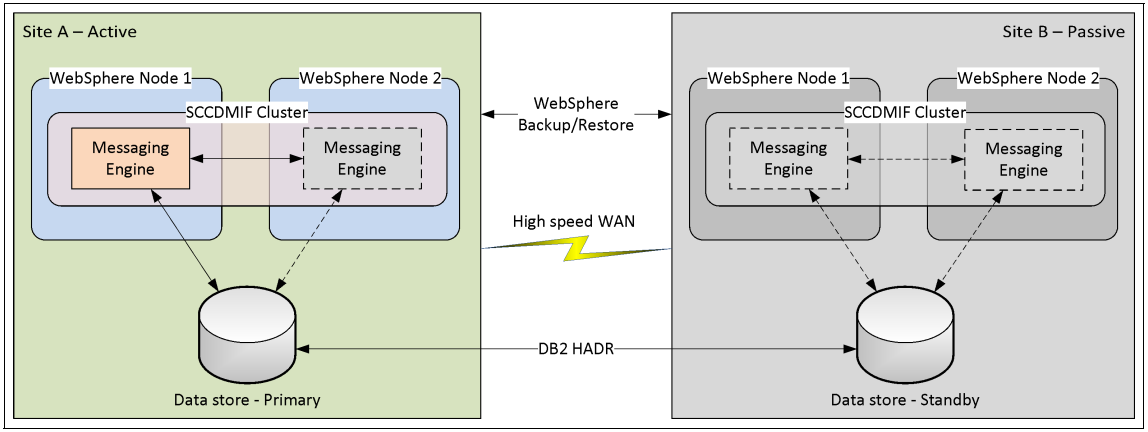

Figure 4-2 Active-passive SIB configuration

To configure the SIB, follow the steps outlined in “Service integration bus configuration” on page 108 using secondary site hosts and IP addresses. On step 24 on page 113, set the *secondary site* database hostname/IP address.

The datastore used by SIB will be replicated to the passive site using database replication techniques as described in 4.8.1, “Database recovery techniques” on page 167.

4.7.2 WebSphere MQ configuration

WebSphere MQ can be configured in an active-passive configuration as the JMS provider. It is advisable to configure MQ for local high availability as well as for disaster recovery. Figure 4-3 on page 166 shows an example multi-instance configuration for MQ with high availability and disaster recovery in an active-passive topology.

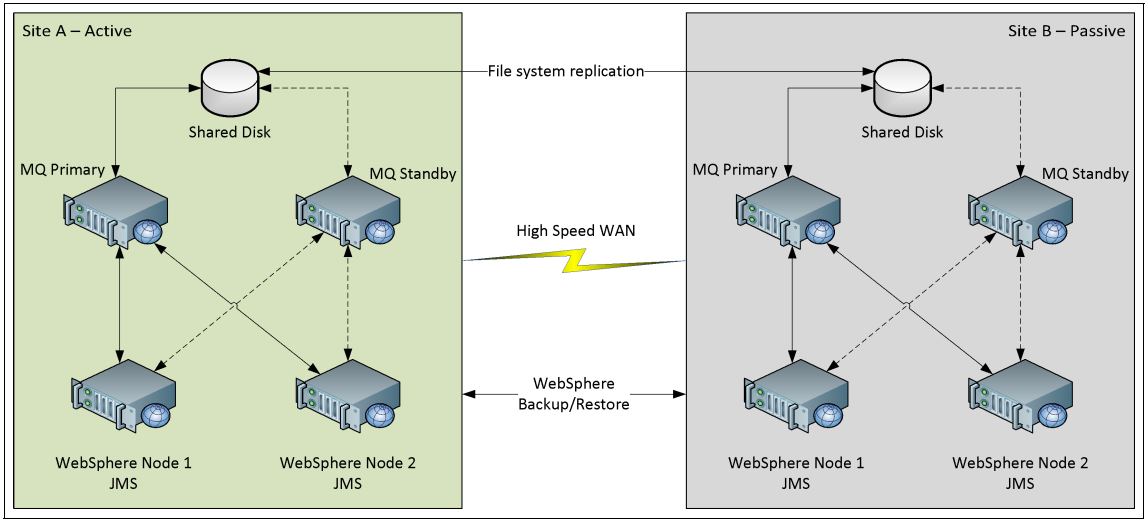

Figure 4-3 WebSphere MQ example configuration

When using MQ as the JMS provider, WebSphere MQ Server must be configured following the steps outlined in "WebSphere MQ configuration" on page 123. Be aware to use secondary site hostnames and IP addresses where they apply, especially in the following steps:

- Steps 20 on page 129 and 47 on page 131 - use the WebSphere MQ secondary site primary server's hostname.
- Step 29 on page 130 and 56 on page 131 - use the WebSphere MQ secondary site primary and secondary server's hostname.

4.8 Database

Disaster recovery for an enterprise application means that all critical business operations are recovered in case of any disaster or site wide outage. Some organizations have little or no tolerance for data loss, in which case the disaster recovery solution needs to be deployed to restore data to the applications rapidly. The solution must ensure the consistency of the data, allowing for restoration of the systems and applications reliably and fast.

The database is one of the most critical components of the IBM SmartCloud Control Desk application. It provides the option of using IBM DB2, Oracle or Microsoft SQL Server for the deployment. The middleware installer program provides the option of installing a new instance of DB2 without high availability, or

using a preexisting instance of the DB2 database. If you choose Oracle or Microsoft SQL Server, then you must install and configure them manually.

Bringing down the database will disrupt the IBM SmartCloud Control Desk function. There are various disaster recovery database configurations available.

This chapter describes the passive disaster recovery setup for IBM SmartCloud Control Desk using DB2 and Oracle databases. We describe some of the replication features of DB2 and Oracle and how to maintain database consistency using the disk mirroring techniques.

4.8.1 Database recovery techniques

Backup and recovery are the fundamental technologies of a disaster recovery solution. The database backup image can be stored on a tape or disk, transported to the secondary site and used to rebuild the database. This process can take a long time with a potential for loss of data. To manage the recovery time objective (RTO) and recovery point objective (RPO), the following solutions can be implemented to keep the database synchronized across the sites. In case of disaster, the application can be failed over to the secondary database. This section describes the following IBM SmartCloud Control Desk setup using database recovery techniques:

- DB2 HADR for disaster recovery
- Storage mirroring
- Oracle Active Data Guard

DB2 HADR for disaster recovery

For organizations who have little or zero tolerance for data loss, DB2 High Availability and Disaster Recovery allows replication of any logged database activity to a local or remote location. A DB2 HADR primary database uses internal processes to ship database log buffers to an HADR standby database. A process on the standby server then replays the log records directly to the standby database. The secondary server is always in the *rollforward* mode, in the state of near readiness, so the takeover to the standby server is fast. The standby database can be converted to the primary database and accessed by applications and users in the event of a disaster, or if the primary server fails. Figure 4-4 on page 168 displays the DB2 database setup using two remote locations. For more information about the HADR requirements and considerations, refer to “HADR requirements” on page 64.

It is important to understand that with DB2 v9.7.x there can only be one HADR standby database. For this reason, you cannot configure DB2 HADR for local high availability as well as HADR across sites. For local high availability, the DB2

shared disk configuration should be implemented combined with HADR across the sites, as shown in Figure 4-4.

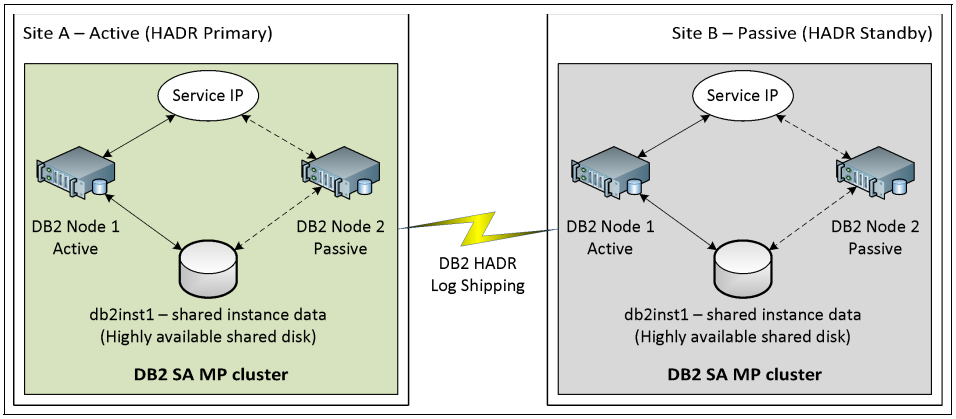

Figure 4-4 DB2 disaster recovery setup across two remote locations

If local high availability on the disaster recovery site is not required, a single node can be used instead of a shared disk cluster on both sites. Figure 4-5 shows an example of a single node disaster recovery standby.

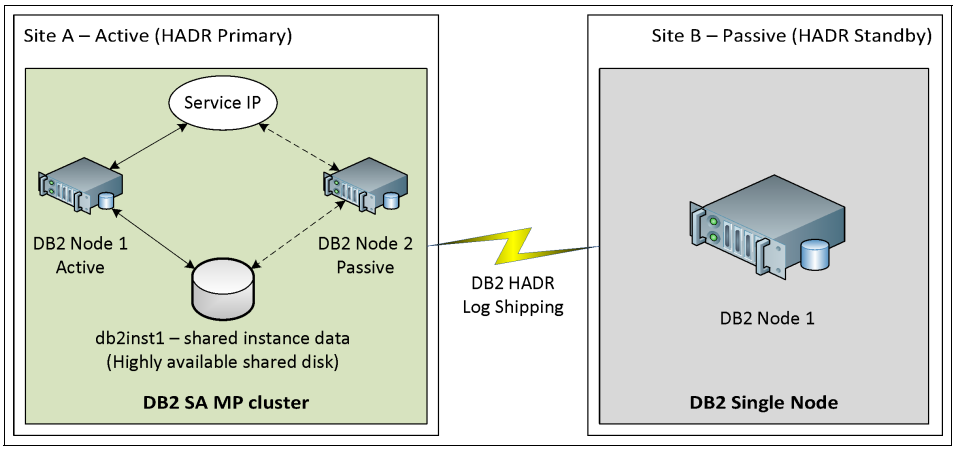

Figure 4-5 HADR with a single node standby on Site B

DB2 HADR configuration

This section describes how to set up HADR for the IBM SmartCloud Control Desk database in an active-passive disaster recovery mode. The setup is described using the command line interface.

The installation path and other variables are listed in Table 4-2.

Table 4-2 DB2 setup with active-passive disaster recovery

Variables	Description	Values
DB2_INSTANCE	DB2 instance name	db2inst1
DB2_DBNAME	DB2 database name	maxdb75
DB2_DR_HOME	DB2 instance home on remote site	/sharedhome/db2inst1
DB2_DR_INSTANCE	DB2 instance name on remote site	db2inst1
DB2_DR_DBNAME	DB2 database name	maxdb75
db2_sd_svcip	DB2 shared disk service IP	9.12.4.167
db2drhost	DB2 secondary site hostname	ti-2021-3.itso.ral.ibm.com

The database on the primary site can be a standalone database with HADR providing data synchronization on the secondary passive site. The following topology describes a DB2 shared disk high availability setup on the primary site with HADR providing data synchronization on the secondary passive site. For more information about the DB2 shared disk high availability setup, refer to “DB2 shared disk high availability setup” on page 79.

To set up HADR for DB2 in the disaster recovery setup using the command line interface, complete the following steps:

1. Set the required database configuration parameters on the *primary* database server.

   If archive logging is not configured already, then update the LOGRETAIN and LOGARCHMETH1 parameters by running the following commands:

   db2 update db cfg for DB2_DBNAME using LOGRETAIN recovery

   db2 update db cfg for DB2_DBNAME using LOGARCHMETH1 LOGRETAIN

   Set the LOGINDEXBUILD parameter so that the index creation and reorganization operations are logged by running the following command.

   db2 update db cfg for DB2_DBNAME using LOGINDEXBUILD ON

2. Back up the database on the primary node by running the following command. The database backup should be an *offline* backup, which means no user connections are allowed on the database.

   db2 backup database DB2_DBNAME to BACKUP_PATH

3. Transfer the backup image to the secondary server.
4. Restore the database on the secondary server by running the following command. The standby database must be in the *Rollforward pending* mode.

   db2 restore database DB2_DBNAME from BACKUP_PATH taken at BACKUP_TIMESTAMP replace history file

   Tip: Check the database Rollforward pending status issuing the db2 get db cfg for DB2_DBNAME |grep “Rollforward pending” command.

5. Update the database configuration parameters on the primary database server.

Example 4-6 DB2 HADR setup in disaster recovery mode on the primary database server

```
db2 “update db cfg for DB2_DBNAME using HADR_LOCAL_HOST db2_sd_svcip”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_HOST db2drhost”
db2 “update db cfg for DB2_DBNAME using HADR_LOCAL_SVC db2hadrlocalsvc”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_SVC db2hadrremotesvc”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_INST DB2_INSTANCE”
db2 “update db cfg for DB2_DBNAME using HADR_TIMEOUT 120”
db2 “update db cfg for DB2_DBNAME using HADR_SYNCMODE SYNC”
db2 “update db cfg for DB2_DBNAME using HADR_PEER_WINDOW 120”
```

6. Run the db2 “get db cfg for DB2_DBNAME” |grep HADR command.

   Example 4-7 lists the db cfg parameter configuration for HADR on the primary database.

Example 4-7 HADR configuration for the primary site

```
db2inst1@ti2022-i7:~> db2 "get db cfg for maxdb75" |grep HADR
 HADR database role                                      = PRIMARY
 HADR local host name                  (HADR_LOCAL_HOST) = 9.12.4.167
 HADR local service name                (HADR_LOCAL_SVC) = 55001
 HADR remote host name                (HADR_REMOTE_HOST) = ti-2021-3
 HADR remote service name              (HADR_REMOTE_SVC) = 55002
 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) = 120
```

7. Update the db configuration parameters on the secondary database server.

Example 4-8 DB2 HADR setup in disaster recovery mode on secondary database server

```
db2 “update db cfg for DB2_DBNAME using HADR_LOCAL_HOST db2drhost”
```

```
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_HOST db2_sd_svcip”
db2 “update db cfg for DB2_DBNAME using HADR_LOCAL_SVC db2hadrremotesvc”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_SVC db2hadrlocalsvc”
db2 “update db cfg for DB2_DBNAME using HADR_REMOTE_INST DB2_INSTANCE”
db2 “update db cfg for DB2_DBNAME using HADR_TIMEOUT 120”
db2 “update db cfg for DB2_DBNAME using HADR_SYNCMODE SYNC”
db2 “update db cfg for DB2_DBNAME using HADR_PEER_WINDOW 120”
```

8. Run the **db2** “**get db cfg for DB2_DBNAME**“ **|grep HADR** command. Example 4-9 lists the db cfg parameter for HADR on the secondary db server.

Example 4-9 HADR configuration for the secondary site

```
db2inst1@ti-2021-3:~> db2 "get db cfg for maxdb75" |grep HADR
 HADR database role                                      = STANDBY
 HADR local host name                  (HADR_LOCAL_HOST) = ti-2021-3
 HADR local service name                (HADR_LOCAL_SVC) = 55002
 HADR remote host name                (HADR_REMOTE_HOST) = 9.12.4.167
 HADR remote service name              (HADR_REMOTE_SVC) = 55001
 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) = 120
```

Note: Adjust the HADR_TIMEOUT, SYNCMODE and HADR_PEER_WINDOW values as per your requirements. SYNCMODE parameter becomes important with HADR configuration across the two sites. For more information on the HADR SYNCMODE refer to http://pic.dhe.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.ha.doc/doc/c0011724.html

9. From DB2 version 9.7 Fixpack 5, the secondary database can be configured with read-only access, which allows the application and users to run queries and reports against the database. This step is optional. Execute the steps only if the read only access is provided for the secondary database for reporting needs only.

   Example 4-10 shows the commands to set the read-only access on the secondary database.

   Example 4-10 Commands to set the read-only access on the secondary database.

```
db2set -i DB2_INSTANCE DB2_STANDBY_ISO=UR
db2set -i DB2_INSTANCE DB2_HADR_ROS=ON
```

10. Start HADR on the standby node by running the following commands.

db2 deactivate database DB2_DBNAME

db2 start hadr on database DB2_DBNAME as standby

11. Start HADR on the primary node by running the following commands.

    db2 start hadr on database DB2_DBNAME as primary

12. Verify HADR status by running the following command.

    db2pd -db DB2_DBNAME -hadr

    Example 4-11 displays the HADR status.

Example 4-11 HADR status output

```
db2inst1@ti2022-i7:~> db2pd -db maxdb75 -hadr

Database Partition 0 -- Database MAXDB75 -- Active -- Up 0 days 00:31:26 -- Date
11/08/2012 12:08:33

HADR Information:
Role    State                SyncMode   HeartBeatsMissed   LogGapRunAvg (bytes)
Primary Peer                 Sync     0                  4147958

ConnectStatus ConnectTime                            Timeout
Connected     Thu Nov  8 11:37:17 2012 (1352392637) 120

PeerWindowEnd                          PeerWindow
Thu Nov  8 12:10:25 2012 (1352394625) 120

LocalHost                                LocalService
9.12.4.167                               55001

RemoteHost                               RemoteService      RemoteInstance
ti-2021-3                                55002              db2inst1

PrimaryFile  PrimaryPg  PrimaryLSN
S0000052.LOG 1017       0x00000000A7739A7A

StandByFile  StandByPg  StandByLSN
S0000052.LOG 0          0x00000000A7340010
```

Configuration hint: It is important to note that the application servers are configured to point to the local database on their own respective sites. For this reason, there is no need to configure DB2 ACR across the sites.

Storage mirroring

This section describes a topology that utilizes a disk replication system for the DB2 database server. There are three types of data that are important to be captured and replicated to the secondary site:

- Installation data associated with the DB2 installation product
- Configuration data associated with the application and the resources needed to run them
- Run data associated with the specific instance of process and business data

The data can be divided into three independent but related consistency groups or a single consistency group. In some cases the consistency group should be expanded to include data from the WebSphere Application server. The rationale for including the data for all the servers in a single consistency group is that it is required by all the data to be consistent. In some cases, as the number of nodes grows, it may be important to limit the number of consistency groups.

The rationale for dividing the data into these consistency groups is that the actions that make the data inconsistent are different for each group:

- The install data for each of the servers is included in the same consistency group. This includes the DB2 installation folder on all the servers.
- The configuration data for each of the nodes is included in the same consistency group. This includes the DB2 instance and associated home directory, which is needed for the operation of the DB2 server.
- The run data for each of the databases in the environment is included in the same consistency group. This includes the DB2 tablespace devices, database backups, and database logs.

It is important to ensure that the write order is preserved on both sites to maintain the consistency of the data.

Oracle Active Data Guard

This is Oracle's standby database replication technology. Data Guard allows for two or more databases to synchronize through a log shipping mechanism and can act as a disaster recovery solution for the IBM SmartCloud Control Desk database. The standby database can also be opened in read-only mode, which could allow for queries to the database for reporting functionality.

Active Data Guard can be used as a local high availability solution or can be extended across sites for disaster recovery. Active Data Guard can also be combined with Oracle Real Application Clusters (RAC) technology for performance, high availability, and disaster recovery.

Active Data Guard provides several protection modes for log shipping and synchronization. It is important to research and determine which configuration works best for your organization. For more information about Active Data Guard, consult Oracle's website at:

http://www.oracle.com/ha

4.9 IBM SmartCloud Control Desk configuration

After building the infrastructure on the second site, some IBM SmartCloud Control Desk specific configurations must be implemented for a successful failover and to help lower the recovery time.

4.9.1 EAR configuration

The EAR files must be configured to be built for *primary* and *secondary* sites separately. This section describes the steps to split existing EAR build scripts into primary and secondary site specific scripts.

Consideration: If the secondary site can be configured with the same hostnames used on the primary site, and resolved within the secondary environment (DNS, routing, hosts file, for example), then additional EAR files may not be required.

For this book the variables shown in Table 4-3 are assumed. These values are not mandatory for all installations and might vary in other environments.

Table 4-3 Variables

Name	Description	Value
SCCD_HOME	IBM SmartCloud Control Desk installation path	`/opt/IBM/SMP`
SCCD_APP	IBM SmartCloud Control Desk application path	`SCCD_HOME/maximo/applications/maximo`
SCCD_DEPLOY	IBM SmartCloud Control Desk deployment path	`SCCD_HOME/maximo/deployment`

The steps are:

1. Go to the `SCCD_APP/properties` directory and copy the `maximo_UI.properties` file to `maximo_UI_primary.properties` and `maximo_UI_secondary.properties`.

2. Modify `mxe.db.url` property on `maximo _UI_secondary.properties` to use secondary site DB addresses as described in 4.9.2, “Database-related changes” on page 176.
3. Repeat steps 1 through 2 for the files `maximo_MIF.properties` and `maximo_CRON.properties`.

Example 4-12 The maximo.properties files list

```
ti2022-l10:SCCD_APP/properties # ls -l maximo*.properties
-rw-r--r-- 1 root root 723 Nov  9 14:10 maximo.properties
-rw-r--r-- 1 root root 696 Nov  8 11:43 maximo_CRON_primary.properties
-rw-r--r-- 1 root root 696 Nov  8 11:44 maximo_CRON_secondary.properties
-rw-r--r-- 1 root root 696 Nov  8 11:45 maximo_MIF_primary.properties
-rw-r--r-- 1 root root 696 Nov  8 11:45 maximo_MIF_secondary.properties
-rw-r--r-- 1 root root 318 Oct 25 18:01 maximo_ORIG.properties
-rw-r--r-- 1 root root 723 Nov  8 11:45 maximo_UI_primary.properties
-rw-r--r-- 1 root root 723 Nov  8 11:46 maximo_UI_secondary.properties
```

4. Go to the `SCCD_DEPLOY` directory and copy the `buildmaximoear_UI.sh` file to `buildmaximoear_UI_primary.sh` and `buildmaximoear_UI_secondary.sh`.
5. Modify the properties file replace command and EAR file name for `buildmaximoear_UI_primary.sh`, as shown in Example 4-13.

Example 4-13 buildmaximo_UI_primary.sh build script files

```
# Changes SCCD default EAR build definition and properties file
export BASE_DIR=./../applications/maximo
cp buildmaximoear_UI.xml buildmaximoear.xml
cp $BASE_DIR/properties/maximo_UI_primary.properties \
      $BASE_DIR/properties/maximo.properties
cp $BASE_DIR/mboweb/webmodule/WEB-INF/web_UI_CRON.xml \
      $BASE_DIR/mboweb/webmodule/WEB-INF/web.xml


export BUILD_DIR=./default
export EAR_FILENAME=sccdui_primary.ear
export MAXIMO_PROPERTIES=maximo.properties
```

6. Modify the properties file replace command and EAR file name for `buildmaximoear_UI_secondary.sh` as shown in Example 4-14.

Example 4-14 buildmaximoear_UI_secondary.sh build script

```
# Changes SCCD default EAR build definition and properties file
export BASE_DIR=./../applications/maximo
cp buildmaximoear_UI.xml buildmaximoear.xml
```

```
cp $BASE_DIR/properties/maximo_UI_secondary.properties \
      $BASE_DIR/properties/maximo.properties
cp $BASE_DIR/mboweb/webmodule/WEB-INF/web_UI_CRON.xml \
      $BASE_DIR/mboweb/webmodule/WEB-INF/web.xml


export BUILD_DIR=./default
export EAR_FILENAME=sccdui_secondary.ear
export MAXIMO_PROPERTIES=maximo.properties
```

d. Repeat steps 4 on page 175 for files `buildmaximoear_MIF.sh` and `buildmaximoear_CRON.sh`.

7. Generate new EAR files running the four custom build scripts.

Example 4-15 Build scripts

```
buildmaximoear_UI_primary.sh
buildmaximoear_MIF_primary.sh
buildmaximoear_CRON_primary.sh
buildmaximoear_UI_secondary.sh
buildmaximoear_MIF_secondary.sh
buildmaximoear_CRON_secondary.sh
```

8. Deploy *primary* EAR files on the *primary* site environment and *secondary* EAR files on the *secondary* site environment, as outlined in “Ear file deployment on WebSphere Application Server” on page 100.

4.9.2 Database-related changes

Because the secondary site is a replica of the primary, the application on the secondary site should be modified to point to the correct database.

Active-passive with DB2 HADR

On the passive site, the hostname or IP address of the database server needs to be changed to reflect the correct address of the standby database. The `maximo.ear` file needs to be rebuilt and redeployed with this new hostname or IP address. This redeployment should be done in advance of any disaster scenario so that the environment is ready to be started at any time. Any changes to the application deployment on the primary site also need to be made on the secondary site immediately after.

If the primary and secondary sites’ WebSphere Application Server and Tivoli Process Automation Engine application files are constantly synchronized using storage replication, then the `maximo.ear` file may not be able to be preconfigured

to the second site's database hostname or IP address. Another option is to use only hostnames for the database in the `maximo.properties` file and edit the hosts file on the second site's WebSphere Application Server servers to eliminate the need for application redeployment. If the second site can resolve the same hostname to the IP address of the standby database, this can help to reduce failover time and eliminate the need for application redeployment. It is important to review and plan this with the network administrators for both sites because network reconfiguration may not be possible. In this case, a redeployment of the application is required.

Tip: In all cases, the secondary site's application servers should not be running until a failover is required.

When a site switch or failover occurs, the standby database becomes the primary and the IBM SmartCloud Control Desk can be brought online. When developing a disaster recovery plan, it is important to ensure that the standby database on the passive site becomes the primary before attempting to start the application.

Tip: IBM System Automation Application Manager (AppMan) can help simplify the DB2 HADR failover to the secondary site based on using System Automation for Multiplatforms clusters on both sites. The AppMan site switch is triggered by an operator that then uses the policies defined to start DB2 and perform the HADR takeover commands. Although this configuration is not included in this book, more information can be found at:

http://www-304.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW10SA08#

Active-passive with Oracle Active Data Guard

When using Oracle Active Data Guard it is important to know which versions of Oracle are supported. IBM SmartCloud Control Desk v7.5 supports Oracle 10g Release 1, 11g Release 1 and 2. The features available may vary depending on the release used.

On a passive disaster recovery site, `mxe.db.url` in the `maximo.properties` file should point to the hostname or IP address of the standby database. Sometimes, DNS and networking configuration can help eliminate the need for the application changes. If the second site can resolve the hostname and IP address properly to its own database, then this can help reduce RTO and the need for a redeployment of the application. It is important to review and plan this with the network administrators for both sites. Network reconfiguration may not be possible or may not be an acceptable solution for some organizations, so application reconfiguration may be required.

When a site switch or failover occurs, the standby database becomes the primary and the IBM SmartCloud Control Desk can be brought online. When developing a disaster recovery plan, it is important to ensure that the standby database on the passive site becomes the primary before attempting to start the application. For more information about the Oracle Active Data Guard failover procedure, consult Oracle's documentation.

4.10 Failover scenarios and testing

When the configuration for an active-passive topology is complete, failures should be simulated and tested to ensure that failover is smooth. Environments will vary depending on the organization, but this example uses the configuration outlined in Figure 4-1 on page 151.

4.10.1 Switching sites gracefully

The first type of failover should be a graceful switch from the primary to the standby site. This scenario can be used when performing system maintenance or when there is the potential for a disaster to affect the primary site. The following is just an example of a site switch.

On the *primary* site:

1. Shut down the IBM HTTP Servers with the following command:
   - With SA MP:

     ```
     chrg -o offline ihs-rg
     ```

   - Without SA MP:

     ```
     $IHS_ROOT/bin/apachectl stop
     ```

2. Shut down the application server clusters, which will stop the application server JVMs.
   a. Log in to the Integrated Solutions Console for WebSphere Application Server as the WebSphere administrative user.
   b. Go to **Servers** → **Clusters** → **WebSphere application server clusters.**
   c. Select all the clusters from the list and click **Stop.**

d. The servers should stop and the status should update, showing offline; see Figure 4-6.

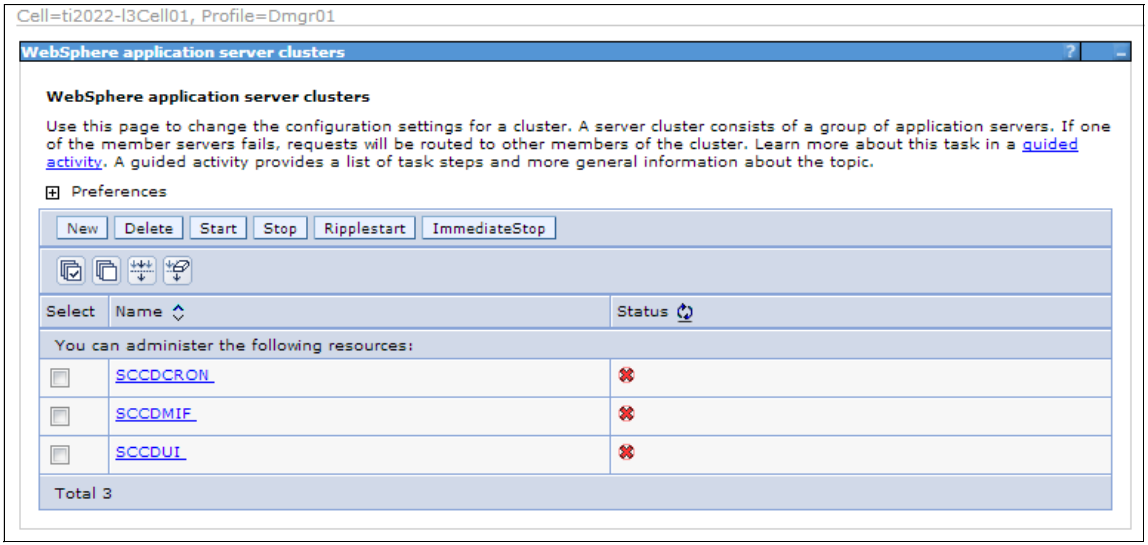

Figure 4-6 Clusters on primary location stopped

3. Shut down the WebSphere Application Server nodeagents with the following command for each nodeagent:
   - With System Automation for Multiplatforms (for all nodeagent resource groups):

     ```
     chrg -o offline nodeagent-nodename-rg
     ```

   - Without System Automation for Multiplatforms (for all nodes):

     ```
     $WAS_HOME/profiles/profile_name/bin/stopNode.sh -user userid
     -password password
     ```

4. Shut down the WebSphere Application Server Deployment Manager by running:
   - With SA MP:

     ```
     chrg -o offline dmgr-rg
     ```

   - Without SA MP:

     ```
     $WAS_DMGR_PATH/bin/stopManager.sh -user userid -password password
     ```

5. If using WebSphere MQ, the server should also be stopped. Run the following command on the active server:

   ```
   endmqm -w SCCDMIF
   ```

On the *secondary* site:

6. Execute the HADR takeover command on the active standby node as the DB2 instance administrator:

   ```
   db2 takeover hadr on database DB2_DBNAME
   ```

   Important: If the WebSphere Service Integration Bus Messaging Engine is persisted in a DB2 datastore with HADR, the DB2 takeover command should be run on the Messaging Engine database as well. This only applies if the database is different from the IBM SmartCloud Control Desk database.

7. Verify that the database role has switched to primary by running:

   ```
   db2pd -db DB2_DBNAME -hadr
   ```

   The output should show that the *HADR Role* is *Primary*. Notice the *State* is *Peer*. This is because the original site is online and being synchronized; see Example 4-16.

Example 4-16 Example db2pd output after switching roles

```
db2inst1@ti-2021-3:~> db2pd -db maxdb75 -hadr

Database Partition 0 -- Database MAXDB75 -- Active -- Up 4 days 00:33:12 -- Date
11/12/2012 18:51:30

HADR Information:
Role    State               SyncMode   HeartBeatsMissed   LogGapRunAvg (bytes)
Primary Peer                Sync     0                  0

ConnectStatus ConnectTime                           Timeout
Connected     Thu Nov  8 18:18:22 2012 (1352416702) 120

PeerWindowEnd                         PeerWindow
Mon Nov 12 18:53:11 2012 (1352764391) 120

LocalHost                               LocalService
ti-2021-3                               55002

RemoteHost                              RemoteService      RemoteInstance
9.12.4.167                              55001              db2inst1

PrimaryFile  PrimaryPg  PrimaryLSN
S0000083.LOG 3394       0x00000000C7082624
```

```
StandByFile  StandByPg  StandByLSN
S0000083.LOG 3394       0x00000000C7082624
```

8. If using WebSphere MQ, start the active and standby servers by running:

   ```
   strmqm -x SCCDMIF
   ```

9. Start the Deployment Manager by running:
   - With SA MP:

     ```
     chrg -o online dmgr-rg
     ```

   - Without SA MP:

     ```
     $WAS_DMGR_PATH/bin/startManager.sh
     ```

10. Start the nodeagents by running the following command for each nodeagent:
    - With System Automation for Multiplatforms (for all nodeagent resource groups):

      chrg -o online nodeagent-nodename-rg

    - Without System Automation for Multiplatforms (for all nodes):

      $WAS_HOME/profiles/profile_name**/bin/startNode.sh**

11. Start the IBM HTTP Servers by running:
    - With SA MP:

      ```
      chrg -o online ihs-rg
      ```

    - Without SA MP:

      ```
      $IHS_ROOT/bin/apachectl start
      ```

12. Start the WebSphere Application Server clusters, which will start the application server JVMs:
    a. Log in to the Integrated Solutions Console for WebSphere Application Server as the WebSphere administrative user.
    b. Go to **Servers** → **Clusters** → **WebSphere application server clusters.**
    c. Select all the clusters from the list and click **Start or Ripplestart**.
13. The servers should start and the status should update showing online; see Figure 4-7 on page 182.

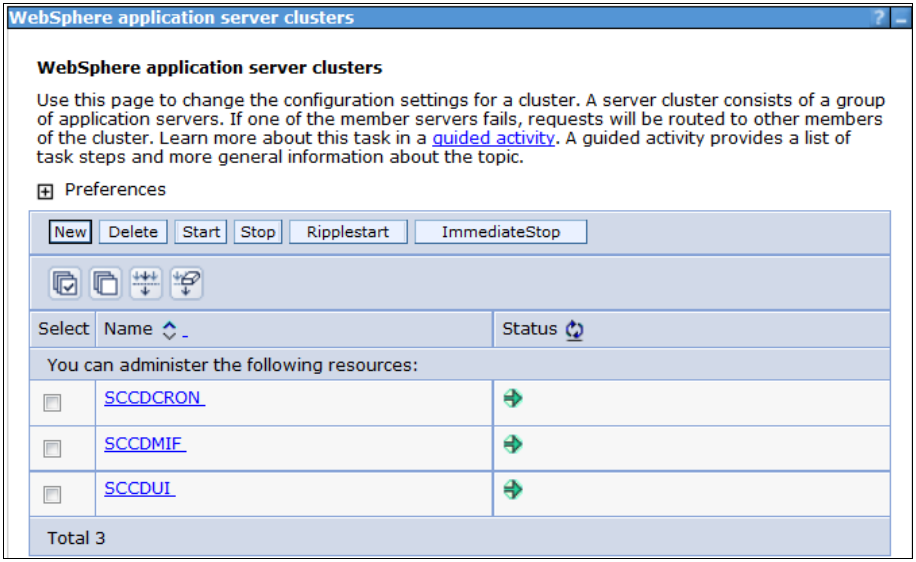

Figure 4-7 Clusters online on the second site

14. If using WebSphere Application Server SIB, verify that the Messaging Engine comes back online on the second site:

    a. Log in to the WebSphere Integrated Solutions Console.

    b. Navigate to **Service integration** → **Buses** → **intjmsbus** → **Messaging Engines.**

    c. Verify the Messaging Engine status is online; see Figure 4-8.

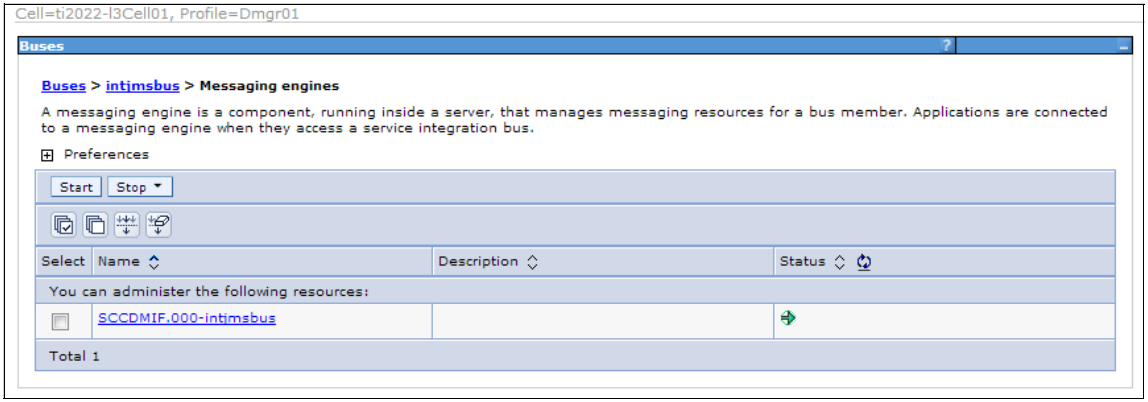

Figure 4-8 Messaging engine status

15. If using WebSphere MQ, run the following command to check the status of the queue managers:

```
dspmq -x -o all
```

16. If using a load balancer, ensure that the load balancer redirects users to the new site.

If the failover operation succeeds, it is a good idea to document the time and any notes that may be important for future reference. After the failover you should perform the same procedure again to fail back to the primary and make sure it works in both directions.

Optional step: It is good practice to warn the users when there will be a site switch to minimize the impact on operations. The Bulletin Board application in IBM SmartCloud Control Desk is one way to notify users:

1. Log in to the IBM SmartCloud Control Desk application as an administrative user.
2. On the Go To Applications menu, click **Administration** → **Bulletin Board.**
3. Click the **New Message** icon.
4. Fill out the message form (Figure 4-9) and select the appropriate dates. Try to give users plenty of notice. You can leave the Organizations, Sites and Person Groups empty to send to all users or specify an audience.

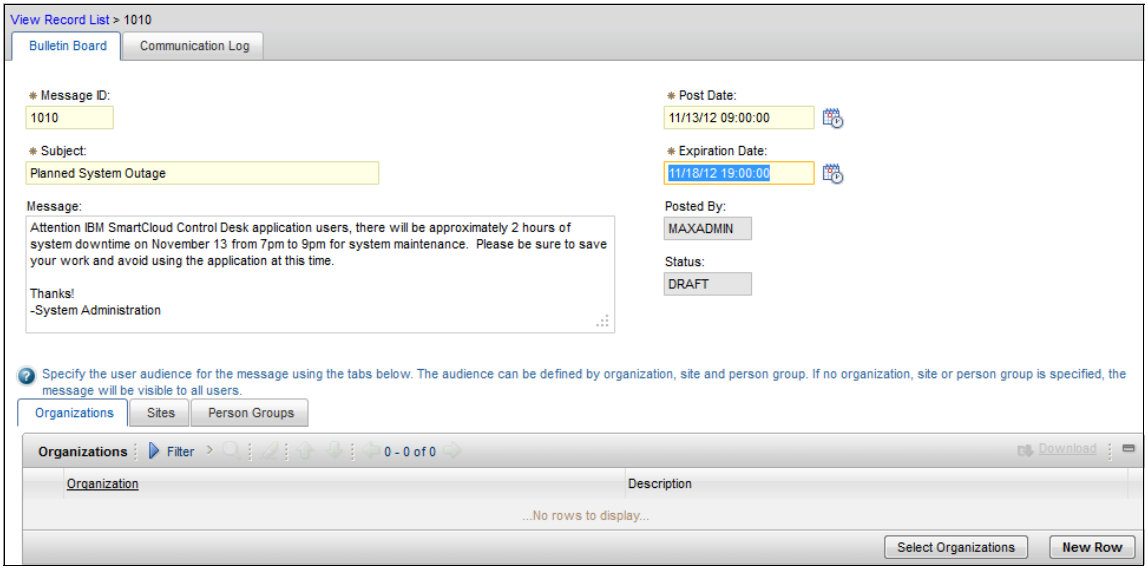

Figure 4-9 Bulletin board form example

5. **Save** the record.
6. Select the **Change Status** icon and change to **Approved**.

The message should now show on the *Bulletin Board* on users' *Start Centers* during the time period specified.

4.10.2 Disaster failover

When the active site fails, the failover procedure should be executed as efficiently as possible. It is important to simulate this type of site failure and test the recovery plan. This section outlines an example disaster recovery failover for the topology outlined in Figure 4-1 on page 151.

On the primary site:

Simulate a complete site failure. This can be done by powering off all of the systems involved in the topology or disconnecting all Ethernet interfaces. Other methods of site failure may need to be simulated depending on the infrastructure of your organization.

On the standby site:

1. Power on all operating systems if they are not already online.
2. Run the DB2 HADR takeover command. Connection to the primary will have been lost, so a takeover by force will be required. The following command will force an HADR takeover:

   ```
   db2 takeover hadr on database DB2_DBNAME by force
   ```

   Important: If the WebSphere Service Integration Bus Messaging Engine is persisted in a DB2 datastore with HADR, the DB2 takeover by force command should be run on the Messaging Engine database as well. This only applies if the database is different from the IBM SmartCloud Control Desk database.

3. Verify that the database role has switched to primary by running:

   ```
   db2pd -db DB2_DBNAME -hadr
   ```

   The output (Example 4-17) should show that the *HADR Role* is *Primary*. Notice that the *State* is *Disconnected*. This is because the original site is offline.

Example 4-17 db2pd output for HADR

```
db2inst1@ti-2021-3:~> db2pd -db maxdb75 -hadr
```

```
Database Partition 0 -- Database MAXDB75 -- Active -- Up 4 days 17:03:44 -- Date
11/13/2012 11:22:02

HADR Information:
Role    State                SyncMode   HeartBeatsMissed   LogGapRunAvg (bytes)
Primary Disconnected         Sync     0                  0

ConnectStatus ConnectTime                            Timeout
Disconnected  Tue Nov 13 11:21:35 2012 (1352823695) 120

PeerWindowEnd                          PeerWindow
Null (0)                               120

LocalHost                                 LocalService
ti-2021-3                                 55002

RemoteHost                                RemoteService      RemoteInstance
9.12.4.167                                55001              db2inst1

PrimaryFile  PrimaryPg  PrimaryLSN
S0000090.LOG 4095       0x00000000CE33FFFB

StandByFile  StandByPg  StandByLSN
S0000000.LOG 0          0x0000000000000000
```

4. If using WebSphere MQ, start the active and standby servers by running:

   ```
   strmqm -x SCCDMIF
   ```

5. Start the Deployment Manager by running:
   - With SA MP:

     ```
     chrg -o online dmgr-rg
     ```

   - Without SA MP:

     ```
     $WAS_DMGR_PATH/bin/startManager.sh
     ```

6. Start the nodeagents by running the following command for each nodeagent:
   - With System Automation for Multiplatforms (for all nodeagent resource groups):

     chrg -o online nodeagent-nodename-rg

   - Without System Automation for Multiplatforms (for all nodes):

     $WAS_HOME/profiles/profile_name**/bin/startNode.sh**

7. Start the IBM HTTP Servers by running:
   - With SA MP:

     ```
     chrg -o online ihs-rg
     ```

   - Without SA MP:

     ```
     $IHS_ROOT/bin/apachectl start
     ```

8. Start the WebSphere Application Server clusters, which will start the application server JVMs.
   a. Log in to the Integrated Solutions Console for WebSphere Application Server as the WebSphere administrative user.
   b. Go to **Servers** → **Clusters** → **WebSphere application server clusters.**
   c. Select all the clusters from the list and click **Start** or **Ripplestart**.
9. The servers should start and the status should update showing online (Figure 4-10).

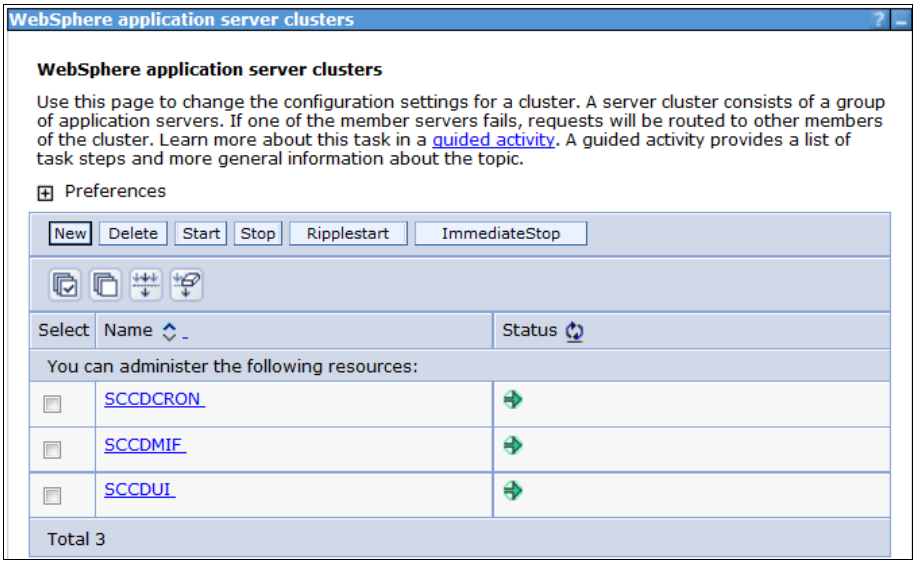

Figure 4-10 Clusters online on the second site

10. If using WebSphere Application Server SIB, verify that the Messaging Engine comes back online on the second site:
    a. Log in to the WebSphere Integrated Solutions Console.
    b. Navigate to **Service integration** → **Buses** → **intjmsbus** → **Messaging Engines.**
    c. Verify that the Messaging Engine status is online; see Figure 4-11.

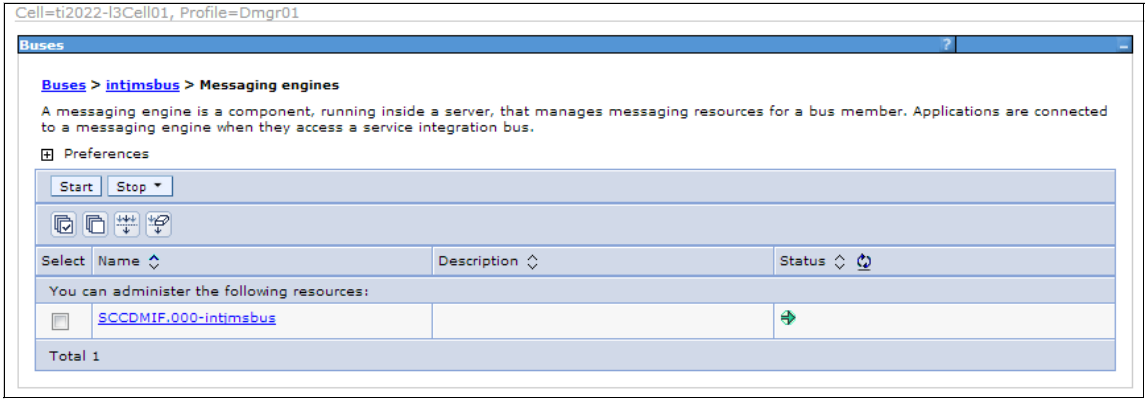

Figure 4-11 Messaging engine status

11. If using WebSphere MQ, run the following command to check the status of the queue managers (Example 4-18):

```
dspmq -x -o all
```

Example 4-18 WebSphere MQ queue manager status

```
mqm@ti2022-l11:~> dspmq -x -o all
QMNAME(SCCDMIF)       STATUS(Running) DEFAULT(no) STANDBY(Permitted)
    INSTANCE(ti2022-l11) MODE(Active)
    INSTANCE(ti2022-l9) MODE(Standby)
```

12. If using a load balancer, ensure that the load balancer redirects users to the new site.

13. Log in and test the application.

    If the failover operation succeeds, it is a good idea to document the time and any notes that may be important for future reference.

    When the original site becomes available again, you will need to reconnect the HADR back to a *peer state* for synchronization.

14. On the primary DB2 site, start the DB2 instance by running **db2start** as the instance administrator. If you are using System Automation for Multiplatforms to manage DB2, then the DB2 services should come back automatically if the *nominal status* is *online*.

15. As the DB2 instance administrator, enable HADR by running:

```
db2 deactivate database DB2_DBNAME
db2 start hadr on database DB2_DBNAME as standby
```

Tip: If System Automation for Multiplatforms is installed, it will attempt to start DB2. Once DB2 starts, crash recovery is attempted by the database server and it will try to start HADR in primary mode. This will fail but the database will be left in an activated state. Do not attempt to stop HADR.

16. Run the following command and ensure that the *State* is back in *peer*. It may take some time to go back to the peer state depending on how much data needs to synchronize.

```
db2pd -db DB2_DBNAME -hadr
```

After the failover, perform the same procedure again to fail back to the primary and make sure it works in both directions.

4.11 Symptoms of failover

When a site fails, any users connected to the IBM SmartCloud Control Desk application will lose their sessions. Any new users trying to connect to the system will receive an error page. The error page may be a 404, 503 or other HTTP error code. If users are connecting to a load balancer and receive an HTTP error 503, it may be good practice to change the 503 error page served by the load balancer to give more information that there could be a site outage and try back later.

When the site switch is complete and services (cron tasks, integrations, web services, and others) are restored, the load balancer should now send users to the active site. Depending on the networking and load balancer configuration, users should be able to access the same web address and continue using the application.

4.12 Conclusion

Adding a second site as a backup to your IBM SmartCloud Control Desk topology can allow administrators to restore services in case of a site failure or disaster. The active-passive topology is a sort of insurance policy when maintaining system availability is critical. By developing a disaster recovery plan, an organization can restore essential services and reduce downtime.

5

Utilizing multiple sites with a hybrid-active configuration

In this chapter we provide information on utilizing multiple sites simultaneously with IBM SmartCloud Control Desk. This is considered a hybrid-active configuration.

- Introduction
- Disaster recovery plan
- Prerequisites
- Web servers and load balancer
- Application server
- Database
- IBM SmartCloud Control Desk configuration
- Failover scenarios and testing

5.1 Introduction

For some organizations, having a completely passive site means that there are resources that are not utilized unless a disaster recovery is needed, so there is a desire to make use of this infrastructure as much as possible. The ability to bring certain services online and process user requests is a possibility in a multisite disaster recovery topology. You should consider many factors before choosing this type of topology.

IBM SmartCloud Control Desk does not support a completely active-active environment where there are primary databases located in each site. For this reason, the application in both sites will need to point to the same database server. Figure 5-1 shows an example of a topology where two sites have active application servers that point to the same database on `Site A`. This active database can be replicated using DB2 HADR or other database replication technologies to `Site B`. If `Site A` were to fail, the applications on `Site B` will need to be reconfigured to point to the database locally.

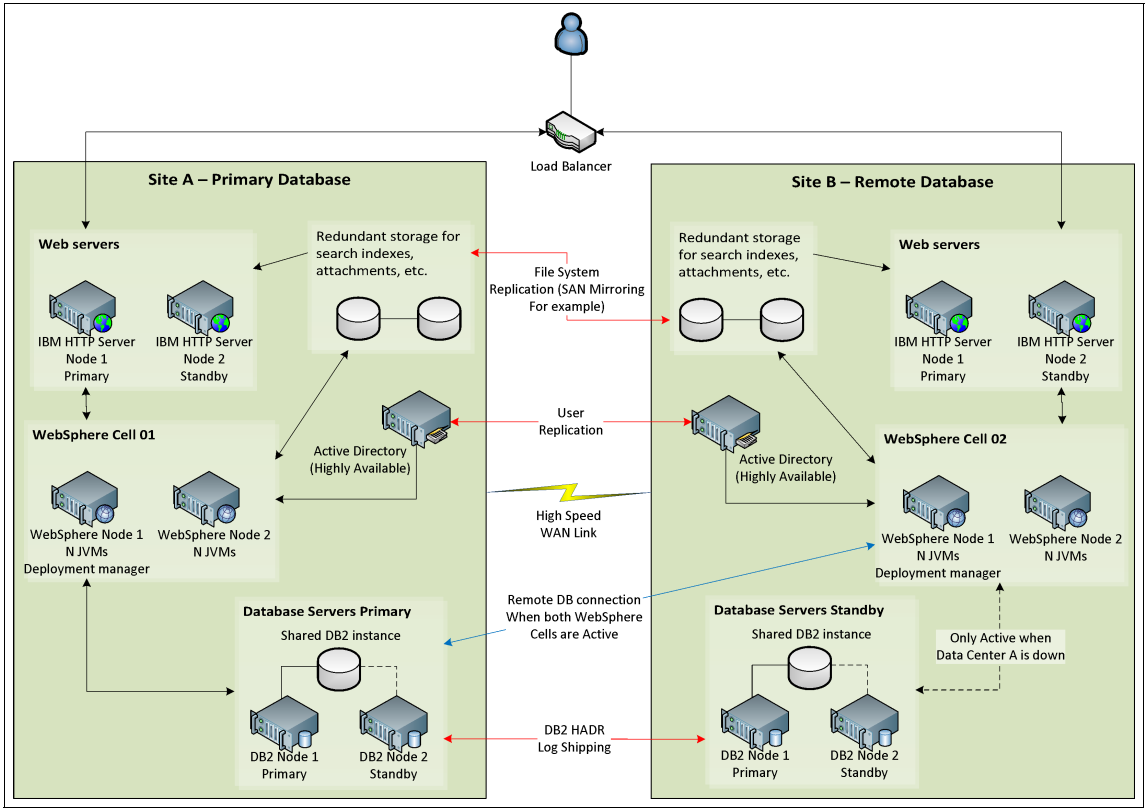

Figure 5-1 Example of a hybrid-active topology

Important: Implementing DB2 Automatic Client Reroute (ACR) or hosts file changes can speed up the recovery by eliminating the need for application redeployment on the second site after failover.

This cross-site communication relies heavily on a reliable high-speed WAN link. Application servers on `Site B` will need to process database transactions over this WAN link so users on this site might experience degraded performance when compared to users on `Site A`. This will also increase the amount of network traffic between the sites.

A load balancer is required for the topology to distribute users across the sites. This load balancer should itself be highly available so it does not act as a single point of failure. Even though the application server load is distributed to both sites, database load is all on one site.

More information: Although the IBM SmartCloud Control Desk database is only accessed on one site at a time, the standby database on the second site can be used for reporting. This database can be configured using DB2 HADR and Oracle Active Data Guard in a read-only mode, which can be queried by BIRT reports. For more information about using a standby database for reporting, review Appendix A, “Reporting” on page 219.

In an active-passive multisite environment, two sites can use *global mirroring* or other file system replication techniques for backup and restore during site failure. This type of synchronous or asynchronous replication of the entire topology, including all middleware file systems, is not possible in a hybrid-active topology. Independent WebSphere cells and web servers will be implemented on both sites.

There is also the possibility of processing background tasks other than user sessions exclusively on the second site. For instance, if the cron tasks are segregated from the user interface JVMs, the cron task cluster can be offline in one site and online in the second site. Be careful to ensure that these processes can be brought online on the opposite site in case of a site failure. If a disaster were to occur, each site should be able to take over all processing from the other and handle full capacity.

5.1.1 Things to consider with hybrid-active

Before implementing a hybrid-active topology for your organization, it is important to consider the implications of doing so. Having a passive disaster recovery site versus trying to utilize the resources on the second site may be ideal for some

organizations due to the overall complexity of a hybrid-active topology and the potential performance impact on the second site. Sometimes, it is difficult to invest in a second location and leave the resources idle, so a hybrid-active topology can be implemented.

- Recovery Time Objective (RTO)

  When you decide to activate the resources on the disaster recovery site, you need to configure certain connections to point to the primary site. The database, for instance, will need to point to the primary database server. When you have a passive disaster recovery site, the deployment can be preconfigured and ready to become active when a failover occurs. Having to reconfigure the second site when a failover or disaster happens can increase the time required to execute the disaster recovery plan.

- Performance

  Performance is a major concern with a hybrid-active topology. IBM SmartCloud Control Desk does not support a full active-active multisite environment where each site has its own primary database server. For this reason, the disaster recovery site will need to point to the database on the primary site. Remotely connecting to a database over a WAN link can dramatically affect the performance of the application on the second site and is generally not recommended. Loss of connection between the sites will render the second site inactive and user sessions on this site will fail. A high-speed reliable WAN link must exist when implementing this topology.

- Scalability and capacity planning

  Another consideration for the hybrid-active topology is the user capacity of the environment. In order to have a true disaster recovery topology, each site must be able to handle the full user load in case one of the sites were to fail. If each site is running at 40% capacity, then they could theoretically handle the entire workload if one site were to go offline. If the amount of users grows to the point where each site alone cannot handle the entire load, then there may be issues when a failover occurs. System crashes and performance degradation are examples of undesirable side effects if a site is pushed beyond capacity.

  Some organizations determine that they do not require full capacity on a disaster recovery site. Such topologies should consider user *throttling* mechanisms to ensure that systems do not become overloaded when only one site is operational.

- Complexity

  Implementing a hybrid-active topology increases the complexity of the overall environment. Failover procedures and the system configuration will become more intricate and the potential for human error increases. Integrations using WebSphere Service Integration Bus (SIB) also become more complex with a

higher chance of losing messages during a failover. In a disaster scenario, complexity may prove to be a serious issue. Complexity may have a negative effect on the recovery time.

- Licensing

  License agreements should be reviewed with your IBM sales representative. Some middleware components may have specific limited usage agreements depending on the type of license your company has. Having two sites active, even if only at the application server level, might increase licensing costs. This should be reviewed during the planning phase.

- Integration

  Integrating with external systems using IBM SmartCloud Control Desk's integration framework and WebSphere SIB will become more difficult to manage in a failure scenario. SIB is limited to the scope of a WebSphere cell, and stretching a cell across multiple sites is not recommended. Therefore, if two sites are using independent SIB messaging engines, there will potentially be stuck or lost transactions if a site fails with messages in the queue.

  WebSphere MQ can be used to help with messaging high availability and guaranteed delivery.

If a hybrid-active topology is the right choice for your organization, this chapter offers information and configuration examples for this setup.

5.2 Disaster recovery plan

In a hybrid-active topology, site failover may not work as quickly as in an active-passive topology, depending on which site fails. It may seem that because both sites are online and the applications are active that there will be next to no downtime. Quite the opposite can be true.

Because IBM SmartCloud Control Desk does not support a fully active-active configuration with independent databases in both sites, both sites will need to point to the same database server. If Site B is pointing to the database on Site A and A fails, reconfiguration of the application on B will take time. Rebuilding and redeploying the application, restarting application servers and redirecting any integrations can all increase the recovery time. Reconfiguring the networking, or modifying the host files on the recovery site can help speed up this process. Additionally, implementing DB2 ACR can help speed up the recovery by allowing the application to reconnect to the database on the second site after failover.

If the site that is not hosting the active database goes down, users here can be routed to the remaining site with the load balancer. Essential services that were

exclusive to the secondary site can be restored on the active site and continue processing. This type of failure will have less of an impact. The application will not need to be reconfigured or restarted and users who were already on this site will be able to continue.

5.2.1 Failover overview

Some steps can be taken ahead of time to prepare for and speed up the failover process. For instance, having the EAR files prebuilt and building scripts to make manual configuration changes can help. Load balancer redirection can help put users and web services onto the correct site when one fails. Network reconfiguration, DNS changes, or host file modifications can also help bring down the failover time by eliminating the need for redeployment. It is important to test the disaster recovery plan to come up with time-saving solutions that are specific to your configuration.

Understanding the potential causes of site failure is important when designing a disaster recovery plan. Some cases that might bring down one of your active sites include:

- Testing

  After implementing a second site for disaster recovery it is important to simulate a complete site failure to ensure service can be restored properly. The testing should also occur at designated intervals to be sure the environment is ready in case of an actual disaster. Documenting the lessons learned after failover tests will help evolve the overall disaster plan.

  In a hybrid-active environment it is also important to test to make sure all essential services can run at an acceptable level of performance on one site. Bringing down a site and processing all requests from the remaining site will allow administrators to monitor and confirm whether one site can handle the load. This should be tested for both sites.

- Planned takeover

  Sometimes a planned site takeover might be necessary and not just for testing purposes. Situations that may require a planned takeover include:

  – Site maintenance

    There may be situations where an entire site or several essential components of a site need to come down for maintenance. Middleware fixpacks, hardware upgrades and networking changes are a few examples of such scenarios. Redirecting all users to a single site in this situation can help decrease the downtime. One site can remain active while the other undergoes necessary upgrades.

  – Possibility of a disaster

Many disaster situations are not forecasted and happen unexpectedly. Massive power outages, earthquakes and hardware failure are examples of disasters that may happen without any warning. Other types such as hurricanes and other weather-related disasters are things that you often have warning of. When there is the possibility of these types of disasters it may be a good idea to divert all the processing to the site that is least likely to be affected.

- Disaster

  Unpredictable events can affect one of your sites and bring down some of the essential services. Weather, human error, hardware failures, malicious users are some of the many types of problems that can bring down a site. When a disaster occurs, it is time to execute the disaster recovery plan. Restoring lost services as quickly as possible on the remaining site will allow for a minimal impact on operations.

5.2.2 Designing a disaster recovery plan

Another component to the disaster recovery infrastructure is a well-designed disaster recovery plan. In a hybrid-active topology, when both sites are operational to some level, only one site will host the active primary database—and there could even be some difference over which site processes certain tasks.

When a site fails, then any essential services from that site will need to be brought online on the remaining site. This procedure could differ depending on which site fails. For instance, if Site A hosts the active database and Site B fails, then the database role switch will not need to occur. If Site A were to fail, then the standby database on Site B will need to become the new primary. Other services such as integrations, cron tasks and reporting may need to be brought online or redirected depending on which site fails. The procedure will vary depending on how the workload is distributed across the sites.

Without a procedure in place, this failure recovery can take too long or fail completely. A plan that prepares for potential loss of either site will be needed. Disaster recovery plans should include:

- Names and contact information of all parties involved with the failover procedure. It is a good idea to have backups for everyone in case someone cannot be contacted at that time.
- A detailed step-by-step outline of the order of operations for failover.
- System information required for administrators to restore services.
- Test cases so administrators can verify that the system is functioning properly before allowing users to reconnect.

- A good system backup plan must be devised. Having good backup mitigates the risk of failure in case both disaster recovery sites fail. The frequency of the backups will be determined by the data loss tolerance for your organization.

These were just a few of the common needs in a disaster recovery plan. Detailed documentation and thorough testing can help with creating a solid plan. Communication of this plan to all administrators is extremely important. Many disaster recovery procedure failures are attributed to a lack of internal planning and coordination amongst all system administrators.

Important: Because each environment is unique and the distribution of services across the sites will vary from company to company, there is no specific plan outlined in this book. The complexity of the disaster recovery plan depends on the complexity of the overall configuration.

5.3 Prerequisites

There are many prerequisites to implementing a disaster recovery topology. Some of these topics are covered in this book and others are assumptions. It is best to research which solutions are best for your organizational needs.

Repetition: Although the prerequisites for this scenario are exactly the same as for the implementation of a passive disaster recovery site, we decided to repeat them here for convenience.

- Local high availability

  Most disaster recovery topologies are supplemental to a local high availability solution. Often, process and hardware failures can be corrected quickly without the need for an entire site to fail. Refer to Chapter 3, “Local high availability topology” on page 25 for more information.

- Load balancer

  A load balancer can be used in front of the web servers to provide balancing across several web servers and also provide a transparent access point for users across the active sites. There are hardware and software solutions for load balancers but care should be taken to ensure that this is not a single point of failure. Having a load balancer that can detect when one site is offline can help ease the transition in a disaster scenario. Most load balancing solutions have high availability and disaster recovery options.

- Networking

The network link between the two sites is critical for synchronization of the application and data. Network administrators should be involved in the planning process. The network may need to be upgraded when connecting a second site. Redundant network links could help avoid synchronization and communication loss.

- Storage replication and sharing

  When implementing a second site there are files that will get stored on the primary and should be replicated to the passive site. Attached documents, global search indexes, integration framework files are examples of such files. If the primary site fails, these files may be needed on the standby site to continue with full application functionality. An example storage solution is disk-based mirroring as described in 4.4, “Storage replication” on page 156.

- A second site

  An obvious prerequisite to a disaster recovery plan is a second site that can take over from the primary when a disaster occurs. Careful consideration should be taken when selecting a location. Sites too close could both be affected in a disaster, but sites too far will not be able to synchronize as quickly and have data loss. In a hybrid-active configuration, this proximity of the second site could affect performance because there will be a remote database connection.

- Administrators with necessary skills

  When implementing disaster recovery technologies such as database and file system replication, the complexity of the IBM SmartCloud Control Desk topology increases. Administrators who are familiar with these technologies and posses the skills required to configure, maintain and test the infrastructure are critical. Lack of coordination amongst the team can lead to a failed disaster recovery.

- Application installation files on both sites

  It is important that the application installation directory for IBM SmartCloud Control Desk are copied to the secondary site. If the primary site fails and application reconfiguration is required, these files will need to be accessible from the secondary. These directories should also be kept synchronized with each other after any changes. Frequent backups of the application installation directories is advised.

5.4 Web servers and load balancer

Load balancing is an important part of the hybrid-active topology to help spread users and web services across both sites. The load balancer can send users to the web servers or directly to the application server JVMs.

5.4.1 IBM HTTP Server

The IBM HTTP Server (IHS) can be used in both sites to balance the user load across the application servers. The WebSphere Plug-in should be installed alongside IHS. In a hybrid-active topology, it is advisable to configure the load balancer to direct users and web services across all the IHS servers.

If all of the application servers or other essential components of a site go down and the site is unusable, the IHS servers should also be shut down so the load balancer will not direct users there.

The IHS is also used to retrieve *attached documents* if this functionality is enabled. In order to have both sites access the attachments, it is important to replicate these files to the second site. Otherwise, users will only be able to retrieve attachments from the site they are connected to. The IBM HTTP Servers should be configured to point to the replicated file system on the local site to avoid the need for reconfiguration upon failover.

More information: Details on configuring attached document functionality can be found at:

`http://pic.dhe.ibm.com/infocenter/tivihelp/v58r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Ffm_sag%2Fattacheddocs%2Fc_attached_doc_config.html`

5.4.2 Load balancer

Load balancers play an important role in the hybrid-active topology. Load balancers can be used to balance users and web services across the sites. An intelligent load balancer can also detect when web servers or a complete site are offline and stop directing traffic to that site. If the load balancer is located in one of the sites, careful consideration should be taken to ensure that it is not a single point of failure. Having a standby load balancer at the second site can help ensure that users are able to access the application when a site failure occurs.

For more information about load balancers in a disaster recovery topology, review 4.5.2, “Load balancer” on page 159.

5.5 Application server

In the hybrid-active topology, the load will be processed simultaneously by both sites utilizing two different application server installations. Capacity should be

considered because in the event of a site failure, the remaining site should be able to handle the extra load.

5.5.1 WebSphere Application Server

The configuration for the WebSphere Application Server for the hybrid-active topology is the same as described in 4.6.1, “WebSphere Application Server” on page 160 for the active-passive topology.

5.6 Integration framework

The integration framework must be able to recover all transactions without loss when a failure occurs. After configuring the WebSphere Application Server, either Service Integration Bus (SIB) or WebSphere MQ Server can be chosen as the JMS provider.

Important: When utilizing SIB, two different datastores will be configured to handle both sites simultaneously. In the event of a site failure, *all integration transactions* of the site that failed will be stuck until service is restored.

This section describes the necessary configuration for each of these JMS providers.

5.6.1 Service Integration Bus configuration

To enable integrations to be processed in both environments, two datastores must be defined within the messaging engine of each site. The messaging engine topology will be configured as shown in Figure 5-2 on page 200.

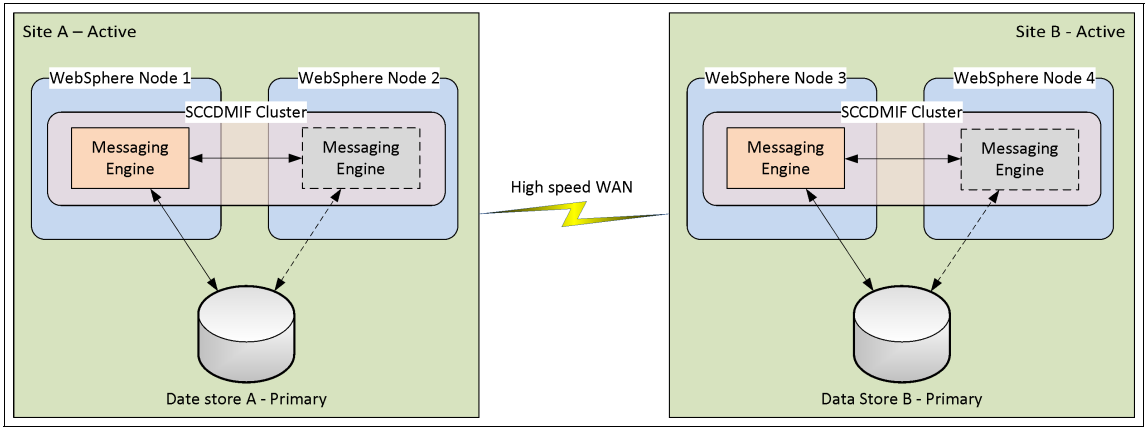

Figure 5-2 Independent messaging engines

Datastore configuration

Following are the steps to configure different data stores for each WebSphere Application Server (refer to Figure 5-3 on page 201):

1. Log in to the primary site Integrated Solutions Console.
2. Navigate to **Service integration** → **Buses** → **intjmsbus** → **Messaging engines** → **SCCDMIF.000-intjmsbus** → **Message store**.
3. Type `SIBSITEA` as `Schema name`.

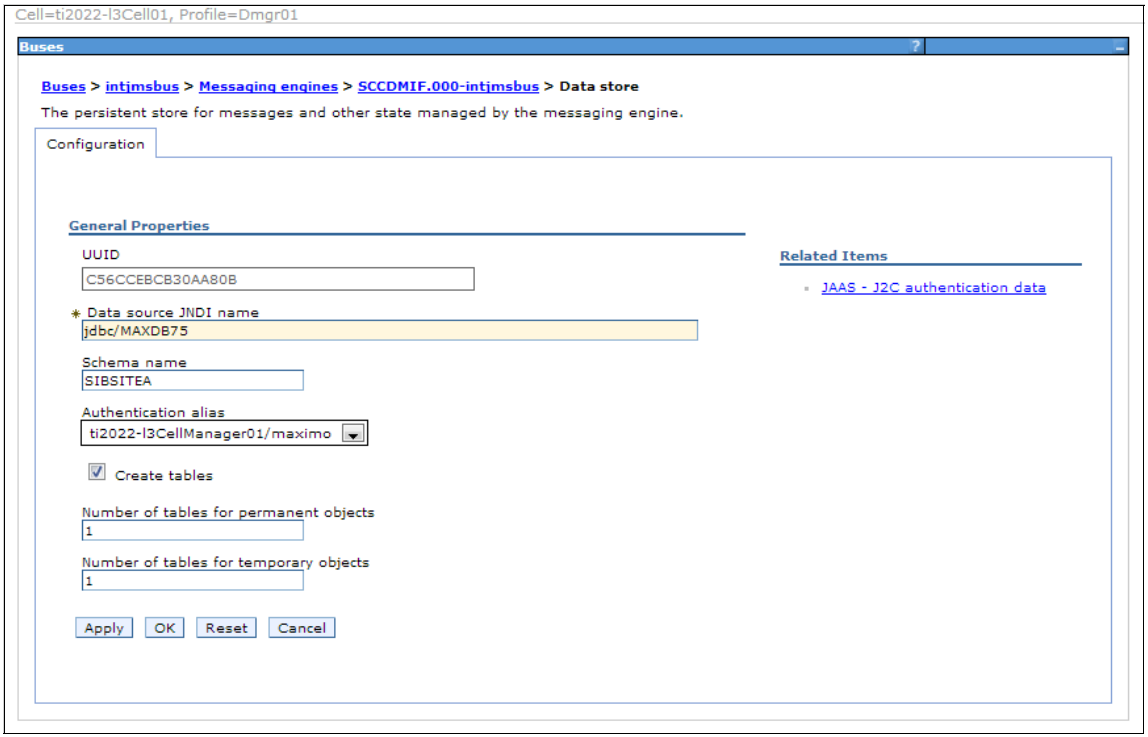

Figure 5-3 SIBSITEA data store

4. Select **OK**.
5. Save and synchronize changes.
6. Repeat steps 1 on page 200 through 2 on page 200 on the secondary site WebSphere Application Server using `SIBSITEB` as `Schema name`.

Cron task configuration

To ensure that both environment's JMS resources will be utilized, two new `JMSQSEQCONSUMER` cron task instances must be created. Following are the steps to configure the cron tasks.

1. Log in to IBM SmartCloud Control Desk and navigate to **System Configuration** → **Platform Configuration** → **Cron Task Setup**.
2. Select `JMSQSEQCONSUMER` cron task.
3. Uncheck the `Active?` check box for `SEQQIN` and `SEQQOUT` instances.
4. Select `SEQQIN` instance and select **Duplicate.**

5. Type SEQQIN_SITEA as Cron Task Instance Name and check the Active? check box.
6. In the Cron Task Parameters section, set the TARGETENABLED parameter to 1.
7. Select SEQQIN instance and select **Duplicate.**
8. Type SEQQIN_SITEB as Cron Task Instance Name and check the Active? check box.
9. In the Cron Task Parameters section, set the TARGETENABLED parameter to 1.
10. Select **Save.**
11. Repeat steps 4 through 10 for the SEQQOUT instance.
12. Navigate to **System Configuration** → **Platform Configuration** → **System Properties**.
13. In the Instance Properties section, filter by the mxe.crontask.donotrun property.
14. Add the following values to SCCDCRON1 and SCCDCRON2 servers (Example 5-1:

Example 5-1 mxe.crontask.donotrun additional values

```
JMSQSEQCONSUMER.SEQQIN_SITEA,JMSQSEQCONSUMER.SEQQOUT_SITEA,JMSQSEQCO
NSUMER.SEQQIN_SITEB,JMSQSEQCONSUMER.SEQQOUT_SITEB
```

15. Restart application servers SCCDCRON1 and SCCDCRON2.

Application server parameter configuration

After creating the JMSQSEQCONSUMER cron task instances, they must be target enabled. Following are the steps to configure the application servers:

1. Log in to primary WebSphere Application Server.
2. Navigate to **Servers** → **Server Types** → **WebSphere application server** → **SCCDMIF1** → **Java and Process Management** → **Process definition** → **Java Virtual Machine**.
3. Add the following values to the Generic JVM arguments field.

   ```
   -DJMSQSEQCONSUMER.SEQQIN_SITEA=1 -DJMSQSEQCONSUMER.SEQQOUT_SITEA=1
   ```

4. Select **OK**.
5. Repeat steps 2 through 4 for application server SCCDMIF2.
6. Save and synchronize changes.
7. Log in to the secondary WebSphere Application Server.
8. Navigate to **Servers** → **Server Types** → **WebSphere application server** → **SCCDMIF1** → **Java and Process Management** → **Process definition** → **Java Virtual Machine**.

9. Add the following values to the `Generic JVM arguments` field:

   `-DJMSQSEQCONSUMER.SEQQIN_SITEB=1 -DJMSQSEQCONSUMER.SEQQOUT_SITEB=1`

10. Select **OK**.
11. Repeat steps 8 through 10 for application server `SCCDMIF2`.
12. Save and synchronize changes.
13. Restart application servers `SCCDMIF1` and `SCCDMIF2`.

5.6.2 WebSphere MQ configuration

To enable integrations to be processed in both environments, one WebSphere MQ multi-instance queue manager will be used as the JMS provider. In this configuration, there will be no transaction loss in the event of a failure as long as the file system replication is synchronized. The queue manager topology will be configured as shown in Figure 5-4. All the WebSphere Nodes on both sites point to the same multi-instance on a single site, here Site A. Upon site failure, the multi-instance can be brought online on the second site, Site B, and the WebSphere Nodes will redirect requests.

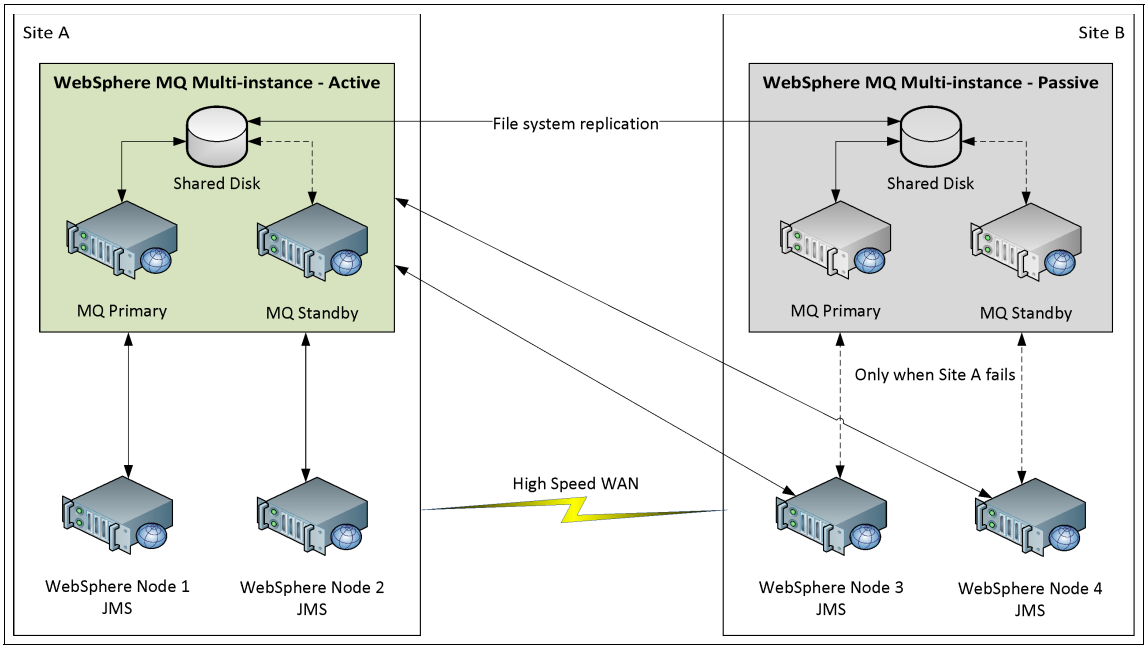

Figure 5-4 WebSphere MQ multi-instance with disaster recovery

More options: Although this highlights a disaster recovery solution for WebSphere MQ, there are other more complex configurations available for MQ, such as MQ clusters that allow for active message processing on both sites. Further research into WebSphere MQ disaster recovery and clustering solutions is suggested.

The queue manager will only be available at one site at a time, and its data and log files will be replicated to the inactive site. The configuration of the queue manager is the same as described in 4.7.2, “WebSphere MQ configuration” on page 165.

For this book the variables shown in Table 5-1 are assumed. These values are not mandatory for all installations and might vary in other environments.

Table 5-1 Variables

Name	Description	Value
siteamqhost1	Site A WebSphere MQ primary server hostname	`ti2022-l11.itso.ibm.com`
siteamqhost2	Site A WebSphere MQ secondary server hostname	`ti2022-l9.itso.ibm.com`

To reconnect to another site in the event of a site failure, the WebSphere Application Server must be configured as described in the following steps:

1. Log in to primary WebSphere Application Server and navigate to **Resources** → **JMS** → **Queue connection factories** → **intconfact** → **Custom properties** → **XMSC_WMQ_CONNECTION_NAME_LIST**.
2. Change the current `Value` to the value shown in Example 5-2.

Example 5-2 WebSphere MQ connection name list

```
siteamqhost1(1414),siteamqhost2(1414),sitebmqhost1(1414),sitebmqhost2(1414)
```

3. Select **OK**.
4. Navigate to **Resources** → **JMS** → **Activation specifications** → **intjmsact** → **Custom properties** → **connectionNameList**.
5. Change the current `Value` to the value shown in Example 5-2.
6. Select **OK**.
7. Repeat steps 4 through 6 for activation specification `intjmsacterr`.
8. Save and synchronize changes.

9. Repeat steps 1 through 8 for the secondary WebSphere Application Server.

5.7 Database

Disaster recovery for an enterprise application means that all critical business operations are recovered in case of any disaster or site-wide outage. Some organizations have little or no tolerance for data loss, in which case the disaster recovery solution needs to be deployed to restore data to the applications rapidly. The solution must ensure the consistency of the data, allowing for restoration of the systems and applications reliably and fast.

Bringing down the database disrupts the IBM SmartCloud Control Desk function. Various disaster recovery database configurations are available. It is suggested that you review IBM SmartCloud Control Desk disaster recovery documents to choose the optimum solution for your environment.

Here we describe the hybrid-active disaster recovery setup for IBM SmartCloud Control Desk using DB2 and Oracle databases, and how to set up replication features using DB2 and Oracle. In this scenario all the WebSphere Application Servers will still point to the database on the primary site. The database on the secondary site can be used for reporting or ad hoc querying. In case of a failover, the database on the secondary site will take over the operations, and all the application servers will now point to this database. Some organizations may choose to mirror the database using disk mirroring techniques, in which case the secondary database cannot be used for any reporting.

5.7.1 DB2 HADR

For the organizations who have little or zero tolerance for data loss, DB2 High Availability and Disaster Recovery allows replication of any logged database activity to a local or remote location. A DB2 HADR primary database uses internal processes to ship database log buffers to an HADR standby database. A process on the standby server then replays the log records directly to the standby database. The secondary server is always in the rollforward mode, in the state of near readiness, so the takeover to the standby server is fast. The standby database can be converted to the primary database and accessed by applications and users in the event of a disaster, or if the primary server fails. For more information, refer to “DB2 HADR for disaster recovery” on page 167.

Optional DB2 HADR with ACR

If the application is configured with hostnames in the `maximo.properties` file, then Automatic Client Reroute (ACR) can be used to help reconnect the

application to the secondary site in case of a disaster. The application server on the secondary site can connect to the database after the HADR takeover and resume the work without needing to restart the application JVMs. There is no guarantee of a seamless failover across sites with ACR, so this should be tested for each environment.

If ACR is configured with the alternate hostname in the database catalog, the alternate server information is cached in memory when the JVMs start. In case the primary site becomes inaccessible, ACR will direct the connections to the secondary site. If the JVMs are restarted after the primary site failure, the alternate server information cannot be read and the JVMs will fail to start because they will not be able to connect to the database server on the primary site and will have no information about the alternate database server on the secondary site. There are two solutions to avoid this scenario. In the first solution the application has to be rebuilt after modifying the database properties in the `maximo.properties` file. In the second solution the hostnames can be modified in the `/etc/hosts` file to avoid the need for rebuild and redeploy.

To set up ACR with HADR in the active-hybrid disaster recovery setup, complete the following steps:

1. Update the `/etc/hosts` file on the WebSphere and DB2 servers on both sites. Add the hostnames for the DB2 server from both both sites in the host file as shown in Example 5-3.

   Example 5-3 Example of hostfile change

   ```
   9.42.170.180      hostname.db2.siteB
   9.12.4.167        hostname.db2.siteA
   ```

2. Update the alternate server information for the database catalog on the DB2 database across both sites.

   On the primary site run this command:

   ```
   db2 "update alternate server for database DB2_DBNAME using hostname
   hostname.db2.siteB port 60000"
   ```

   On the secondary site run this command:

   ```
   db2 "update alternate server for database DB2_DBNAME using hostname
   hostname.db2.siteA port 60000"
   ```

3. In the scenario where the primary site is functional, comment out the entry for hostname for Site B in the `/etc/hosts` file on all WebSphere Application Servers. In this case, the application will be connected to the database on Site A.

4. In case of disaster, take over the primary HADR role on the secondary site:

   `db2 "takeover hadr on database DB2_DBNAME by force"`

5. Update the `/etc/hosts` files on the WebSphere Application Server to swap the IP address for the hostnames, shown in Example 5-4.

   Example 5-4 Example of the hostfile modified during failover

```
9.42.170.180     hostname.db2.siteA
9.12.4.167       hostname.db2.siteB
```

6. The application server on the secondary site will now connect to the database on the same site. Users who were connected to the application server on the primary site will have to relaunch the application and re-login.

Warning: The ACR transition may allow users to connect to the secondary site after the HADR takeover command but there may be errors and inconsistent behavior in the user interface. Although this may hasten the failover/recovery to the secondary site, the application servers should be recycled as soon as possible.

5.7.2 Oracle Active Data Guard

For Oracle Active Data Guard the same concepts apply as specified in “Oracle Active Data Guard” on page 173 in the passive disaster recovery scenario.

5.7.3 Storage mirror

Disk mirroring techniques can be used to synchronize data across the two sites. In this scenario the secondary database will be a mirror image of the primary but in the offline state. The secondary database cannot be used for any reporting or any other ad hoc query. For more information about the database synchronization using disk mirroring, refer to “Storage mirroring” on page 163.

5.8 IBM SmartCloud Control Desk configuration

For the hybrid-active configuration, IBM SmartCloud Control Desk needs to connect to a database system using a different address in the event of a failure. To achieve this, the EAR files must be configured as described in 4.9.1, “EAR configuration” on page 174. After splitting into primary and secondary EAR files, the same file should be deployed to both environments and will vary depending on the active database. Manipulating the hosts file on the second site application

servers during a failover can help eliminate the need for having separate EAR files for the primary versus the secondary site. Adding DB2 ACR to the database configuration can allow the application to reconnect to the second site database without restarting the application servers in the event of a primary site failure.

5.8.1 Database-related configurations

In the failover scenario to a secondary site, the application server will have to be reconfigured to point to the database on the secondary site. The application EAR has to be rebuilt and redeployed. This could potentially take a longer time if the environment is configured with multiple JVMs. If the BIRT Report Only Server (BROS) was configured to point to the secondary database for reporting, then there are no changes required for the BROS server.

Some organizations may modify the network setup or the hostname setup to resolve the database hostname from the primary site to the secondary. In this case the application does not need to be modified. The application has to be restarted before the users can resume their work. Optionally, DB2 ACR can be added to the database configuration, which will allow the application to reconnect to the secondary site database if the primary site fails. Refer to 5.7.1, “DB2 HADR” on page 205 for more information.

5.9 Failover scenarios and testing

If a hybrid-active topology is created, there will be two different failover scenarios, depending on which site fails. If the primary (the site hosting the active database) fails, there will be a more complex failover procedure to reestablish availability. If the secondary (the site with remote database connection) fails, users on the primary should still be able to access the application. Any services running on the failed site will need to be brought online on the remaining site either manually or automatically. Users who were accessing the second (failed) site will need to be redirected to the first site by the load balancer. Their sessions will be lost and will be redirected back to the login page.

5.9.1 WAN link failure

If there is a problem with the WAN link between the sites, the synchronization will halt and the database connection will be severed from the secondary site. To avoid problems with the application and users being directed to the secondary site, stop all processes on the secondary site until communication is restored.

Reporting may have to be reconfigured when a WAN link fails, depending on which database the reports access. Refer to Appendix A, “Reporting” on page 219.

5.9.2 Primary site failure

The primary site is the site that hosts the active database instance. The impact of a primary site failure will be greater than a secondary site failure for this reason. Any other services, such as WebSphere MQ, will have to be failed over to the secondary site if they are essential.

Using the hosts file or DNS to speed up recovery

When the primary site fails, the application needs to be reconfigured to point to the database on the secondary site. Rebuilding and redeploying the EAR files can take a long time, depending on the size of the environment. If the `maximo.properties` file uses a hostname in the database connection string, changing the `/etc/hosts` file entries on the database and application server machines can help eliminate the need for a redeployment.

When both sites are operational, the `/etc/hosts` files on both sites should resolve the hostname to the primary site’s database server. If a failure of the primary site occurs, the `/etc/hosts` files on the standby site can be modified to resolve to the IP address (or service IP address if applicable) on the standby site.

Another option is to modify the DNS server entries to resolve the hostname to the correct IP on the secondary site after failure. Because DNS entries are cached and may have to filter through several DNS servers, this change may take a long time, which may not be considered acceptable.

Changing the `/etc/hosts` file entries or DNS resolution may go against some organization’s security policies. The operating system and network administrators should be involved with the planning of this solution.

Important: The hostname resolutions are cached in the application server JVMs when they start up. For this reason, the application servers will have to be restarted after the `/etc/hosts` file or DNS change is complete and the database takeover command has been run.

DB2 ACR

DB2 Automatic Client Reroute (ACR) can be implemented to automatically reconnect the application to an alternate database host upon failover. Setting the second site as the alternate for the first site and vice versa can allow IBM SmartCloud Control Desk to reconnect without the need for restarting the

application servers. Refer to 5.7.1, “DB2 HADR” on page 205 for more information.

Recovery procedure

When the primary site fails, essential services such as the database will need to be failed over to the secondary site. This is an example procedure and may differ depending on your environment.

Database

The database should be the first priority when a primary site failure occurs. The standby database will need to become the primary. When using DB2 HADR on the standby site:

1. Run the following command on the standby database as the DB2 instance administrator:

   ```
   db2 takeover hadr on database DB2_DBNAME by force
   ```

2. Check the status of the database to ensure that it is now the *primary*:

   ```
   db2pd -db DB2_DBNAME -hadr
   ```

   The output should show that the *HADR Role* is Primary. Notice that the *State* is *Disconnected*. This is because the original site is offline; refer to Example 5-5.

Example 5-5 Database role after takeover

```
db2inst1@db2server2:~> db2pd -db maxdb75 -hadr

Database Partition 0 -- Database MAXDB75 -- Active -- Up 2 days 14:49:46 -- Date
11/16/2012 09:51:30

HADR Information:
Role    State               SyncMode   HeartBeatsMissed   LogGapRunAvg (bytes)
Primary Disconnected        Sync     0                  0

ConnectStatus ConnectTime                           Timeout
Disconnected  Thu Nov 15 13:41:18 2012 (1353004878) 30

PeerWindowEnd                       PeerWindow
Null (0)                            120

LocalHost                              LocalService
ti-2021-3                              55002

RemoteHost                             RemoteService      RemoteInstance
9.12.4.167                             55001              db2inst1
```

```
PrimaryFile  PrimaryPg  PrimaryLSN
S0000122.LOG 2653       0x00000000EDD9DB12

StandByFile  StandByPg  StandByLSN
S0000000.LOG 0          0x0000000000000000
```

When the primary site becomes available again, you will need to reconnect the HADR back to a *peer state* for synchronization.

3. On the primary DB2 site, start the DB2 instance by running **db2start** as the instance administrator. If you are using System Automation for Multiplatforms to manage DB2. then the DB2 services should come back automatically if the *nominal status* is *online.*
4. As the DB2 instance administrator, enable HADR by running:

   ```
   db2 deactivate database DB2_DBNAME
   db2 start hadr on database DB2_DBNAME as standby
   ```

   Tip: If System Automation for Multiplatforms is installed, it will attempt to start DB2. Once DB2 starts, crash recovery is attempted by the database server and will try to start HADR in primary mode. This will fail but the database will be left in an activated state. Do not attempt to stop HADR.

5. Run the following command and ensure that the *State* is back in *peer*. It may take some time to go back to the peer state depending on how much data needs to synchronize.

   ```
   db2pd -db DB2_DBNAME -hadr
   ```

6. It may be desirable to make the original site the primary site again when synchronization is complete by running a graceful takeover. To do this, run:

   ```
   db2 takeover hadr on database DB2_DBNAME
   ```

   Important: The application may need to be reconfigured to point back to the primary site after takeover is complete. For more details, see “Application” on page 212. Other essential services, such as IBM HTTP Server, WebSphere Application Servers, and integrations should be started on the primary site after the takeover is complete.

If the site has been down for a long time or has to be completely rebuilt, a backup and restore will be necessary to synchronize the databases. HADR will need to be reconfigured on the rebuilt server as well. For more information about this procedure, refer to “DB2 HADR configuration” on page 168.

Application

If the primary site fails, then the remaining applications on the standby site will be pointing to a database that is no longer accessible. For this reason, the application will need to point to the new primary database on the secondary site.

There are a couple of different options for reconfiguring the application:

- Rebuild and redeploy
  a. Edit the `maximo.properties` file for the IBM SmartCloud Control Desk application and specify the hostname or IP address of the secondary database in the database connection string.
  b. Stop the application servers and clusters.
  c. Rebuild the `maximo.ear` files and redeploy them to the application servers. More information about rebuild and redeploy can be found at:

     `http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mam.inswas.doc%2Finstall%2Fc_ccmdb_deployccmdbearfiles.html`

     Helpful tip: Modifying `maximo.properties` and rebuilding the EAR files can take a long time and delay the failover process. The EAR files should be built beforehand and ready in case of a disaster to speed up the process.

  d. Restart the application servers and clusters for the changes to take effect. The database takeover commands will need to have been run before attempting to start the application.
- Hosts file and DNS switch

  For information about this procedure, review “Using the hosts file or DNS to speed up recovery” on page 209.
- DB2 ACR

  For more information, review 5.7.1, “DB2 HADR” on page 205.

Important: Additional application servers may need to be brought online on the second site to handle the user load of all users accessing a single site. Special care should be taken to ensure that the extra load does not crash the remaining servers.

Reporting

If reports have been configured to point to the standby reporting database as explained in Appendix A, “Reporting” on page 219, then no additional reconfiguration should be required.

Integrations

If using SIB as JMS provider, the messages on the failed site will remain on the datastore and will be processed when the site recovers from the failure automatically.

When using WebSphere MQ, if the active multi-instance queue manager was on the failed site, it must be started on the remaining site. Utilize the **`strmqm -x SCCDMIF`** command on both active and standby servers for the remaining active site.

Symptoms of failure of a primary site

The symptoms of a primary site failure will vary depending on the topology for each organization and which services are enabled on this site. Some common symptoms include:

- Performance

  Performance may be impacted when a site fails because the full user load will now be directed to one site. It is important that both sites can handle the extra processing without overloading the systems.

- Lost user sessions

  Because the primary site hosts the active database, when a failure occurs the application will have to be reconfigured to use the database on the secondary site. For this reason, the application will fail and all users will lose their sessions during failover. ACR may be able to alleviate some of these symptoms, but this is not guaranteed.

- Lost or stuck transactions with SIB

  Upon a failure, SIB messages will be in the datastore and will only be processed when the failed site recovers. This is due to the messaging engine and destination UUIDs used to put the messages on the datastore.

- Integration recovery delay with WebSphere MQ

  When a failure occurs, another instance of the queue manager must be started on the remaining site. After starting the queue manager, the automatic client reconnect feature will reroute active connections to the new instance. The delay will occur until the new instance is active and all connections are rerouted.

- System unavailability during failover

  Users who are trying to access the application during the failover period will not be able to connect. They will most likely receive an HTML error code from the load balancer. When the system becomes available on the secondary site, users can now begin to connect to the application.

- Cron task delay

  Cron tasks that do not have the target enabled to the primary site will recover and start on the secondary after a short delay. Cron tasks running exclusive to the primary site using the *target enabled* or *do not run* functionality may have to be manually reconfigured on the secondary.

5.9.3 Secondary site failure

The secondary site is remotely connected to the database instance on the primary site. For this reason, if the secondary site fails, the users connected to this environment will lose their session. Users should be able to continue accessing the application after logging back in. This reconnection should be handled by the load balancer and should send users to the primary site only.

Cron tasks, integrations, and other services running exclusively on this site will also stop. These services will have to be brought online on the primary manually after the failure. If configuring reporting against the standby database as defined in Appendix A, “Reporting” on page 219, this will have to be reconfigured to point to the primary site database connection string.

Services that are not exclusive to the secondary site should failover automatically to the primary.

Recovery procedure

This is an example procedure for recovering from a standby site failure. The steps may be different, depending on your environment.

Database

If the secondary site fails, the impact from a database level should be minimal because the primary database will still be active.

When the secondary site is repaired and comes back online, the database should be started as HADR standby if using DB2 HADR:

1. Start the DB2 instance by running **`db2start`** as the instance administrator. If you are using System Automation for Multiplatforms to manage DB2, then the DB2 services should come back automatically if the *nominal status* is *online.*
2. As the DB2 instance administrator, enable HADR by running:

   ```
   db2 start hadr on database DB2_DBNAME as standby
   ```

   Tip: If System Automation for Multiplatforms is installed, it will attempt to start DB2 and may bring up HADR as standby already.

If the site has been down for a long time or has to be completely rebuilt on the failed site, a backup and restore will be necessary to synchronize the databases. HADR will need to be reconfigured on the rebuilt server as well. For more information about this procedure, refer to “DB2 HADR configuration” on page 168.

Application

When the secondary site fails, there should be minimal reconfiguration required from the application level because the primary database is still active. Any cron tasks or services that were running exclusively on the secondary site will need to be brought online on the primary.

Important: Additional application servers may need to be brought online on the second site to handle the user load of all users accessing a single site. Special care should be taken to ensure that the extra load does not crash the remaining servers.

Reporting

If reports have been configured to point to the standby reporting database as explained in Appendix A, “Reporting” on page 219, then they will need to be redirected to the primary database to function correctly. Follow the instructions in the Appendix to reconfigure the reports back to the primary database.

Integrations

There are no specific actions required when the failed site comes back online. If using SIB as the JMS provider, the messages stuck in the datastore will continue to process. If using WebSphere MQ as the JMS provider, another active instance should already be in place on the site that did not fail.

Important: Make sure that only one site has WebSphere MQ multi-instance queue manager online.

Symptoms of failure of a secondary site

The symptoms of a secondary site failure should be less than those of a primary site failure but will vary depending on which services are enabled on this site. Some common symptoms include:

- Performance

  Performance may be impacted when a site fails because the full user load will now be directed to one site. It is important that both sites can handle the extra processing without overloading the systems.

- Lost user sessions

  When the secondary site fails any users connected to it will lose their sessions. If the load balancer detects a site failure and stops directing traffic to this location, users should be able to start a new session on the primary by reconnecting to the application address.

- Lost or stuck transactions with SIB

  The messages on the failed site will remain on the datastore and will be processed when the site recovers from the failure automatically.

- Reports unavailable

  If reports are configured to run against the reporting database on the secondary site, these reports will no longer run. The data source for reports will need to be configured to point to the primary database. See Appendix A, “Reporting” on page 219. Having to run reports on the primary site can also impact overall system performance.

- Cron task delay

  Cron tasks that do not have the target enabled to the secondary site will recover and start back on the primary after a short delay. Cron tasks running exclusive to the secondary site using the *target enabled* or *do not run* functionality may have to be manually reconfigured on the primary.

5.10 Conclusion

Activating resources on the secondary site can help to distribute some of the load across both sites. Knowing the implications of this topology is important when selecting this configuration. Understanding how failures can affect each site can help design an effective disaster recovery plan.

Part 3

Appendixes

A

Reporting

This appendix provides details about how to configure BIRT reports within IBM SmartCloud Control Desk to run against a secondary or replicated database.

Business Intelligence and Reporting (BIRT) is an embedded reporting tool in IBM SmartCloud Control Desk. BIRT enables insight into the most detailed levels of integration and provides a seamless integration for the users.

BIRT provides functions such as, for example, print, email reports, schedule reports, reports usage, and monitoring. Users can generate ad-hoc reports, create their own reports by selecting fields, sorting, grouping and filtering the records. Users can also create their own local queries. These reports and queries can be shared with other users, scheduled, and edited to meet business needs.

In this appendix we take a look at the following:

- Overview
- Configuring BIRT to execute against the reporting database
- Configuring BIRT Report Only Server

Overview

Many organizations often request the Business Intelligence and Reporting (BIRT) reports to be executed from a separate reporting database. The reporting database may be a snapshot of the transactional primary database or an up-to-the-minute copy using replication techniques. Using a separate reporting database can reduce the load on the live primary database, and enhance performance for the users. This performance improvement is significant when executing complex reports, which can use up a lot of processor cycles.

This appendix provides details on how to configure BIRT reports to be executed against a secondary database. Additional details are provided about the procedure to separate the BIRT reporting function from the user interface JVM, which can further enhance performance.

Based on the business needs, organizations may want all the reports to execute using a separate reporting database for performance enhancements. In this scenario the IBM SmartCloud Control Desk application is connected to the primary production database, and the BIRT reporting engine is connected to the secondary database. Out-of-the-box the BIRT reports use the default database value defined in the `maximo.properties` file to connect to the database. This default data source needs to be updated to the new reporting database.

Figure A-1 on page 221 displays the typical BIRT engine pointing to the secondary reporting database. In the example, the application JVMs for the user interface, the integration framework, and cron all execute transactions against the primary production database. The BIRT Report Only Server (BROS) is configured to execute reports against the secondary reporting database. Depending on the organization's requirements, the secondary reporting database can be an exact mirror image of the primary database or it can be synchronized at a regular interval. In our topology, a DB2 read-only HADR server was deployed as the reporting database, which maintains an almost exact copy of the primary database at all times. For more information about the DB2 HADR setup, refer to "HADR setup" on page 63.

Important: Query Based Reports (QBR) execute against the primary database, even though a specific reporting database is set up.

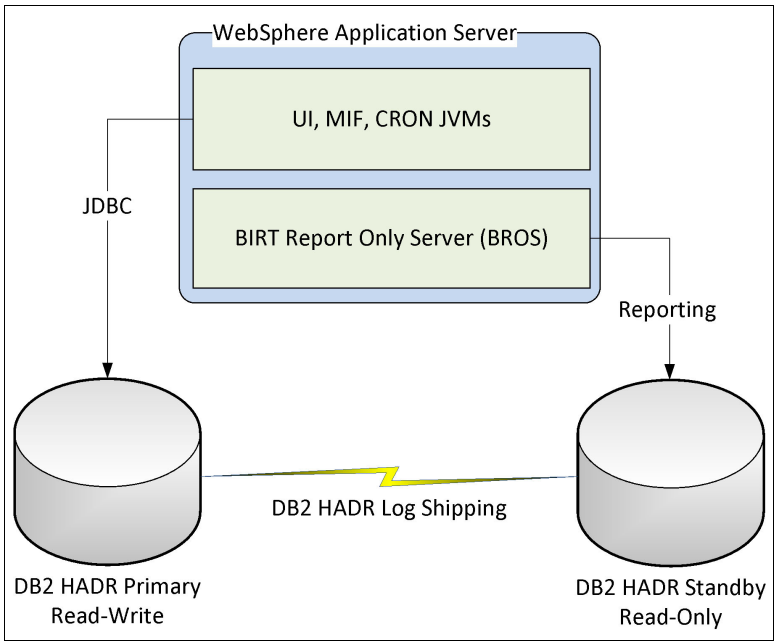

Figure A-1 BIRT reporting engine with secondary reporting database

Configuring BIRT to execute against the reporting database

To enable the BIRT reporting engine to execute reports from the secondary reporting database, execute the following steps:

1. Launch the IBM SmartCloud Control Desk application and log in with administrator user.
2. Navigate to the Report Administration application. **Go to** → **Administration** → **Reporting** → **Report Administration**.
3. Click **Select Action** → **Configure Data Sources** → **New Row**.
4. Fill in the values for the fields based on your reporting database setup. The Data Source Name case and value have to be exactly the same as shown in Figure A-2 on page 222.

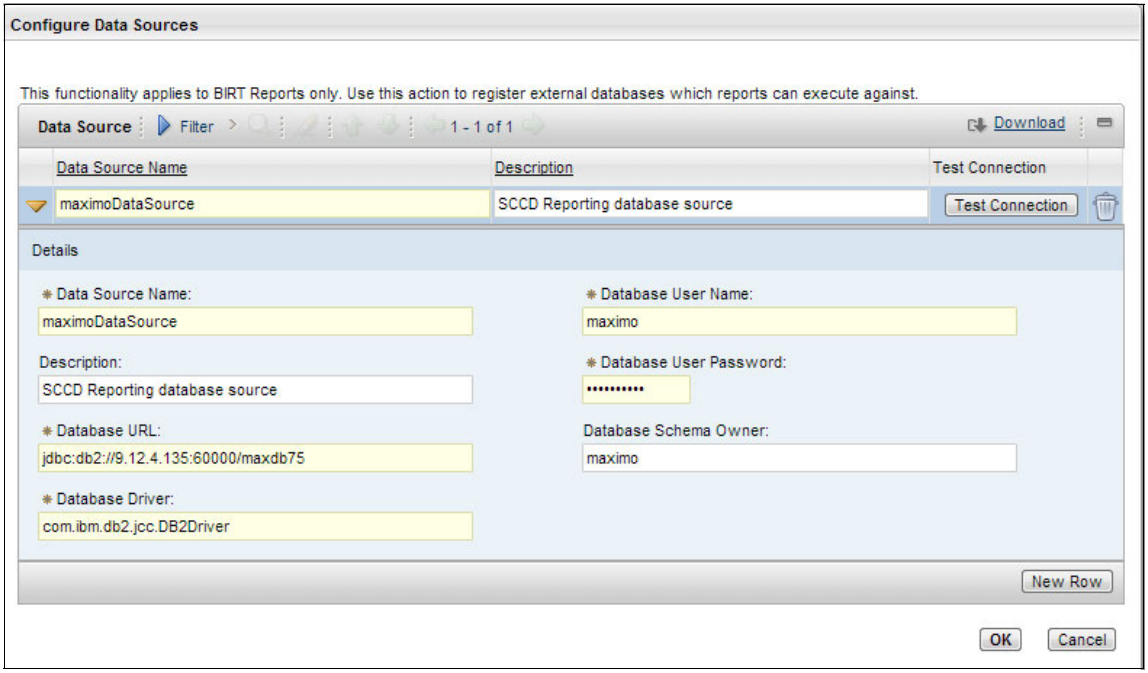

Figure A-2 Reporting database data source setup

5. Click **Test Connection** to verify the connection to the reporting database.
6. Click **OK** to save the data source information.
7. Stop and restart the application server. Execute the report to confirm that it uses the data from the reporting database.
8. After the application server is restarted, execute the following command:

   ```
   db2 "list applications show detail" | more
   ```

   After the application server is restarted, you should see DB2 jdbc connections from the application server using the `maximo` id.
9. In case the reporting database is unavailable, change the database address in step 4 to revert all reports to the primary database.

Configuring portions of reports to execute against multiple sources

Some organizations may want portions of their reports to execute against a separate reporting database. This may be required for a complex report or for reports that are not relying on the frequently updated transactional data. Some reports may execute against the production transactional database, while other reports may execute against the reporting database.

To use multiple data sources with reports, in our case, we assumed that the BIRT designer version 3.7.1 was installed on the client workstation and report source code was available for modification. A local IBM SmartCloud Control Desk with the report source files needed to be available as well. The following steps describe how to configure reports to execute against the primary production transactional database and an external reporting database.

1. In case of two data sources, you need to update the address for the `maximoDataSource` to point to the primary transactional database.
2. Add a second database source, which points to the secondary, or external reporting database. Use the steps described to add a new data source. Figure A-3 displays the BIRT reporting setup with two different data sources.

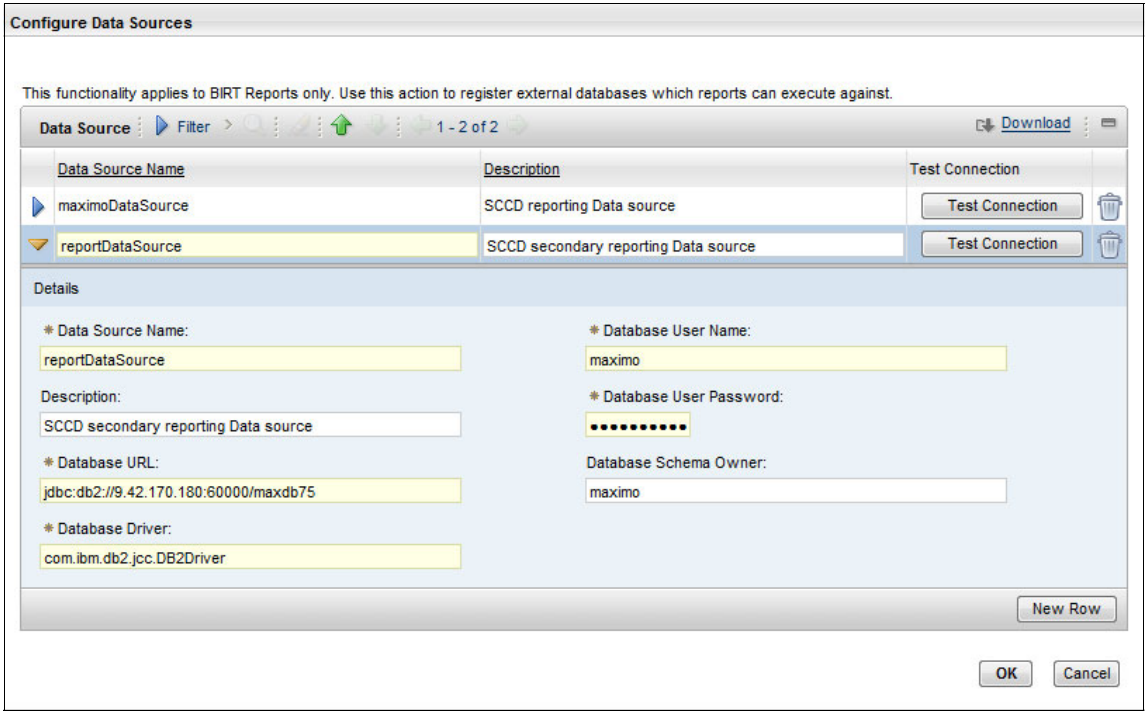

Figure A-3 BIRT reporting with two data sources

3. Locate the compiled class used for the application report scripting from the path `/opt/IBM/SMP/maximo/reports/birt/scriptlibrary/classes`.
4. Navigate to the BIRT designer folder. Note that these are samples using BIRT designer 3.7.1

   `\birt-report-designer-all-in-one-3_7_1\eclipse\plugins\org.eclipse.birt.report.viewer_3.7.1.v20110905\birt\WEB-INF`. Copy the entire classes folder from step 3 to this Eclipse directory.

5. Navigate to `\birt-report-designer-all-in-one-3_7_1\eclipse\plugins\org.eclipse.birt.report.viewer_3.7.1.v20110905\birt\WEB-INF\classes`. Edit the `mxreportdatasource.properties` file. The first entry, `maximoDataSource`, is the default entry. By default it points to the production database for the IBM SmartCloud Control Desk application.
6. Add the data source for the reportDataSource. The name and case of the datasource should be spelled exactly the same way in which it was entered in step 2. Example A-1 lists the definition of the datasource.

Example A-1 Reporting datasource with two different databases

```
maximoDataSource.url=jdbc.db2://9.12.4.167:60000/maxdb75
maximoDataSource.driver=com.ibm.db2.jcc.DB2Driver
maximoDataSource.username=maximo
maximoDataSource.password=xyzpwd
maximoDataSource.schemaowner=maximo

reportDataSource.url=jdbc.db2://9.42.170.180:60000/maxdb75
reportDataSource.driver=com.ibm.db2.jcc.DB2Driver
reportDataSource.username=maximo
reportDataSource.password=xyzpwd
reportDataSource.schemaowner=maximo
```

7. If the database drivers for the databases are not loaded, they can be copied from `/opt/IBM/SMP/maximo/applications/maximo/lib` to `\birt-report-designer-all-in-one-3_7_1\eclipse\plugins\org.eclipse.birt.report.viewer_3.7.1.v20110905\birt\WEB-INF\lib`.
8. Copy the report design file and rename it. Launch the BIRT designer and open the report design file.
9. The default datasource is displayed under Data sources. Highlight the datasource and click the **XML Source** to modify the datasource name. Figure A-4 on page 225 displays the BIRT designer data source.

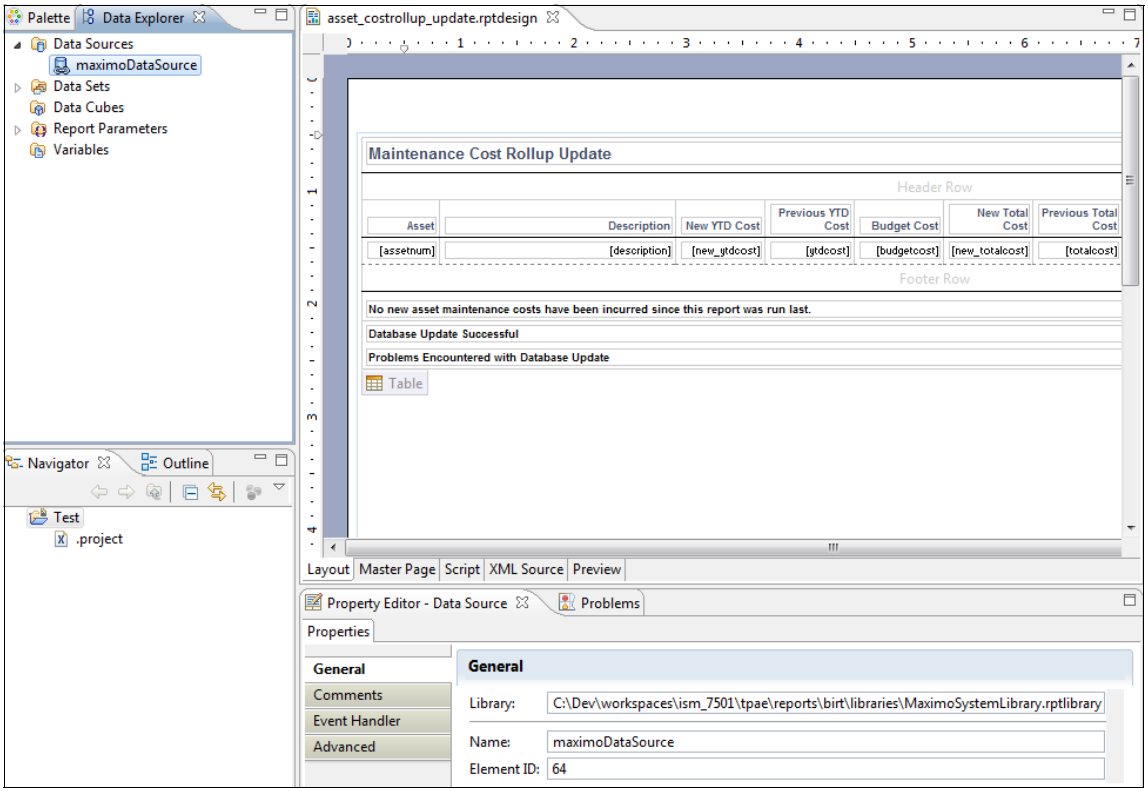

Figure A-4 BIRT designer data source

10. From the top of the XML file, search for the value `maximoDataSource`. The following line will be displayed:

```
<script-data-source name="maximoDataSource" id="64"
extends="MaximoSystemLibrary.maximoDataSource"/>
```

11. Update `maximoDataSource` to `reportDataSource`. The line should look like this:

```
<script-data-source name="reportDataSource" id="64"
extends="MaximoSystemLibrary.maximoDataSource"/>
```

12. Continue to search for `maximoDataSource`. Update all occurrences of the line:

```
<property name="dataSource">maximoDataSource</property>
```

13. The updated line will reflect the new data source:

```
<property name="dataSource">reportDataSource</property>
```

14. After all the occurrences of the original data source are updated to the new one, go back to the layout and save the report design file.

15. Launch the IBM SmartCloud Control Desk application. Log in as a user who has the rights to update reports. Navigate to Reports administration window. **Go To** → **Administration** → **Reports Administration**.
16. Open the report that you want to re-import after the updates from step 14. Click **Select Action** → **Duplicate Report** to keep a copy of the old report.
17. Click **Select Action** → **Import Report**, select the updated report design file and import the report by clicking **OK**. Figure A-5 displays the Import Report dialog for the BIRT report design file.

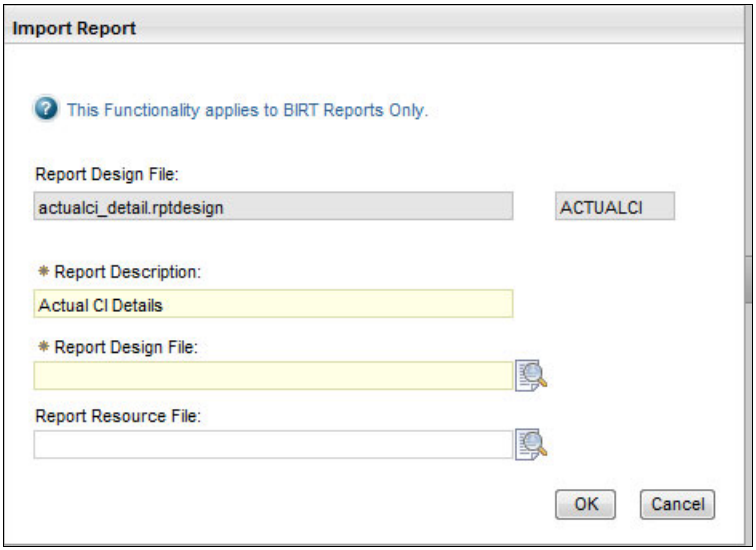

Figure A-5 Import BIRT report design file

18. Next, click **Generate Request Page** to regenerate the report file.
19. Repeat steps 8-18 for updating the data source on the reports that are required to be executed from the secondary reporting database.

Be careful with update reports: There are some out-of-the-box BIRT reports such as Asset Cost Rollup, Inventory Analysis Update, and so on, which update the database when executed. Carefully consider if you want these update reports to execute against your production or transactional database.

Configuring BIRT Report Only Server

In IBM SmartCloud Control Desk, you have an option to configure the environment to include a BIRT Report Only Server (BROS). This server enables you to offload report processing requirements to a different server. Enabling BROS can balance report load processing and improve the overall system performance. The BIRT reports can now be executed from the separate BROS JVM, thus enhancing the UI JVM performance for the users. The BROS is utilized for report processing regardless of what clustered server the users may be on.

Important: Certain report functionality continues to execute from the UI server, even if a BROS server is configured:

- Direct print reports.
- Direct print with attachment reports.
- Query Based Reports (QBR) or ad-hoc reports, as they are being created. Once a QBR is saved, it executes from the BROS if configured.

Configuring a BROS cluster

Extend the IBM SmartCloud Control Desk application cluster to include BROS. This can offload all the reporting from the other clusters and improve performance. Follow the steps below to create a BROS cluster.

1. Create a new cluster SCCDBROS. The new cluster should be configured the same as the SCCDUI cluster. Refer to 3.5.10, “Cluster configuration” on page 56 steps 1-10 for more details on how to create a BROS cluster.
2. Next, split the deployment file for a separate reporting cluster. The steps are the same as the SCCDUI cluster separation tasks. Refer to 3.7.2, “Split deployment files” on page 92 for information about how to split the deployment files.
3. Modify the following property in the `maximo.properties` file for BROS. Update the value to 0. This enables scheduled reports to run from the BROS cluster.

   ```
   mxe.report.birt.disablequeuemanager=0
   ```

4. Modify the `deployment-application.xml` for the BROS cluster as shown in the bolded terms in Example A-2.

   Example A-2 Update deployment-application.xml for BROS

   ```
   ...
   <description>SCCDBROS</description>
   <display-name>SCCDBROS</display-name>
   ```

...

5. Modify the context root in the `web.xml` file:

   `<context-root>/maximobros</context-root>`

6. Build the BROS ear file. Refer to step12 on page 95 for the details about building the ear file.
7. Next, deploy the ear file to the BROS cluster. Follow the steps for deploying the UI ear as explained in 3.7.3, “Ear file deployment on WebSphere Application Server” on page 100.
8. Map `SCCDBROS` and all web servers to all ear modules.
9. Launch the IBM SmartCloud Control Desk application and log in using the administrator user.
10. Navigate to the System Properties application. Click **Go To** → **System Configuration** → **Platform Configuration** → **System Properties**.
11. Click **filter** and search on `viewerurl`. Modify the Property Name with `mxe.report.birt.viewerurl` and save the record, as shown in Figure A-6.

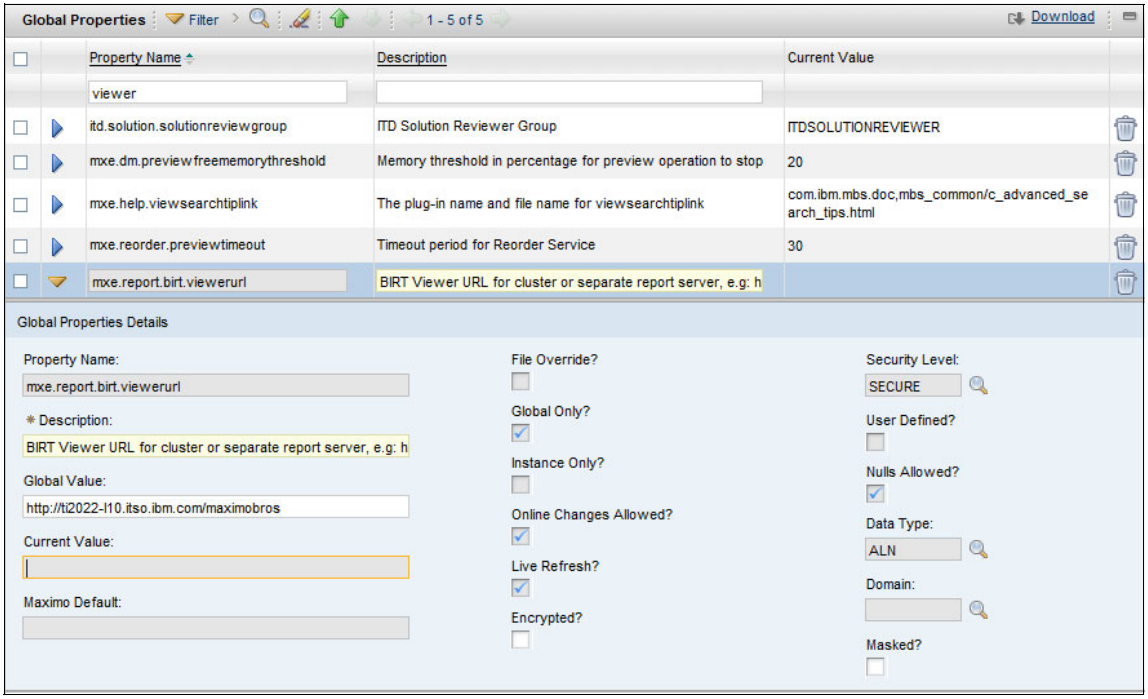

Figure A-6 BROS system property update for the viewerurl

12. Launch the WebSphere Administration console and navigate to the web container cookies at **Application Server** → **MXBrosServer** → **Web containers** → **Cookies**. Modify the cookie value to a distinct unique value that is not used for any other application server in this environment. This will prevent the users from being logged out from their UI session when they launch a report and close the report window. Figure A-7 displays the appropriate dialog where we changed the Cookie name to JSESSIONBIRTID.

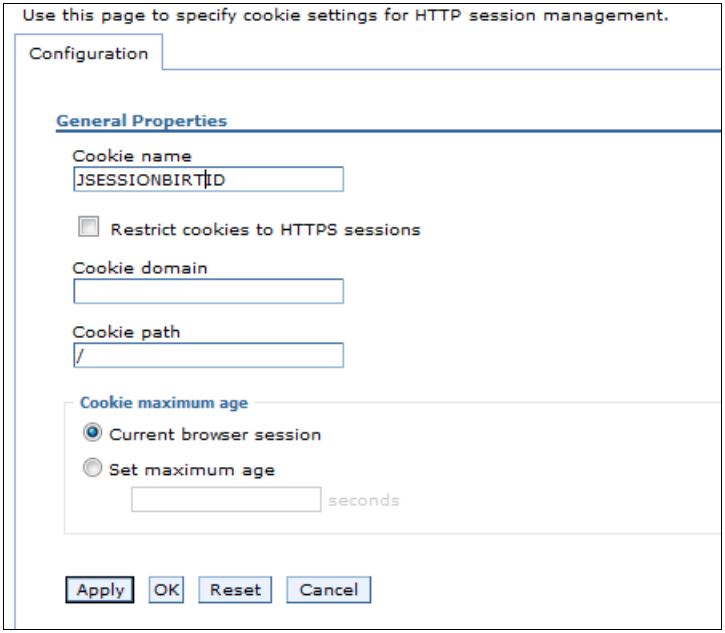

Figure A-7 JSESSIONID cookie setting for BROS JVM

13. Click **Apply** and synchronize the node with the updates. Follow step 12 for all the BIRT JVMs in the cluster.
14. Stop and restart the BIRT cluster. When the user executes the reports from their UI session, it will transfer the execution of the report to the BROS and run in a separate window. The report can be closed by closing the report window. The UI will not be affected.

DB2 HADR restrictions

There are a few restrictions when using DB2 HADR that must be taken into consideration when designing the read only HADR in connection with BIRT reporting:

1. Large object fields (LOB) that are larger than 1 GB cannot be logged, so those are not replicated. The LOB fields come in a default configuration of 1 GB. If these fields are expanded, they will not be replicated.
2. XML and large object (LOB) data must be inline to be successfully queried, otherwise an error is returned (SQL1773N Reason Code 3). Most of the LOB fields in the Maximo database have a limit of 32,000 bytes, but the table definition can allow up to 1 GB of data. To ensure that the data is inline, run the commands shown in Example A-3.

   Example A-3 Sample sql to check whether the LOB data is inline

   ```
   select commlogid, ADMIN_IS_INLINED(<LOB Column>) as IS_INLINED,
   ADMIN_EST_INLINE_LENGTH(<LOB Column>) as EST_INLINE_LENGTH from
   maximo.<Table> where ADMIN_IS_INLINED(<LOB Column>) = 0
   ```

3. Alter the table to set the inline limit on the LOB field to match the limit in Maximo, shown in Example A-4. Repeat these commands for all the LOB columns in the table and run a **reorg** command on those tables.

   Example A-4 Command to make LOB columns inline

   ```
   alter table maximo.<Table> alter column <LOB Column> set INLINE
   LENGTH 32768
   db2 reorg table maximo.<Table> LONGLOBDATA
   ```

4. Rerun the sql in step 2 and verify that all the LOB columns are inline.

This concludes the setup of the BIRT Report Only Server. By changing the reports to use the secondary reporting database and BROS, the load can be shifted from the UI JVMs and primary database and enhance performance.

B

Integration Composer

This appendix provides information on the Integration Composer integration with IBM SmartCloud Control Desk and how it is affected in case of disaster recovery.

Integration Composer is an integration tool that imports hardware and software inventory data from external data sources into the Maximo database tables for deployed assets and configuration items. With Integration Composer, an organization can aggregate data collected by discovery tools and integrate it, creating a central repository for enterprise IT asset management, reporting, and decision support.

Before you import data from an external data source into the Maximo target database, use Integration Composer to create a mapping to transform data from the source format to the target format. A mapping is a set of expressions that tell Integration Composer how to create data in the target using information from a source. For each property that you want to import, define an expression that specifies how to transform the data for that property when Integration Composer imports the data from the source into the target. When you execute a mapping, Integration Composer transforms the collected data and imports it into the target.

When you first implement IT asset management, you can also use Integration Composer's asset initialization adapter to create a baseline set of authorized IT asset records from the deployed asset data that you imported. Authorized IT asset data is managed in the Assets application.

Integration Composer connects to data sources using either Java Database Connectivity (JDBC) technology-enabled drivers or an application programming interface (API).

Overview

Integration Composer is separately installed software that is required by the integration adapter. To use the integration adapter, first you have to install and understand the basics about Integration Composer.

Integration Composer is an integration tool which is used to transform and import inventory data about deployed hardware and software. The data is imported from a discovery or system management tool database into the Maximo database tables for deployed assets. With Integration Composer, an organization can aggregate data collected by external discovery tools and integrate it into the Maximo database, creating a central repository for enterprise IT asset management, reporting and decision support. The Maximo database is the repository used by SmartCloud Control Desk. Integration Composer is the only supported integration tool for importing or updating deployed assets.

Integration Composer is used to import hardware and software inventory data from a discovery tool database into the Deployed Asset, Actual CI, or (for the purposes of asset initialization) Asset tables in the Maximo database. The import or export of data into or out of other tables within the Maximo database is accomplished using a different tool, the integration framework.

Integration Composer consists of a user interface, a command line interface, an engine for processing mapping expressions, connection drivers, and a repository that is a subset of the Maximo database where your information technology data is imported.

Integration Composer includes the IBM Configuration Discovery and Tracking API, which is used only by the Integration Adapter for IBM Tivoli Application Dependency Discovery Manager (TADDM) that is provided with IBM Tivoli Change and Configuration Management Database. Integration Composer uses a JDBC driver or an application programming interface (API) to establish connections to the source data location and target database. For more information about Integration Composer refer to

http://pic.dhe.ibm.com/infocenter/tivihelp/v50r1/topic/com.ibm.tusc.doc/int_comp/c_ctr_ic_overview.html

Disaster Recovery consideration

While Integration Composer is an important integration to IBM SmartCloud Control Desk, it is not mandatory for functioning of IBM SmartCloud Control Desk. Integration Composer is used to import hardware and software inventory data. It is important to keep the inventory data current in the database for providing efficient service. Care should be taken to properly back up the Integration Composer server. In the disaster recovery topology, provisions should be made for deployment of Integration Composer server on both sites.

Integration Composer server can be a stand-alone server on both the sites. The data can be synchronized by taking snapshots of the data on the primary server and restoring the data on the secondary server. If the disk mirroring topology is deployed, then the data can be synchronized by mirroring the disks on which the Integration Composer directory structure resides.

Since Integration Composer uses JDBC drivers to establish the connection to the target IBM SmartCloud Control Desk database, the JDBC connection string within the data source needs to be modified on the secondary site to point to the database server on that site. In a failover scenario, the Integration Composer may need to be modified to integrate with a discovery tool on the secondary site. In some cases during the failover the mappings may have to be recreated.

This concludes the discussion of Integration Composer integration with IBM SmartCloud Control Desk. Integration Composer would be the last component to recover after the IBM SmartCloud Control Desk is operational in case of failover to the secondary site.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *End-to-end Automation with IBM Tivoli System Automation for Multiplatforms*, SG24-7117
- *IBM System Storage DS8000 Copy Services for Open Systems*, SG24-6788
- *IBM XIV Storage System: Copy Services and Migration*, SG24-7759
- *IBM System Storage DS Storage Manager Copy Services Guide*, SG24-7822
- *SAN Volume Controller and Storwize V7000 Replication Family Services*, SG24-7574

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other information

The following information sources may provide additional material of value.

- Implementing highly available systems with IBM Maximo:

  http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_ctr_high_availability.html

- IBM SmartCloud Control Desk, Version 7.5 product documentation Info Center:

  http://pic.dhe.ibm.com/infocenter/tivihelp/v50r1/index.jsp?topic=%2Fcom.ibm.tusc.doc%2Fic-homepage.html

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

(0.2”spine)
0.17”<->0.473”
90<->249 pages

High Availability and Disaster Recovery Configurations

for IBM SmartCloud Control Desk and IBM Maximo Products

Learn how to set up high availability and disaster recovery configuration options

Design a multisite deployment with load balancing

Configure middleware components

In today's global environment, more and more organizations need to reduce their downtime to the minimum possible and look for continuous availability of their systems. Products based on the IBM Tivoli Process Automation Engine (TPAE), such as IBM Maximo Asset Management, Maximo Industry Solutions, and IBM SmartCloud Control Desk, often play a role in such environments and thus also have continuous availability requirements. As part of that, it is important to understand the High Availability (HA) and Disaster Recovery (DR) capabilities of IBM SmartCloud Control Desk and IBM Maximo Products, and how to assure that all the components of an HA/DR solution are properly configured and tested to handle outages. By outlining some of the topologies we have tested, and the documentation we created, we hope to demonstrate how robust the IBM SmartCloud Control Desk and IBM Maximo infrastructure can be.

This IBM Redbooks publication covers alternative topologies for implementing IBM SmartCloud Control Desk and IBM Maximo in High Availability and Disaster Recovery configurations.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-8109-00 ISBN 073843776X