

Design and Deploy WebSphere Applications for Planned, Unplanned Database Downtimes and Runtime Load Balancing with UCP

In Oracle Database RAC and Active Data Guard environments

ORACLE WHITE PAPER AUGUST 2015

Table of Contents

Introduction	3
Issues to be addressed	3
Oracle Database 12c High-Availability and Load Balancing Concepts	4
Configure WebSphere for UCP	4
Create a New JDBC Provider	4
Create a New DataSource	8
Create a JNDI context in the servlet	19
Create a web.xml for the Servlet	20
WebSphere Tips	20
Hiding Planned Maintenance from WebSphere Applications	20
Developer or Web Applications Steps	21
DBA or RDBMS Steps	22
Hiding Unplanned Database Downtime from WebSphere applications	23
Developer or Web Application Steps	23
DBA or RDBMS Steps	24
Runtime Load Balancing (RLB) with WebSphere Servlets	24
Developer or Web Application steps	25
Appendix	26
Enable JDBC & UCP logging for debugging	26
Conclusion	27

Introduction

Achieving maximum application uptime without interruptions is a critical business requirement. There are a number of requirements such as outage detection, transparent planned maintenance, and work load balancing that influence application availability and performance. The purpose of this paper is to help Java Web applications deployed with IBM WebSphere, achieve maximum availability and scalability when using Oracle.

Are you looking for best practices to hide your web applications from database outages? Are you looking at, smooth & stress-free maintenances of your web applications? Are you looking at leveraging Oracle Database's runtime load balancing in your WebSphere applications? This paper covers the configuration of your database and WebSphere Servlets for resiliency to planned, unplanned database outages and dynamic balancing of the workload across database instances, using RAC, ADG, GDS¹, and UCP.

Issues to be addressed

The key issues that impede continuous application availability and performance are:

- » Planned Maintenance:
 - » Achieve transparent maintenance: Make the maintenance process fast and transparent to applications for continuous availability.
 - » Session Draining: When the targeted instance is brought down for maintenance, ensure that all work completes. We will describe how to drain sessions without impacting in-flight work and also avoid logon storms on active instance(s) during the planned maintenance.
- » Unplanned Downtimes:
 - » **Outage detection:** Web application's timeouts are unpredictable and unreliable. This paper describes how to configure WebSphere Servlets to be notified of outages as fast as possible.
 - » Error handling: Several types of SQL exceptions may be received by your Servlets; how to determine that such errors are indicative of database service failure?
 - » Recovery with Response Time Targets: Upon outage the Oracle Database RAC system needs a short period of time to recover before becoming fully operational again. How to react quickly and keep such "brownout" period under SLA targets?
 - » Outcome of in-flight work: Have you ever paid twice for books, flights or taxes? Making a reliable determination of the outcome of the in-flight transaction in the face of database outages was a challenge until Oracle Database 12c. We will describe, how to design Servlets and configure Oracle Database 12c for solving this challenge.
 - » Continuation of in-flight work: How to design Servlets and configure Oracle Database 12c and UCP to allow safe and transparent replay of in-flight transactions in the event of unplanned database outages.
- » Workload Balancing: In RAC, RAC ONE and ADG environments, connection requests are by default distributed randomly by the Net Listener. How to configure your web applications and configure the database for optimal distribution of the workload when the node/services are added/removed?

¹ http://www.oracle.com/technetwork/database/availability/maa-consolidation-2186395.pdf

The paper provides step by step instruction on how to configure JDBC driver, UCP as WebSphere data source and enable high availability properties thereby enabling your applications for planned database maintenance and unplanned database downtimes. Finally the paper discusses the recommended solutions.

Oracle Database 12c High-Availability and Load Balancing Concepts

To support high-availability and load balancing solutions, Oracle Database 12c and prior releases furnish HA configurations (RAC, Data Guard) and features which are leveraged by Oracle Database drivers (e.g., Oracle JDBC) and connection pools (e.g., UCP). This paper will refer to the following features, mechanisms, and concepts described in **Java Programming** with Oracle Database 12c RAC and Active Data Guard² white paper:

- » Universal Connection Pool (UCP)
- » Fast Application Notification (FAN)
- » Oracle Notification Service (ONS)
- » Fast Connection Failover (FCF)
- » Logical Transaction ID (LTXID)
- » Database Request
- » Recoverable Errors
- » Mutable Functions
- » Transaction Guard (TG)
- » Application Continuity (AC)

Configure WebSphere for UCP

Universal Connection Pool (UCP) has the built in support for planned maintenance, unplanned downtimes and runtime load balancing. UCP along with RAC, RAC One and ADG is a tested and certified combination for handling database failovers. UCP has been successfully used by many customers to handle failovers seamlessly. Configuring UCP in IBM WebSphere is explained in detail, hereafter.

Deploying a servlet which accesses Oracle Database through Oracle JDBC driver and Oracle Universal Connection Pool (UCP) in a WebSphere application container requires the following steps:

- » Create a New JDBC Provider
- » Create a New Data Source
- » Create a JNDI lookup in the servlet
- » Create a web.xml for the Servlet

Please note that **WebSphere Application Server version 8.5.5.3** is used in our testing and here are the step by step instructions. Please also, refer to "WebSphere Tips" section of the white paper while using IBM WebSphere.

Create a New JDBC Provider

Define $\{ORACLE_JDBC_DRIVER_PATH\}$ at a location where the Oracle JDBC driver & related libraries are placed. Check Environment \rightarrow WebSphere variables to define the driver's path as $\{ORACLE_JDBC_DRIVER_PATH\}$.

² http://www-content.oracle.com/technetwork/database/application-development/12c-ha-concepts-2408080.pdf

Add a New JDBC Provider: (Refer to Fig 1) Navigate to Resources → JDBC → JDBC Providers Click New to add a New JDBC Provider Step 1: Create a new JDBC provider (Refer to Fig 1.1) Scope: Select the required scope from the drop down menu Database type: Select 'Oracle' from the drop down menu Provider type: Select 'Oracle JDBC Driver UCP' from the drop down menu Implementation type: Select 'Connection pool data source' from the drop down menu Name: This gets auto filled as 'Oracle JDBC Driver UCP' Description: Provide any description Step 2: Enter the database class path information (Refer to Fig 1.2) classpath: Specify the CLASSPATH for ojdbc7.jar, ucp.jar & ons.jar. Use jar files from the same database version Eg: \${ORACLE_JDBC_DRIVER_PATH}/ojdbc7.jar. Please note the significance of each library. ojdbc7_g.jar or ojdbc7.jar \rightarrow JDBC driver with or without debug. ucp.jar → Required for using UCP ons.jar → Required for listening to FAN events. Directory location: Mention the path where the above jar files are placed. Step 3: Summary (Refer to Fig 1.3) Implementation Class Name: Please note that IBM WebSphere correctly chooses and sets the Implementation class as 'oracle.ucp.jdbc.PoolDataSourceImpl' based on the selections in Step 1. PLEASE DO NOT CHANGE THIS. Changing this to any other value will cause connecting to the database to fail. Click FINISH to confirm all the changes.

Refer to Fig 1.4 to check the settings after completing all 3 steps above

Fig 1: Add a New JDBC Provider

	JDI	BC prov	viders		2 =	
Welcome		1000				
Guided Activities		JDBC b	providers			
		Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn				
		more about this task in a <u>quided activity</u> . A guided activity provides a list of task steps and more general				
Services		Information about the topic.				
Resources		Scope: Cell=slc03rznNode01Cell, Node=slc03rznNode01, Server=server1				
 Schedulers Object pool managers JMS JDBC JDBC providers Data sources 	Ш	H Drof	Scope specifies the level a detailed information on w <u>settings help.</u> Node=slc03rznNode01, S	at which the resource definition is visible. That scope is and how it works, <u>see the sco</u> Gerver=server1 v	For	
Data sources (WebSphere Application Server V4)		New	Delete			
Resource Adapters						
Cache instances		Select	Name 🗘	Scope 🗘	Description 🗘	
Mail		You c	an administer the following re	esources:		
ORL Resource Environment			Derby JDBC Provider	Node=slc03rznNode01,Server=server1	Derby embedded non-XA	
Security					JDBC Provider	
			Oracle JDBC Driver UCP	Node=slc03rznNode01,Server=server1	Oracle JDBC Driver UCP	
+ Users and Groups		Total	2			

Fig 1.1 : Create new JDBC provider

Step 1: Create new	Create new JDBC provider
JDBC provider Step 2: Enter database class path information Step 3: Summary	Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values. Scope cells:slc03rznNode01Cell:nodes:slc03rznNode01:servers:server1 * Database type Oracle * Provider type Oracle JDBC Driver UCP * Implementation type Connection pool data source * Name Oracle JDBC Driver UCP Description Oracle JDBC Driver UCP

Fig 1.2: Enter database class path information

Create a new JDBC Provider	
Create a new JDBC Provider	
Step 1: Create new	Enter database class path information
→ Step 2: Enter database class path information Step 3: Summary	Set the class path for the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays a default list of jars and allows you to set the environment variables that define the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).
	Entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). If a value is specified for you, you may click Next to accept the value.
jars from same database version	Class path: \${ORACLE_JDBC_DRIVER_PATH}/ojdbc7.jar \${ORACLE_JDBC_DRIVER_PATH}/ycp.jar \${ORACLE_JDBC_DRIVER_PATH}/gog_jar
	Directory location for "ojdbc6.jar" which is saved as WebSphere variable \${ORACLE_JDBC_DRIVER_PATH} /opt/IBM/Websphere/jdbc/lib
Previous Next Car	ncel

Fig 1.3: Summary

Creat	e a new JDBC Provider		
Cre	eate a new JDBC Provider		
	Step 1: Create new	Summary	
JDBC provider	Summary of actions:		
	database class path	Options	Values
information	Scope	cells:slc03rznNode01Cell:nodes:slc03rznNode01:servers:server1	
→	Step 3: Summary	JDBC provider name	Oracle JDBC Driver
		Description	Oracle JDBC Driver
		Class path	\${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar
		\${ORACLE_JDBC_DRIVER_PATH}	/opt/IBM/Websphere/jdbc/lib
		Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource
	Previous Finish Cancel]	

Fig 1.4: Newly added JDBC provider

<u>JDBC providers</u> > Oracle JDBC Driver UCP

Use this page to edit properties of a Java Database Connectivity (JDBC) provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment.

eneral Properties	Additional Properties
Scope	Data sources
cells:slc03rznNode01Cell:nodes:slc03rznNode01:servers:server1	
Name	
Oracle JDBC Driver UCP	
Description	
Oracle JDBC Driver UCP	
Class path	
\${ORACLE_JDBC_DRIVER_PATH}/ojdbc7.jar	
\${ORACLE_JDBC_DRIVER_PATH}/ycg.jar \${ORACLE_JDBC_DRIVER_PATH}/ong.jar	
\${ORACLE_JDBC_DRIVER_PATH}/uco.jar \${ORACLE_JDBC_DRIVER_PATH}/ons.jar Native library path	
\${ORACLE_DBC_DRIVER_PATH}/uco.jar \${ORACLE_DBC_DRIVER_PATH}/uco.jar Native library path	
\${ORACLE_DB&C_DRIVER_PATH}/uccp.jar \${ORACLE_DB&C_DRIVER_PATH}/uccp.jar Native library path 	
\${ORACLE_DB&C_DRIVER_PATH}/user \${ORACLE_DB&C_DRIVER_PATH}/user Native library path Isolate this resource provider Implementation class name	
<pre>\${ORACLE_JDBC_DRIVER_PATH}/ucp.jar \${ORACLE_JDBC_DRIVER_PATH}/ucp.jar \$ Context and the second of the second</pre>	

Create a New DataSource

A new data source is required for connecting to the Oracle Database. The steps are as highlighted below.

- » Create a New JAAS-J2C Authentication Data
- » Create a New Data Source
- » Verify if WebSphere connection pool is disabled
- » Set Custom Connection Pool Properties i.e., UCP properties
- » Restart the Server after adding a new datasource
- » Test Connection

Each one of these steps is explained in detail with screenshots, hereafter.

Create a New JAAS-J2C Authentication Data (Refer to Fig 2.1 & Fig 2.1.1)

Navigate to Security \rightarrow Global Security \rightarrow Java Authentication and Authorization Service \rightarrow J2C Authentication data

Click New to add a new JAAS-J2C Authentication Data and fill in the following details.

Alias: Choose any appropriate Alias. Such as RAC12c, OracleDB etc.,

User ID: Enter the username of the Oracle Database

Password: Enter the password of the Oracle Database

Refer to Fig 2.1.1 which displays the DB username & password

Create a New Data Source (Refer to Fig 2.2)

Navigate to Resources \rightarrow JDBC \rightarrow Data Sources Click New to add a new Datasource

Step 1: Enter basic data source information (Refer to Fig 2.2.1)

Data source name: Select the appropriate Data source name. E.g., orclDataSource JNDI Name: Please make sure that JNDI name is as mentioned "/jdbc/<datasourcename>" Eg., /jdbc/orclDataSource

Step 2: Select JDBC Provider (Refer to Fig 2.2.2)

Select an existing JDBC provider : Choose the already created JDBC Provider as shown in the screenshot.

Step 3: Enter database specific properties for the data source (Refer to Fig 2.2.3)

URL : Enter the Connect string URL used to connect to the Oracle RAC database. Example:

jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=tcp)(HOST=proddbclusterscan)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=proddb)))

Data store helper class name: Select 'Oracle11g data store helper' from the dropdown menu.

Step 4: Setup security aliases (Refer to Fig 2.2.4)

Component-managed authentication alias: Select the J2C Authentication created as per Fig 2.1 from the dropdown menu.

Mapping-configuration alias: Do not select anything

Container-manager authentication alias: Select the J2C Authentication created as per Fig 2.1 from the dropdown menu

Step 5: Summary (Refer to Fig 2.2.5)

Check all the details to make sure everything is entered correctly and click FINISH

Refer to Fig 2.2.6 to verify the summary of the dataSource anytime.

Verify if WebSphere connection pool is disabled. WEBSPHERE AUTOMATICALLY TAKES CARE OF THIS STEP. The data source will be configured to use UCP with the default settings. The following properties are automatically set on the data source. Do not alter any of these properties. Changing any of these could cause the data source to no longer work properly. Step 1: WebSphere connection pooling is turned off. (Refer to Fig 2.3.1) To verify this, select data source created. Example: orclDataSource Click on Connection pools \rightarrow Maximum connections to see if it is set to 0. Note: Maxconnections =0, indicates that WebSphere connection pooling is turned off. Changing to a value other than zero will cause WebSphere to track the number of connections attempted which conflicts with the number that Oracle UCP is tracking. It is not advisable to change this setting. Step 2: WebSphere prepared statement caching is turned off (Refer to Fig.2.3.2) To verify this, select the data source created. Example:orclDataSource Click on Websphere Application Server data source properties \rightarrow Statement cache size to see if it is set to 0. Note: WebSphere prepared statement caching can only be used when WebSphere connection pooling is turned on. Since, we are using UCP, this should be turned OFF. Step 3: Verify the correct connectionFactoryClassName (Refer to Fig 2.3.3) To check this, select the UCP datasource; e.g., orclDataSource Click on Custom Properties →connectionFactoryClassName, check that it is set to oracle.jdbc.pool.OracleDataSource when you select UCP. Or set it to oracle.jdbc.replay.OracleDataSourceImpl if you want to use use Application Continuity (AC). Note: Setting the connectionFactoryClassName to any other value will throw an exception. Step 4: Custom Property to disable WebSphere connection Pool (Refer to Fig 2.4) disableWASConnectionPooling is set to true, by default. Otherwise, you must explicitly set it to true.as follows: Select the datasource in question; e.g., orclDataSource Click on Custom Properties and create a new property disableWASConnectionPooling; then set it to true

Set Custom UCP Properties such as FCF (Refer to Fig 2.4)

FCF (fastConnectionFailoverEnabled) is an important property which handles failover of instances during both planned and unplanned downtimes. It is mandatory to have this property turned on. For more details on how to form ONSConfiguration string, refer to the Oracle Notification Service (ONS) section of the white paper "Java

Programming with Oracle Database 12c RAC and Active Data Guard ³"

Select the datasource in question e.g.,orclDataSource

Click on Custom Properties then New and add the desired UCP properties shown below.

Property Name	Property Type	Property Details
minPoolSize	java.lang.String	Set the appropriate minimum pool size
maxPoolSize	java.lang.String	Set the appropriate maximum pool size
initialPoolSize	java.lang.String	Should be closer to minPoolSize
fastConnectionFailoverEnabled	java.lang.Boolean	Required. Set it to TRUE
disableWASConnectionPooling	java.lang.Boolean	Required. Set it to TRUE
ONSConfiguration	java.lang.String	Optional. Required for pre 12c Oracle Database version

Restart the Server

Refer to 'WebSphere Tips' for more details on restarting the servers.

³ http://www-content.oracle.com/technetwork/database/application-development/12c-ha-concepts-2408080.pdf

> Test Connection (Refer to Fig 2.5)

Select Datasource \rightarrow Test Connection

Fig 2.1: New J2C Authentication Data

Security Configuration Wizard Security Configuration Report	
Administrative security	Authentication
Enable administrative security Administrative user roles	Authentication mechanisms and expiration
Administrative group roles	ITPA
Administrative authentication	Kerberos and LTPA
	Kerberos configuration
Application security	SWAM (deprecated): No authenticated communication between servers
Enable application security	Authentication cache settings
	Web and SIP security
Java 2 security	RMMTOP security
Use Java 2 security to restrict application access to local resources	Java Authentication and Authorization Service
Warn if applications are granted custom permissions	Application logins
Restrict access to resource authentication data	System logins
User account repository	J2C authentication data
Realm name	Enable Java Autoentication SPI (JASPI)
defaultWIMFileBasedRealm	Providers
Current realm definition	Use realm-qualified user names
Federated repositories	
Available realm definitions	Security domains
Federated repositories Configure Set as current	External authorization providers
	Programmatic session cookie configuration

Fig 2.1.1: Set the Database Username/Password

lobal security		
<u>Global security</u> > <u>JAAS - J2C authenticat</u>	tion data > slc03rznNode01/RAC12c	
Specifies a list of user identities and pass	swords for Java(TM) 2 connector security to use.	
General Properties		
* Alias		
slc03rznNode01/RAC12c]
* User ID		
scott		
* Password		
•••••		
Description		
RAC 12.1 DB]	
Apply OK Reset Cancel		

Fig 2.2: Adding a new DataSource

Data sources Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource objection objective settings of a datasource objective setting set					
Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object					
Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a <u>guided activity</u> . A guided activity provides a list of task steps and more general information about the topic. Scope: Cell=slc03rznNode01Cell, Node=slc03rznNode01, Server=server1 Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, <u>see the scope settings help</u> . Node=slc03rznNode01, Server=server1					
New Delete Test connection Manage state					
Select Name 🗘 JNDI name 🗘 Scope 🗘 Provider 🗘 Description 🗘 Category					
You can administer the following resources:					
Default DefaultDatasource Node=slc03rznNode01,Server=server1 Derby Datasource Datasource Datasource Vode=slc03rznNode01,Server=server1 Derby Datasource for the Provider Vode=slc03rznNode01,Server=server1 Derby Datasource Datasource Image: Datasource Vode=slc03rznNode01,Server=server1 Derby Datasource For the Image: Datasource Vode=slc03rznNode01,Server=server1 Derby Derby Default Image: Datasource Vode=slc03rznNode01,Server=server1 Derby Default Default Image: Datasource Vode=slc03rznNode01,Server=server1 Derby Default Default Image: Datasource Vode=slc03rznNode01,Server=server1 Default Default Default					
orclDataSource /jdbc/orclDataSource Node=sic03rznNode01,Server=server1 Oracle JDBC Driver UCP Data source for the Oracle JDBC UCP Driver. WebSphere connection pooling and statement caching are disabled.					
Total 2					

Fig 2.2.1: Enter some basic data source information

÷	Step 1: Enter basic	Enter basic data source information
	data source	
	information	Set the basic configuration values of a datasource for association with your
	Step 2: Select JDBC provider	application server and the database.
		Requirement: Use the Datasources (WebSphere(R) Application Server V4)
	Step 3: Enter database specific properties for the	console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.
	data source	Scope
	Step 4: Setup	cells:slc03rznNode01Cell:nodes:slc03rznNode01:servers:server1
	security aliases	* Data source name
	Step 5: Summary	orclDataSource
		* JNDI name
		/idbc/orclDataSource

Fig 2.2.2: Select JDBC provider

Create a data source		
Create a data source		
Step 1: Enter basic	Select JDBC provider	
information	Specify a JDBC provider to support the datasource. If you choose to create a	
→ Step 2: Select JDBC provider	new JDBC provider, it will be created at the same scope as the datasource. If you are selecting an existing JDBC provider, only those providers at the current scope are available from the list.	
Step 3: Enter database specific properties for the	Create new JDBC provider	
data source	Select an existing JDBC provider	
Step 4: Setup security aliases	Select	
Step 5: Summary	Select Derby JDBC Provider Oracle JDBC DriverNUCP	
Previous Next Car	ncel Select an existing JDBC provider	

Fig 2.2.3: Enter the database specific properties for the data source

Dreate a data source				
Create a data source				
Step 1: Enter basic data source	Enter database specific	properties for the data source		
information				
Step 2: Select JDBC provider	Set these database-specific driver to support the connect	properties, which are required by the database vendor JDBC tions that are managed through the datasource.		
→ Step 3: Enter	Name	Value		
database specific properties for the	* URL	jdbc:oracle:thin:@(DESCRIPTION=(ADD)		
data source	* Data store helper class na	ame		
Step 4: Setup	Oracle11g data store help	per 💌		
security aliases	Use this data source in	container managed persistence (CMP)		
Step 5: Summary				
Previous Next Cancel				

Fig 2.2.4: Set the security aliases

Сг	eate a data source	
	Create a data source	
	Step 1: Enter basic data source	Setup security aliases
	information Step 2: Select JDBC	Select the authentication values for this resource.
	provider	Component-managed authentication alias
	Step 3: Enter database specific properties for the data source	Mapping-configuration alias (none)
	→ Step 4: Setup security aliases	Container-managed authentication alias slc03rznNode01/RAC12c
	Step 5: Summary	Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.
	Previous Next Ca	Security domains
	Previous next Ca	sicer

Fig 2.2.5: Summary

Step 1: Enter basic	Summary	
information	Summary of actions:	
Step 2: Select JDBC	Options	Values
provider	Scope	cells:slc03rznNode01Cell:nodes:slc03rznNode01:servers:server1
Step 3: Enter	Data source name	orclDataSource
properties for the	JNDI name	/jdbc/orclDataSource
data source Step 4: Setup	Select an existing JDBC provider	Oracle JDBC Driver UCP
security aliases	Implementation class name	oracle.ucp.jdbc.PoolDataSourceImpl
Step 5: Summary	URL	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS= (PROTOCOL=tcp)(HOST=proddbcluster-scan)(PORT=1521)))(CONNECT_DATA (SERVICE_NAME=proddb)))
	Data store helper class name	com.ibm.websphere.rsadapter.Oracle11gDataStoreHelper
	Use this data source in container managed persistence (CMP)	true
	Component-managed authentication alias	slc03rznNode01/RAC12c
	Mapping-configuration alias	(none)
	Container-managed authentication alias	slc03rznNode01/RAC12c

Fig 2.2.6: Details of the JDBC Datasource

orcIDataSource		Related Items
NDI name		JAAS - J2C authentication di
dbc/orclDataSource		
Use this data source in conta	iner managed persistence (CMP)	
accription		
)ata source for the Oracle JDBC	UCP Driver, WebSphere connection pooling	
and statement caching are disat	oled.	
ategory		
Data store helper class na	ne	
Select a data store helper	class	
Data store helper classes	provided by WebSphere Application Server	
Oracle11g data store help	ber	
(com.ibm.websphere.r	sadapter.Oracle11gDataStoreHelper)	
Specify a user-defined da	ta store helper	
Enter a package-qualified	data store helper class name	
Security settings		
Select the authentication value	s for this resource.	
Component-managed authen	tication alias	
slc03rznNode01/RAC12c	•	
Mapping-configuration alias		
(none)		
Container-managed autho		
SICOSIZINIOGEOT/RACIZC		
Common and required data	source properties	
Common and required data	source properties Value	
Common and required data	value	

Fig 2.3.1: WebSphere connection pooling is turned off

Data sources > orclDataSource > Connection pools

Use this page to set properties that impact the timing of connection management tasks, which can affect the performance of your application. Consider the default values carefully; your application requirements might warrant changing these values.

Scope		Advanced connection
cells:slc03rznNode01Cell:nodes:slc03	3rznNode01:servers:server1	pool properties
Connection timeout	seconds	Connection pool custom properties
Maximum connections	connections	
Minimum connections	connections	
* Reap time 180	seconds	
+ Unused timeout 1800	seconds	
* Aged timeout 0	seconds	
Purge policy EntirePool		
Arely OK Breet Creed		

Fig 2.3.2: WebSphere prepared statement caching is turned off

<u>Data sources</u> > <u>orclDataSource</u> > WebSphere Application Server data source properties

Use this page to set WebSphere(R) Application Server connection management-specific properties that affect a connection pool.

Stateme	ent cache size	
)		statements
Enab	le multithreaded access d	detection
Enab	le database reauthenticat	tion
Enab	le JMS one-phase optimiz	zation support
Log	missing transaction conte:	xt
Non-	transactional data source	1
Error	detection model	
U	e WebSphere Application	Server exception checking model
0 Us 0 Us	e WebSphere Application WebSphere Application	Server exception checking model Server exception mapping model
0 Us 0 Us	e WebSphere Application WebSphere Application	Server exception checking model Server exception mapping model
© ∪⊴ ● ∪⊴ Conne	e WebSphere Application WebSphere Application Ction validation prope	Server exception checking model Server exception mapping model
© Us ● Us Conne	e WebSphere Application WebSphere Application Ction validation prope	Server exception checking model Server exception mapping model
© Us © Us Conne Va	e WebSphere Application webSphere Application ction validation prope lidate new connections	Server exception checking model Server exception mapping model
Conne	e WebSphere Application webSphere Application ction validation prope lidate new connections mber of retries	9 Server exception checking model 9 Server exception mapping model erties
© Us © Us Conne Va Nu 10 Ref	e WebSphere Application webSphere Application ction validation prope lidate new connections mber of retries	Server exception checking model Server exception mapping model
Us Us	e WebSphere Application webSphere Application ction validation proper- lidate new connections mber of retries 00 atry interval	server exception checking model
Conne	e WebSphere Application webSphere Application ction validation proper- lidate new connections umber of retries 00 etry interval	server exception checking model Server exception mapping model erties seconds
Conne	e WebSphere Application webSphere Application cction validation proper- lidate new connections imber of retries 00 etry interval	a Server exception checking model a Server exception mapping model erties seconds mnections
Conne	e WebSphere Application webSphere Application cction validation proper lidate new connections imber of retries 00 etry interval lidate existing pooled con etry interval	erties seconds
Conne	e WebSphere Application webSphere Application cction validation prope lidate new connections imber of retries oo etry interval	a Server exception checking model a Server exception mapping model erties seconds nections seconds
Conne	se WebSphere Application se WebSphere Application cction validation prope lidate new connections imber of retries 00 etry interval lidate existing pooled con etry interval	a Server exception checking model a Server exception mapping model erties seconds seconds seconds seconds
Us	se WebSphere Application se WebSphere Application cction validation prope lidate new connections imber of retries 00 etry interval lidate existing pooled con etry interval	a Server exception checking model a Server exception mapping model erties seconds nections seconds
Conne	e WebSphere Application e WebSphere Application cction validation prope lidate new connections imber of retries 00 etry interval lidate existing pooled con etry interval	a Server exception checking model a Server exception mapping model erties seconds seconds seconds seconds

Fig 2.3.3: Verify the connectionFactoryClassName

Data sour	rces			2
Data sources > orclDataSource > Custom properties Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.				
± Pre	ferences			
Nev	v Delete			
	0 # \$			
Select	Name 🗘	Value 🗘	Description 🗘	Required 🗘
You c	an administer the following resources:			
	connectionFactoryClassName	oracle.jdbc.pool.OracleDataSource	The class that Oracle UCP will use to create a connection. Do not change this property.	true
	oracleLogFileSizeLimit	0	Oracle10g and beyond: The oracleLogFileSizeLimit specifies the maximum number of bytes to be written to any one file. Property is relevant only if trace file is specified. Default is unlimited	false

Fig 2.4: Enabling FCF

Data sources > orclDataSource > Custom properties

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.

Preferences New... Delete 00 # 9 Select Name 🗘 Value 🗘 Description 🗘 Required 🗘 You can administer the following resources: webSphereDefaultIsolationLevel Specifies a default transaction isolation false level for new connections. Resource References and Access Intents override this value. To configure a default transaction isolation level, use the constants defined by JDBC: 1 (READ UNCOMMITTED), 2 (READ COMMITTED), 4 (REPEATABLE READ), 8 (SERIALIZABLE). webSphereDefaultQueryTimeout Sets a default query timeout, which is the false number of seconds (0 means infinite) that a SQL statement may execute before timing out. This default value is overridden during a JTA transaction if custom property syncQueryTimeoutWithTransactionTimeout is enabled. false Enables the implicit passing of client enableClientInformation false information on database connections. The client information provided on each connection is the same as for the WAS.clientinfo trace group. Whereas the WAS.clientinfo trace is configured on an application server, the enableClientInformation property applies to a data source configuration. nodes=slc06bmu:25227,slc06bmv:25227,slc06bmw:25227 Required only for 11g and Not required in false onsConfiguration 12c which has auto-ONS fastConnectionFailoverEnabled true Required setting false minPoolSize 2 false 50 false maxPoolSize disableWASConnectionPooling true false initialPoolSize 15 false

Page: 2 of 2 Total 29

Fig 2.5: Test the Connection with the Oracle Database

ata sour	ces						2
Pref New	Node=slc03	see the scope set BrznNode01, Serve Test connection	er=server1 Manage state				
Ø	ð 👯 🦃						
Select	Name 🗘	JND)I name 🗘	Scope 🗘	Provider 🗘	Description 🗘	Category 🗘
You ca	an administer th <u>Default Dataso</u>	ne following resour	rces: iaultDatasource	Node=slc03rznNode01,Server=server1	Derby JDBC Provider	Datasource for the WebSphere Default Application	
	orclDataSource	/jdb	bc/orclDataSource	Node=slc03rznNode01,Server=server1	Oracle JDBC Driver UCP	Data source for the Oracle JDBC UCP Driver. WebSphere connection pooling and statement caching are disabled.	
Total	2				1		

Create a JNDI context in the servlet

The following code snippet shows how to get a database connection by referring to the JNDI datasource created in Websphere.

```
PoolDataSource pds = getPoolInstance();
conn = pds.getConnection();
private PoolDataSource getPoolInstance() throws SQLException {
    javax.naming.InitialContext ctx = null;
    javax.sql.DataSource pds = null;
    System.out.println ("Attempting connection..." + DateUtil.now());
    ctx = new javax.naming.InitialContext();
    javax.sql.DataSource ds = (javax.sql.DataSource) ctx
        .lookup("java:comp/env/jdbc/orclDataSource");
    PoolDataSource pds = (PoolDataSource) ds;
    return pds;
  }
```

Create a web.xml for the Servlet

The data source resource reference should also be present in web.xml as illustrated hereafter.

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"</pre>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app 3 0.xsd"
       version="3.0">
       <display-name>test1</display-name>
       <servlet-mapping>
               <servlet-name>com.test1.DemoServlet</servlet-name>
               <url-pattern>/DemoServlet</url-pattern>
       </servlet-mapping>
       <resource-ref>
               <description> Datasource to connect to DB </description>
               <res-ref-name>jdbc/orclDataSource</res-ref-name>
               <res-type>javax.sql.DataSource</res-type>
               <res-auth>Container</res-auth>
       </resource-ref>
</web-app>
```

WebSphere Tips

Refer to this section when you require more details on how to access WebSphere console, start/stop an application server, how to set java system property in the console etc., These tips come handy during application deployment.

Description	Details
WebSphere Administrative	http://localhost:9060/ibm/console/login.do Usually 9060 is the default port where
Console	admin console is accessed.
Startup and shutdown scripts	{WAS_INSTALL_DIR}/IBM/Websphere/AppServer/profiles/ <appservprofilename< td=""></appservprofilename<>
location	>/bin
	Example:/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin/
Start an application server	Start Command :
	./startServer.sh <name of="" server="" the=""> -profileName <appserverprofilename></appserverprofilename></name>
	Example: ./startServer.sh server1 –profileName AppSrv01
Stop an application server	Stop Command:
	./stopServer.sh <name of="" server="" the=""></name>
	Example: ./stopServer.sh server1
Increase the number of	The default number of threads in Websphere will be 10. If you want to change
threads	this, go to Servers \rightarrow WebSphere application servers \rightarrow <server name=""> \rightarrow Thread</server>
	Pools \rightarrow Default. Change the Maximum size to the required value (Eg. 50)
Setting up a System Property	Servers \rightarrow WebSphere Application servers \rightarrow <servername> \rightarrow "Java & Process</servername>
	Management" (Process Definition) \rightarrow Java Virtual Machine \rightarrow Custom Properties
	Add any JVM system property required.
Check if ONS is running or	Make sure to add \$ORACLE_CONFIG_PATH to the path where ONS is running.
configured	Environment → WebSphere variables → (Add ORACLE_CONFIG_HOME)

Hiding Planned Maintenance from WebSphere Applications

For maintenance purposes (e.g., software upgrades), the Oracle Database instances can be gracefully shutdown one or several at a time without disrupting the operations and availability of the Web applications. Upon FAN **DOWN** event⁴, UCP drains sessions away from the instance(s) targeted for maintenance. What is the configuration of Web applications and the database to achieve session draining at service stop or relocation? In a nutshell, the procedure consists in stopping non-

⁴ status=down reason=user

singleton services running on the target database instance or relocating singleton services from the target instance to a new instance.

Developer or Web Applications Steps

To hide the planned database maintenance, Web applications need to:

(i) enable Fast Connection Failover (FCF) as mentioned above. Please refer to "Fig 2.4: Enabling FCF" for more details. FCF can also be enabled programmatically as illustrated hereafter;

PoolDataSource pds = new PoolDataSourceFactory.getPoolDataSource();
// not required with auto-ONS in 12c
pds.setONSConfiguration("nodes=<RACNode1>:<port1>,<RACNode2>:<port2>,<RACNode3>:port3");
pds.setFastConnectionFailoverEnabled(true);

(ii) check that ons.jar is in the classpath.

(iii) In addition, with release 12.1.0.2, UCP introduces PlannedDrainingPeriod, a new system property which allows a graceful draining period. It can be specified as a JDK system property (i.e., using -D)

-Doracle.ucp.PlannedDrainingPeriod=30

In IBM WebSphere, the JVM system property can be set as follows. (Refer to Fig.3) Servers \rightarrow WebSphere application servers \rightarrow <servername> \rightarrow Java and Process Management (Process Definition) \rightarrow Java Virtual Machine \rightarrow Custom properties

Fig 3: Setting PlannedDrainingPeriod as System property

pplication servers						
Applica	Application servers > server1 > Process definition > Java Virtual Machine > Custom properties					
Use thi configu	Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can set internal system configuration properties.					
🗄 Pref	erences					
New	Delete					
Select	Name 🗘	Value 🗘	Description 🗘			
You ca	an administer the following resources:					
	<u>com.ibm.security.jqss.debuq</u>	off				
	com.ibm.security.krb5.Krb5Debug	off				
	java.util.logging.config.file /opt/IBM/WebSphere/AppServer_1/profiles /AppSrv01/logs/server1/Oracletrace /oracletrace.properties					
	java.util.logging.configureByLoggingPropertiesFile	true				
	oracle.idbc.Trace	true				
	oracle.ucp.PlannedDrainingPeriod	30				
Total	6					

DBA or RDBMS Steps

DBAs should perform the following steps⁵ to stop all services on the target machine where the database instance is scheduled for maintenance. . For each service repeat the following actions:

1. Stop the service without using -force option or relocate the service. Service relocation is required for singleton service (i.e., runs only on one instance at a time)

```
$srvctl stop service -db <db_name> -service <service_name> -instance <instance_name
or (NOTE: Omitting -service stops all services)
$srvctl relocate service -db <db_name> -service <service_name> -oldinst <oldins> -
newinst <newinst>
```

 Disable the service and allow sessions some time to drain. E.g., 2-30 minutes. This avoids the logon storm on the other active instance where the workload gets transferred. Disabling service is optional if you choose to disable the instance.

\$srvctl disable service -db <db name> -service <service name> -instance <instance name>

- 3. Wait to allow sessions to drain Example: 10-30 minutes
- 4. Check for long-running sessions and terminate these (you may check again afterwards)

```
SQL> select count(*) from ( select 1 from v$session where service_name in
upper('<service_name>') union all
select 1 from v$transaction where status = 'ACTIVE' )
SQL> exec dbms_service.disconnect_session ('<service_name>',
DBMS SERVICE.POST TRANSACTION);
```

- 5. Repeat steps 1-4 for all services targeted for planned maintenance.
- 6. Stop the database instance immediately.

\$srvctl stop instance -db <db name> -instance <instance name> -stopoption immediate

7. Disable instance to prevent restarts during maintenance

srvctl disable instance -db <db_name> -instance <instance_name>

- 8. Apply patch or carry out the scheduled maintenance work
- 9. Enable and then start the instance again

⁵ See Metalink note 1593712.1 @ https://support.oracle.com/epmos/faces/DocumentDisplay?id=1593712.1 for more details

\$srvctl enable instance -db <db_name> -instance <instance_name>
\$srvctl start instance -db <db name> -instance <instance name>

10. Enable then start the service back and check if the service is up and running

```
$srvctl enable service -db <db_name> -service <service_name> -instance <instance_name>
$srvctl start service -db <db name> -service <service name> -instance <instance name>
```

Figure 4, shows connections distribution of XYZ service across two RAC instances before and after Planned Downtime. Notice that the connection workload goes from fifty-fifty across both instances to one hundred-zero. In other words, RAC_INST_1 can be taken down for maintenance without any impact on the business operation.



Hiding Unplanned Database Downtime from WebSphere applications

WebSphere Servlets can be configured to handle unplanned database outages using the following features and mechanisms:

- » Fast Connection Failover (FCF)
- » Transaction Guard (TG)
- » Application Continuity (AC)

Please refer to the white paper, Java Programming with Oracle Database 12c RAC and Active Data Guard ⁶ for understanding these concepts in detail.

Developer or Web Application Steps

Need to set FCF to true for handling unplanned outages. FCF enables UCP to detect dead instance and helps in

transferring the work load to the surviving active instance as soon as the unplanned down event occurs. Enable

⁶ http://www-content.oracle.com/technetwork/database/application-development/12c-ha-concepts-2408080.pdf

Transaction Guard and Application Continuity to achieve continuous service without any interruption of in-flight work. Please refer to the white paper **Java Programming with Oracle Database 12c RAC and Active Data Guard**⁷ for understanding how TG and AC will protect your application from unplanned downtimes.

DBA or RDBMS Steps

To simulate Fast Connection Failover, the DBA may either stop the service on one instance with –force option (as specified hereafter) or, alternatively, kill the Oracle instance SMON background process. An even more drastic approach consists in powering down of one of the nodes supporting the database.

```
$srvctl stop service -db <db_name> -service <service_name> -instance <instance_name> -
force
```

Figure 5, shows connections distribution of XYZ service across two RAC instances before and after unplanned downtime. Notice that the connection workload goes from fifty-fifty across both instances to hundred-zero. In other words, the remaining instances sustain the workload without disrupting the business operation.



Runtime Load Balancing (RLB) with WebSphere Servlets

Runtime Connection Load Balancing enables routing of work requests across RAC or ADG instances to achieve predictable runtime performance. RAC and GDS post runtime load balancing advisories every 30 seconds. UCP uses the Load Balancing advisory to balance the work across RAC instances, dynamically and thereby achieving best scalability. Runtime Load Balancing comes also into play when new node(s)/instance(s) are added/removed to/from the service; the work load gets balanced in both situations without any manual intervention.

⁷ http://www-content.oracle.com/technetwork/database/application-development/12c-ha-concepts-2408080.pdf

Developer or Web Application steps

Web applications need to set the UCP property 'setFastConnectionFailover' to true as already described (refer to "Fig 2.4:Enabling FCF" for more details) to allow receiving FAN Load Balancing advisories. UCP dispenses connections from the least loaded database instance (in RAC or GDS environments). Ultimately the workload is uniformly spread across the databases in question (RAC or GDS).

DBA or RDBMS steps

Configure the Oracle RAC Load Balancing Advisory with the following values.

Set 'Runtime Load Balancing Goal' to SERVICE_TIME or THROUGHPUT

```
$srvctl modify service -db <db_name> -service <service_name> -rlbgoal SERVICE_TIME
$gdsctl modify service -db <db name> -service <service name> -rlbgoal SERVICE TIME
```

Set 'Connection Load Balancing Goal' to SHORT

```
$srvctl modify service -db <db_name> -service <service_name> -clbgoal SHORT
$gdsctl modify service -db <db_name> -service <service_name> -clbgoal SHORT
```

Figure 6, shows connections distribution of XYZ service across three RAC instances. Notice that the workload is gradually distributed across the available instances with 50-50 connections each between RAC_Instance_1 and RAC_Instance_2. When a new instance, RAC_Instance_3 is added, the load will be re-distributed evenly to 34-34-32. After some time, RAC_Instance_3 is removed, UCP gradually rebalances the load between the remaining instances and in this case, achieves 50-50 connection workload distribution.



Appendix

Enable JDBC & UCP logging for debugging

Enable JDBC & UCP logging when there are issues. This helps to debug and find the root cause of the problem.

There are few steps for enabling JDBC & UCP logging.

- » Configure debug jar in the classpath
- » Enable logging
- » Setup a config file for advanced logging

Configure debug jar in the classpath:

Make sure to have ojdbc7_g.jar in the classpath under JDBC&UCP provider created as shown below.

<u>JDBC providers</u> > Oracle JDBC Driver UCP

Use this page to edit properties of a Java Database Connectivity (JDBC) provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment.

	Additional Properties
cope	Data sources
ells:slc03rznNode01Cell:nodes:slc03rznNode01:servers:server1	
Name	
)racle JDBC Driver UCP	
escription	
racle JDBC Driver UCP	
lass path	from the same
{ORACLE_JDBC_DRIVER_PATH}/ojdbc7_g.jar	
lass path	Use the jar files from the same

Enable logging

In order to get any log output from the Oracle JDBC drivers you must enable logging. Enable logging by setting the system property -Doracle.jdbc.Trace = TRUE. This turns logging ON. Refer to *Fig 5. Enable JDBC/UCP Logging in WebSphere.*

Setup a config file for advanced logging

Create a configuration file, for example oracletrace.properties and insert the following and save the file. Enable the config file by setting the system properties **–Djava.util.logging.config.file =<localtion of the config file>.** Refer to *Fig 5. Enable JDBC/UCP Logging in WebSphere.*

FOR UCP logs .level=WARNING oracle.ucp.jdbc.oracle.level=FINEST oracle.ucp.jdbc.level=FINEST oracle.ucp.common.level=FINEST oracle.ucp.jdbc.oracle.rlb.level=FINEST # For JDBC Driver logs level=SEVERE oracle.jdbc.level=ALL oracle.jdbc.driver.level=FINEST oracle.jdbc.pool.level=FINEST oracle.jdbc.util.level=OFF oracle.jdbc.handlers=java.util.logging.FileHandler java.util.logging.FileHandler.level=FINE java.util.logging.FileHandler.pattern=jdbc.log java.util.logging.FileHandler.count=1 java.util.logging.FileHandler.formatter=java.util.logging.SimpleFormatter

Fig.5: Enable JDBC/UCP Logging in WebSphere

Application servers						
Applic	Application servers > server1 > Process definition > Tava Virtual Machine > Custom properties					
Use th	Itee this name to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can set internal system.					
configu	configuration properties.					
± Pref	erences					
New	Delete					
	∎ ₩ ₽					
Select	Name 🗘	Value 🗘	Description 🗘			
You c	an administer the following resources:					
	com.ibm.security.jgss.debug	off				
	com.ibm.security.krb5.Krb5Debug	off				
	java.util.logging.config.file	/opt/IBM/WebSphere/AppServer_1/profiles /AppSrv01/logs/server1/Oracletrace /oracletrace.properties				
	java.util.logging.configureByLoggingPropertiesFile	true				
	oracle.jdbc.Trace	true				
	oracle.ucp.PlannedDrainingPeriod	30				
Total	6					

Conclusion

This paper furnishes a comprehensive and practical coverage of high-availability and load balancing in of WebSphere web applications with Oracle Database 12c; more specifically how to design Web applications and configure the RDBMS, UCP and the WebSphere container for resiliency to planned, and unplanned database downtimes and workload balancing. The steps described in this paper are valid for all Oracle Database 12c high availability and scalability configurations including RAC, RAC One and Active Data Guard. The complete UCP WebSphere demo referenced in this paper will be posted on https://github.com/oracle/jdbc-ucp. Java architects, Web application designers and DBAs may now design robust and reliable WebSphere Web applications for better user experience and application continuity.

ORACLE

Oracle Corporation, World Headquarters

500 Oracle Parkway Redwood Shores, CA 94065, USA Worldwide Inquiries Phone: +1.650.506.7000 Fax: +1.650.506.7200

CONNECT WITH US

B blogs.oracle.com f facebook.com/oracle twitter.com/oracle

oracle.com

Hardware and Software, Engineered to Work Together

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Oracle is committed to developing practices and products that help protect the environment