Oracle® Fusion Middleware

Administering Oracle WebLogic Server with Cloud Control

12*c* (12.2.1)

E55182-01

October 2015

This document describes how to configure, manage, and monitor Oracle WebLogic Server using Cloud Control.

Oracle Fusion Middleware Administering Oracle WebLogic Server with Cloud Control, 12*c* (12.2.1)

E55182-01

Contents

3 WebLogic Server Clusters

4 WebLogic Server Machines

5 WebLogic Server JDBC Data Sources

6 WebLogic Server Templates

Preface

This preface describes the document accessibility features and conventions used in this guide—*Administering Oracle WebLogic Server with Cloud Control*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 WebLogic Server Domains

An Oracle WebLogic Server administration domain is a logically related group of Oracle WebLogic Server resources.

Domains include a special Oracle WebLogic Server instance called the Administration Server, which is the central point from which you configure and manage all resources in the domain. Usually, you configure a domain to include additional Oracle WebLogic Server instances called Managed Servers. You deploy Web applications, EJBs, Web services, and other resources onto the Managed Servers and use the Administration Server for configuration and management purposes only.

1.1 Configure domains

This section describes how to configure your domain. This section includes the following tasks:

- Configure general settings
- Configure domain JTA settings
- Configure the default JPA persistence provider
- Configure domain EJBs
- Configure domain Web applications
- Create domain notes

1.1.1 Configure general settings

To configure general settings for a domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Administration**, then select **General Settings**.

   From the General Settings page you can configure administrative options that apply to all server instances in the current domain, such as:

   - Enable Administration Port
   - Administration Port
   - Production Mode
   - Enable Exalogic Optimizations

- Enable Cluster Constraints
- On-demand Employment of Internal Applications
- Enable Oracle Guardian Agent

For more information about these fields, see Configuration Options.

Optionally, expand **Advanced** to define advanced settings for this domain.

4. Click **Save**.

1.1.1.1 Configure the domain-wide administration port

Before you begin:

The administration port accepts only secure, SSL traffic, and all connections via the port require authentication by a server administrator. Because of these features, enabling the administration port imposes the following restrictions on your domain:

- The Administration Server and all Managed Servers in your domain must be configured with support for the SSL protocol.
- All server instances in the domain, including the Administration Server, enable or disable the administration port at the same time.

> **Note:** The administration port cannot be dynamically enabled on a Managed Server. You must shut down each Managed Server, enable the administration port, then restart.

WebLogic Server provides the option to enable an SSL administration port for use with all server instances in the domain. Using the administration port is strongly recommended. It provides three capabilities:

- Since communication uses SSL, administration traffic (which includes such things as administrator passwords) is more secure.
- It enables you to start a server instance in the `STANDBY` state.
- It enables you to separate administration traffic from application traffic in your domain.

To enable the administration port for a domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Administration**, then select **General Settings**.
4. Select the **Enable Administration Port** check box to enable the SSL administration for this domain.
5. In the **Administration Port** field, enter the SSL port number that server instances in the domain use as the administration port. You can override an individual server instance's administration port assignment on the **Advanced** options portion of the General Settings page.
6. Click **Save**.

After you finish:

Start all Managed Server instances in the domain. You do not need to restart the Administration Server.

1.1.1.2 Archive domain configuration files

To configure how many archive files are retained:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Administration**, then select **General Settings**.
4. Expand **Advanced**, then select the **Configuration Archive Enabled** check box.
5. In the **Archive Configuration Count** field, enter the number of archive files to retain.
6. Click **Save**.

1.1.1.3 Change to production mode

All server instances in a domain run either in development mode or production mode. In general, production mode requires you to configure additional security features.

To configure all server instances in a domain to run in production mode:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Administration**, then select **General Settings**.
4. Select the **Production Mode** check box.
5. Click **Save**.
6. Shut down any server instances that are currently running.
7. Invoke the domain's `startWebLogic` script. The Administration Server starts in the new mode.
8. If the domain contains Managed Servers, start the Managed Servers.

1.1.2 Configure domain JTA settings

To configure the Java Transaction API (JTA) of a domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Administration**, then select **Java Transaction API (JTA)**.

   From the JTA page you can define the JTA configuration settings for your domain, such as:

   - Timeout Seconds
   - Abandon Timeout Seconds

- Before Completion Iteration Limit
- Max Transactions
- Max Unique Name Statistics
- Checkpoint Interval Seconds
- Forget Heuristics
- Unregister Resource Grace Period
- Execute XA Calls in Parallel
- Enable Two Phase Commit
- Enable Tightly Coupled Transactions
- Enable Cluster-Wide Recovery

For more information about these fields, see Configuration Options.

Optionally, expand **Advanced** to define advanced settings for this domain.

4. Click **Save**.

1.1.3 Configure the default JPA persistence provider

You can specify which Java Persistence API (JPA) persistence provider to use for each persistence entity in the `persistence.xml` file. However, if no persistence provider is specified, the domain-wide default provider is used.

Changing the default provider does not affect applications that are already deployed. The setting takes effect when the server instance is restarted or the application is manually redeployed.

To specify the default JPA provider in a domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Administration**, then select **Java Persistence API (JPA)**.

   From the JPA page you can define the JPA configuration for your domain by selecting a JPA provider from the **Default JPA Provider** menu.

   For more information, see Configuration Options.
4. Click **Save**.

1.1.4 Configure domain EJBs

To configure the EJBs in a domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Administration**, then select **EJBs**.

   From the EJB page you can define domain-wide EJB configuration settings, such as:

- Java Compiler
- Prepend Java Compiler Options
- Append Java Compiler Options
- Extra RMIC Options
- Keep Generated EJBC Source Files
- Force Generation
- Temporary Directory
- Extra EJBC Options

For more information about these fields, see Configuration Options.

4. Click **Save**.

1.1.5 Configure domain Web applications

To configure Web applications in a domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Administration**, then select **Web Applications**.

   From the Web Applications page you can define the domain-wide Web application configuration settings, such as:

   - Relogin Enabled
   - Allow All Roles
   - Filter Dispatched Requests
   - Overload Protection Enabled
   - X-Powered-By Header
   - Mime Mapping File
   - Optimistic Serialization
   - Error on Name Request Time Value
   - Client Cert Proxy Enabled
   - HTTP Trace Support Enabled
   - WebLogic Plug-in Enabled
   - Auth Cookie Enabled
   - Change Session ID on Authentication
   - WAP Enabled
   - Post Timeout
   - Maximum Post Timeout
   - Maximum Post Size
   - Work Context Propagation Enabled

- P3P Header Value
- JSP Compiler Backwards Compatible
- Archived Real Path Enabled

For more information about these fields, see Configuration Options.

4. Click **Save**.

1.1.6 Create domain notes

To create notes for domain configuration:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** menu, select **Administration**, then select **Notes**.
4. On the Notes page, enter your notes.
5. Click **Save**.

For more information, see Configuration Options.

2 WebLogic Servers

A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.

When you create a domain, you also create an Administration Server, which distributes configuration changes to other servers in the domain. In a typical production environment, you create one or more Managed Servers in the domain to host business applications and use the Administration Server only to configure and monitor the Managed Servers.

2.1 Configure servers

This section describes how to configure servers. This section includes the following tasks:

- Define general server configuration
- Configure cluster settings
- Configure server services
- Configure server keystores
- Change the keystore configuration
- Configure server SSL settings
- Change the identity and trust location
- Configure server migration
- Configure server SAML Federation Services
- Configure server deployment staging
- Configure server tuning
- Configure server overload
- Configure server health monitoring
- Configure server startup
- Configure server startup and shutdown settings
- Configure server Web Services
- Configure server Coherence cluster settings
- Configure server protocol settings
- Create server notes

- Configure the server template for a dynamic server

2.1.1 Define general server configuration

To configure general settings for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **General Settings**.
5. From the General Settings page, you can define the configuration settings for this server instance, such as:
   - Template
   - Machine
   - Cluster
   - Listen Address
   - Listen Port Enabled
   - Listen Port
   - SSL Listen Port Enabled
   - SSL Listen Port
   - Client Cert Proxy Enabled
   - Java Compiler
   - Diagnostic Volume

   For more information about these fields, see Configuration Options.
6. Optionally, expand **Advanced** to define advanced configuration settings for this server instance.
7. Click **Save**.

2.1.1.1 Change server template

To change the server template associated with a server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **General Settings**.
5. Click **Change** and select the server template from the Server Templates dropdown menu.
6. Click **Yes**.

2.1.1.2 Specify a startup mode

The startup mode specifies the state in which a server instance should be started. The default is to start in the `RUNNING` state.

To specify the startup mode for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **General Settings**.
5. On the General page, expand **Advanced** at the bottom of the page to display the advanced configuration options.
6. In the **Startup Mode** field, select:

   - **Running**: In the `RUNNING` state, a server offers its services to clients and can operate as a full member of a cluster.
   - **Administration**: In the `ADMIN` state, the server is up and running, but available only for administration operations, allowing you to perform server and application-level administration tasks without risk to running applications.
   - **Standby**: In the `STANDBY` mode, the server listens for administrative requests only on the domain-wide administration port and only accepts life cycle commands that transition the server instance to either the `RUNNING` or `SHUTDOWN` state. Other administration requests are not accepted. If you specify `STANDBY`, you must also enable the domain-wide administration port.

   For more information, see Configuration Options.
7. Click **Save**.

2.1.1.3 Configure listen addresses

To configure the listen address of a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **General Settings**.
5. On the General Settings page, enter a value in **Listen Address**.

   For more information, see Configuration Options.
6. Click **Save**.

2.1.1.4 Configure listen ports

To configure the listen ports of a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.

3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **General Settings**.
5. If you want to disable the non-SSL listen port so that the server listens only on the SSL listen port, deselect **Listen Port Enabled**. If you want to disable the SSL listen port so that the server listens only on the non-SSL listen port, deselect **SSL Listen Port Enabled**.

   > **Note:** You cannot disable both the non-SSL listen port and the SSL listen port. At least one port must be active.

6. If you are using the non-SSL listen port and you want to modify the default port number, change the value in **Listen Port**.
7. If you want to modify the default SSL listen port number, change the value in **SSL Listen Port**.

   For more information, see Configuration Options.
8. Click **Save**.

2.1.1.5 Change server compiler options

To change the standard Java compiler values for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **General Settings**.
5. On the General Settings page, update the **Java Compiler** field with the full path of the compiler to use for all applications hosted on this server that need to compile Java code.
6. Expand **Advanced** to display advanced configuration options.
7. Update the following compile options as necessary:
   - **Prepend to classpath**: Options to prepend to the Java compiler classpath when compiling Java code.
   - **Append to classpath**: Options to append to the Java compiler classpath when compiling Java code.
   - **Extra RMI Compiler Options**: Options passed to the RMIC compiler during server-side generation.
   - **Extra EJB Compiler Options**: Options passed to the EJB compiler during server-side generation.

   For more information about these fields, see Configuration Options.
8. Click **Save**.

2.1.2 Configure cluster settings

A WebLogic Server cluster is a group of servers that work together to provide a scalable and reliable application platform.

To configure cluster settings for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Clusters**.
5. From the Cluster page, you can define the cluster configuration settings for this server instance, such as:
   - Replication Group
   - Preferred Secondary Group
   - Cluster Weight
   - Interface Address
   - Replication Ports

   For more information about these fields, see Configuration Options.
6. Click **Save**.

2.1.3 Configure server services

To configure WebLogic service settings for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Services**.
5. From the Services page, you can define the service configuration settings for this server instance, such as:
   - JMS Configuration
   - Default Store
   - Transaction Log Store
   - Messaging Bridge Configuration
   - XML Services Configuration

   For more information about these fields, see Configuration Options.
6. Click **Save**.

2.1.4 Configure server keystores

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). You can view and define various keystore configurations which help you manage the security of message transmissions.

To configure keystore settings for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Keystores**.
5. To change your keystore configuration:
    1. Click **Change**.
    2. From the Keystores dropdown menu, select the keystore configuration you want to use.
    3. Click **Save**.
6. From the Keystore page, you can define other keystore configuration settings for this server instance, such as:
    - Keystores
    - Demo Identity Keystore
    - Demo Identity Keystore Type
    - Demo Identity Keystore Passphrase
    - Demo Trust Keystore
    - Demo Trust Keystore Type
    - Demo Trust Keystore Passphrase
    - Java Standard Trust Keystore
    - Java Standard Trust Keystore Type
    - Java Standard Trust Keystore Passphrase

    For more information about these fields, see Configuration Options.
7. Click **Save**.

2.1.5 Change the keystore configuration

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). You can view and define various keystore configurations which help you manage the security of message transmissions.

To change the keystore:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.

4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Keystores**.
5. Click **Change**.
6. From the Keystores dropdown menu, select the keystore configuration you want to use.

   For more information about these fields, see Configuration Options.
7. Click **Save**.

2.1.6 Configure server SSL settings

You can view and define various Secure Sockets Layer (SSL) settings for a server instance, which help you manage the security of message transmissions.

To configure SSL settings for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **SSL**.
5. To change your identity and trust location:
   1. Click **Change**.
   2. From the Identity and Trust Locations dropdown menu, select the location you want to use.
   3. Click **Save**.
6. From the SSL page, you can define the SSL configuration settings for this server instance, such as:
   - Identity and Trust Locations
   - Private Key Location
   - Private Key Alias
   - Private Key Passphrase
   - Certificate Location
   - Trusted Certificate Authorities

   For more information about these fields, see Configuration Options.
7. Optionally, expand **Advanced** to define advanced configuration settings for this server instance.
8. Click **Save**.

2.1.7 Change the identity and trust location

You can view and define various Secure Sockets Layer (SSL) settings for a server instance, which help you manage the security of message transmissions.

To change the identity and trust location:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.

2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **SSL**.
5. Click **Change**.
6. From the Identity and Trust Locations dropdown menu, select the location you want to use.

   For more information about these fields, see Configuration Options.
7. Click **Save**.

2.1.8 Configure server SAML Federation Services

You can configure a WebLogic Server instance to function as a producer or as a consumer of SAML assertions that can be used for the following:

- Web single sign-on between online business partners
- Exchange of identity information in Web services security

The general process of configuring Federation Services depends upon the version of SAML you are using. WebLogic Server supports both SAML 1.1 and SAML 2.0.

To configure WebLogic Server to serve as a SAML 1.1 federated partner:

- Configure SAML 1.1 source services
- Configure SAML 1.1 destination services

To configure WebLogic Server to serve as a SAML 2.0 federated partner:

- Configure SAML 2.0 general services
- Configure SAML 2.0 Identity Provider services
- Configure SAML 2.0 Service Provider services

2.1.8.1 Configure SAML 1.1 source services

Before you begin

You must first configure a SAML Credential Mapper V2 security provider in the server's security realm.

You can configure a WebLogic Server instance to function as a SAML source site. A SAML source site is a site that provides an Intersite Transfer Service (ITS). A source site generates assertions that are conveyed to a destination site using one of the single sign-on profiles.

To configure a server as a SAML source site:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Federation Services**.
5. Select the **SAML 1.1 Source Site** page.

6. Select the **Source Site Enabled** attribute to cause this server to act as a source for SAML assertions.
7. From the SAML 1.1 Source Site page, you can also define other desired configuration settings for this server instance, such as:
   - Source Site URL
   - Signing Key Alias
   - Intersite Transfer URIs
   - ITS Requires SSL
   - Assertion Retrieval URIs
   - ARS Requires SSL
   - ARS Requires Two-Way SSL Authentication
   - Assertion Store Class Name
   - Assertion Store Properties

   For more information about these fields, see Configuration Options.
8. Click **Save**.

2.1.8.2 Configure SAML 1.1 destination services

Before you begin

You must first configure a SAML Identity Asserter V2 security provider in the server's security realm.

You can configure a WebLogic Server instance to function as a SAML destination site. A destination site can receive SAML assertions and use them to authenticate local subjects.

To configure a server as a SAML destination site:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Federation Services**.
5. Select the **SAML 1.1 Destination Site** page.
6. Select the **Destination Site Enabled** attribute to enable the Assertion Consumer Service.
7. From the SAML 1.1 Destination Site page, you can also define other desired configuration settings for this server instance, such as:
   - Assertion Consumer URIs
   - ACS Requires SSL
   - SSL Client Identity Alias
   - POST Recipient Check Enabled
   - POST One-Use Check Enabled
   - Used Assertion Cache Class Name

- Used Assertion Cache Properties

  For more information about these fields, see Configuration Options.

8. Click **Save**.

2.1.8.3 Configure SAML 2.0 general services

You can configure general SAML 2.0 services for a server. If you are configuring SAML 2.0 Web single sign-on services with your federated partners, the site information you configure is published in a metadata file that you send to your federated partners.

To configure the general SAML 2.0 properties of this server:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Federation Services**.
5. Select the **SAML 2.0 General** page.
6. Select the **Replicated Cache Enabled** attribute to use the persistent cache for storing SAML 2.0 artifacts.

   This option is required if you are configuring SAML 2.0 services in two or more WebLogic Server instances in your domain. For example, if you are configuring SAML 2.0 services in a cluster, you must enable this option in each Managed Server instance individually.

   > **Note:** If you are configuring SAML 2.0 services in two or more WebLogic Server instances in your domain, you must configure the RDBMS security store. The embedded LDAP server is not supported in these configurations.

7. In the **Site Info** section, enter the following information about your SAML 2.0 site:
   - Contact person details
   - Your organization's name and URL
   - The Published Site URL, which is the top-level URL for your site's SAML 2.0 service endpoints. This URL must be appended with the string `/saml2`, which will be automatically combined with constant suffixes to create full endpoint URLs.
8. In the **Bindings** section, enter the common binding information to be used by this SAML 2.0 server instance.

   If you do not specify a Transport Layer Security key alias and passphrase, the server's configured SSL private key alias and passphrase from the server's SSL configuration is used for the TLS alias by default.

9. If the Artifact binding is enabled for any SAML 2.0 security provider hosted on this server instance, define the Artifact Resolution Service settings in the **Artifact Resolution Service** section.
10. In the **Single Sign-on** section, enter the keystore alias and passphrase for the key to be used for signing documents sent to federated partners.

If you do not specify a single sign-on signing key alias and passphrase, the server's configured SSL private key alias and passphrase from the server's SSL configuration is used by default.

For more information, see Configuration Options.

11. Click **Save**.

After you finish

After you have configured this server's general SAML 2.0 services, select the **SAML 2.0 Identity Provider** page or the **SAML 2.0 Service Provider** page to configure this server as an Identity Provider or Service Provider, respectively.

For more information, see Configure SAML 2.0 Identity Provider services and Configure SAML 2.0 Service Provider services.

2.1.8.4 Configure SAML 2.0 Identity Provider services

You can configure a server instance in the role of SAML 2.0 Identity Provider. A SAML 2.0 Identity Provider creates, maintains, and manages identity information for principals, and provides principal authentication to other Service Provider partners within a federation by generating SAML 2.0 assertions for those partners.

To configure a server as a SAML 2.0 Identity Provider:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Federation Services**.
5. Select the **SAML 2.0 Identity Provider** page.
6. Select the **Enabled** attribute to activate this server's SAML 2.0 services in the role of Identity Provider.
7. Select **Only Accept Signed Authentication Requests** if you want to ensure that any incoming authentication requests must be signed.
8. If you are using a custom login Web application to which unauthenticated requests are directed:
   - Select **Login Customized**.
   - Enter the URL of the custom login Web application.
   - Enter the login return query parameter

     The query parameter is a unique string that the SAML 2.0 services uses to hold the login return URL for the local single sign-on service servlet. (Note that, as an alternative, the login return URL can also be specified in the login Web application.)
9. Set the SAML bindings for which this server instance is enabled, and select the preferred binding type.

   For more information, see Configuration Options.
10. Click **Save**.

After you finish

Coordinate with your federated partners to ensure that the SAML bindings you have enabled for this SAML authority, as well as your requirements for signed documents, are compatible with your partners.

2.1.8.5 Configure SAML 2.0 Service Provider services

You can configure a WebLogic Server instances as a SAML 2.0 Service Provider. A Service Provider is a SAML authority that can receive SAML assertions and extract identity information from those assertions. The identity information can then be mapped to local Subjects, and optionally groups as well, that can be authenticated.

To configure a server as a SAML 2.0 Service Provider:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Federation Services**.
5. Select the **SAML 2.0 Service Provider** page.
6. Select the **Enabled** attribute to activate SAML 2.0 services in this server instance in the role of Identity Provider.
7. Set the configuration options for the local SAML 2.0 Service Provider services as appropriate. Note the following:
   - Choose options for **Always Sign Authentication Requests** and **Only Accept Signed Assertions** as desired and in a manner that is coordinated with your federated partners so that authentication requests and assertions are accepted.
   - Communicate the SAML bindings settings for this server instance with your federated partners to ensure compatibility.

   For more information, see Configuration Options.
8. Click **Save**.

After you finish

Coordinate with your federated partners to ensure that the SAML bindings you have enabled for this SAML authority, as well as your requirements for signed documents, are compatible with your partners.

2.1.9 Configure server deployment staging

To configure the default deployment staging on a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Deployment**.
5. From the Deployment page, you can define the default deployment staging configuration for this server instance, such as:
   - Staging Mode

- Staging Directory Mode
- Upload Directory Name

For more information about these fields, see Configuration Options.

6. Click **Save**.

2.1.10 Configure server migration

To configure migration settings for clustered servers:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Migration**.
5. From the Migration page, you can define the configuration settings for this server instance, such as:
   - Automatic Server Migration Enabled
   - Candidate Machines
   - JMS Service Candidate Servers
   - Automatic JTA Migration Enabled
   - JTA Candidate Servers
   - Pre-Migration Script Path
   - Post-Migration Script Path
   - Post-Migration Script Fail Cancels Automatic Migration
   - Allow Post-Migration Script to Run on a Different Machine
   - Enable Strict Ownership Check

   For more information about these fields, see Configuration Options.
6. Click **Save**.

2.1.11 Configure server tuning

You can tune the performance and functionality of a server instance by configuring the tuning settings.

To configure tuning settings for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Tuning**.
5. From the Tuning page, you can define the tuning configuration settings for this server instance, such as:

- Enable Native IO
- JavaSocketMuxer Socket Readers
- Enable Gathered Writes
- Enable Scattered Reads
- Maximum Open Sockets
- Stuck Thread Max Time
- Stuck Thread Timer Interval
- Accept Backlog
- Login Timeout
- SSL Login Timeout
- Reverse DNS Allowed

For more information about these fields, see Configuration Options.

6. Optionally, expand **Advanced** to define advanced configuration settings for this server instance.
7. Click **Save**.

2.1.11.1 Enable native IO

To enable native IO for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Tuning**.
5. On the Tuning page, if the Enable Native IO check box is not selected, select the check box.

   For more information, see Configuration Options.
6. Click **Save**.

2.1.11.2 Enable NIOSocketMuxer

To enable non-blocking IO for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Tuning**.
5. Expand **Advanced** to access advanced tuning parameters.
6. Enter `weblogic.socket.NIOSocketMuxer` in the **Muxer Class** field.

   For more information, see Configuration Options.

7. Click **Save**.

2.1.11.3 Tune connection backlog buffering

You can tune the number of connection requests that a WebLogic Server instance will accept before refusing additional requests.

To tune connection backlog buffering for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Tuning**.
5. Modify the **Accept Backlog** value as necessary to tune the number of TCP connections the server instance can buffer in the wait queue.

   - If many connections are dropped or refused at the client, and no other error messages are on the server, the **Accept Backlog** value might be set too low.
   - If you receive "connection refused" messages when you try to access WebLogic Server, raise the **Accept Backlog** value from the default by 25 percent. Continue increasing the value by 25 percent until the messages cease to appear.

   For more information about these fields, see Configuration Options.
6. Click **Save**.

2.1.11.4 Tune stuck thread detection behavior

WebLogic Server diagnoses a thread as stuck if it is continually working (not idle) for a set period of time. You can tune a server's thread detection behavior by changing the length of time before a thread is diagnosed as stuck, and by changing the frequency with which the server checks for stuck threads.

To configure stuck thread detection behavior for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Tuning**.
5. From the Tuning page, update the following options as necessary:

   - **Stuck Thread Max Time**: Amount of time, in seconds, that a thread must be continually working before a server instance diagnoses a thread as being stuck.
   - **Stuck Thread Timer Interval**: Amount of time, in seconds, after which a server instance periodically scans threads to see if they have been continually working for the configured **Stuck Thread Max Time**.

   For more information about these fields, see Configuration Options.
6. Click **Save**.

2.1.11.5 Replicate domain configuration files for Managed Server independence

The server instance for which you configure Managed Server Independence (MSI) replication does not need to be running.

To configure a Managed Server to replicate a domain's configuration files:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Tuning**.
5. On the Tuning page, click **Advanced** at the bottom of the page to display advanced configuration settings.
6. Ensure that the **Managed Server Independence Enabled** check box is selected.

   For more information, see Configuration Options.
7. Click **Save**.

After you finish

If the Managed Server is running, restart it.

2.1.12 Configure server overload

You can configure how WebLogic Server should react in the case of an overload or failure condition.

To configure overload settings for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Overload**.
5. From the Overload page, you can define the overload configuration settings for this server, such as:

   - Shared Capacity for Work Managers
   - Failure Action
   - Panic Action
   - Free Memory Percent High Threshold
   - Free Memory Percent Low Threshold
   - Max Stuck Thread Time
   - Stuck Thread Count

   For more information about these fields, see Configuration Options.
6. Click **Save**.

2.1.13 Configure server health monitoring

WebLogic Server provides a self-health monitoring capability to improve the reliability and availability of servers in a WebLogic Server domain. Selected subsystems within each server monitor their health status based on criteria specific to the subsystem.

You can configure the frequency of a server's automated health checks and the frequency with which the Node Manager application (optional) checks the server's health state. You can also use this page to specify whether Node Manager automatically stops and restarts the server if the server reaches the "failed" health state.

To configure health monitoring settings for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Health Monitoring**.
5. From the Health Monitoring page, you can define the health monitoring configuration settings for this server instance, such as:
   - Health Check Interval
   - Auto Kill If Failed
   - Auto Restart
   - Restart Interval
   - Max Restarts Within Interval
   - Restart Delay Seconds

   For more information about these fields, see Configuration Options.
6. Click **Save**.

2.1.14 Configure server startup

Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. You can configure the startup settings that Node Manager will use to start this server on a remote machine.

To configure the startup options that Node Manager uses to start a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Server Start**.
5. From the Server Start page, you can define the startup configuration settings for this server instance, such as:
   - Java Home

- Java Vendor
- BEA Home
- Root Directory
- Class Path
- Arguments
- Security Policy File
- User Name
- Password
- Confirm Password

For more information about these fields, see Configuration Options.

6. Click **Save**.

2.1.15 Configure server startup and shutdown settings

To configure the startup and shutdown settings for a server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Start/Stop**.
5. From the Start/Stop page, you can define the startup and shutdown configuration settings for this server instance, such as:
   - Ignore Sessions During Shutdown
   - Graceful Shutdown Timeout
   - Startup Timeout
   - Server LifeCycle Timeout

   For more information about these fields, see Configuration Options.
6. Click **Save**.

2.1.16 Configure server Web Services

You must install a Web service before you can view and modify its configuration. This section includes the following tasks:

- Configure message buffering for Web services
- Configure Web service reliable messaging
- View logical stores

2.1.16.1 Configure message buffering for Web services

To configure message buffering for Web services:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.

2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Web Services**.
5. Select the **Buffering** page.
6. From the Buffering page, you can define the message buffering configuration properties for this server instance, such as:
   - Retry Count
   - Retry Delay
   - Request Queue Enabled
   - Request Queue Connection Factory JNDI Name
   - Request Queue Transaction Enabled
   - Response Queue Enabled
   - Response Queue Connection Factory JNDI Name
   - Response Queue Transaction Enabled

   For more information about these fields, see Configuration Options.
7. Click **Save**.

2.1.16.2 Configure Web service reliable messaging

Web service reliable messaging is a framework that enables an application running on one application server to reliably invoke a Web service running on another application server, assuming that both servers implement the WS-Reliable Messaging specification. Reliable is defined as the ability to guarantee message delivery between the two Web services. Use this page to customize reliable messaging configuration on the Web service endpoint.

To configure reliable messaging for Web services:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Web Services**.
5. Select the **Reliable Message** page.
6. From the Reliable Messaging page, you can define the reliable messaging configuration properties for this server instance, such as:
   - Base Retransmission Interval
   - Enable Retransmission Exponential Backoff
   - Non-buffered Source
   - Non-buffered Destination
   - Acknowledgement Interval

- Inactivity Timeout
- Sequence Expiration

For more information about these fields, see Configuration Options.

7. Click **Save**.

2.1.16.3 View logical stores

A logical store is a named unit of storage that provides the business configuration requirements and connects the Web service to the physical store and buffering queue.

To view the logical stores configured for this server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Web Services**.
5. Select the **Logical Stores** page.
6. From the Logical Stores page, you can view configuration information for each logical store configured for this server instance, such as:
   - Name
   - Persistence Strategy
   - Request Buffering Queue JNDI Name
   - Response Buffering Queue JNDI Name
   - Default

   For more information about these fields, see Configuration Options.

2.1.17 Configure server Coherence cluster settings

To configure Coherence for a specific server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Coherence**.
5. From the Coherence page, you can specify the Coherence cluster you want to use for this server instance and also define configuration settings such as:
   - Unicast Listen Address
   - Unicast Listen Port
   - Unicast Port Auto Adjust
   - Local Storage Enabled
   - Coherence Web Logical Storage Enabled

    - Site Name
    - Rack Name

6. Click **Save**.

   For more information, see Configuration Options.

2.1.18 Configure server protocol settings

This section describes how to configure server protocols.

This section includes the following tasks:

- Configure server protocol general settings
- Configure server HTTP settings
- Configure server jCOM settings
- Configure server IIOP settings
- Monitor server network channel settings
- Configure server network channel settings

2.1.18.1 Configure server protocol general settings

To configure general protocol settings for a server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Protocols**.
5. Select the **General Settings** page.
6. From the General Settings page, you can define connections settings for various communication protocols that this server instance uses, such as:
    - Complete Message Timeout
    - Idle Connection Timeout
    - Enable Tunneling
    - Tunneling Client Ping
    - Tunneling Client Timeout
    - Maximum Message Size

   For more information about these fields, see Configuration Options.
7. Click **Save**.

2.1.18.2 Configure server HTTP settings

To configure HTTP protocol settings for a server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.

3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Protocols**.
5. Select the **HTTP** page.
6. From the HTTP page, you can define the HTTP settings for this server instance, such as:
   - Default WebApp Context Root
   - Post Timeout
   - Max Post Size
   - Enable Keepalives
   - Duration
   - HTTPS Duration
   - Frontend Host
   - Frontend HTTP Port
   - Frontend HTTPS Port
   - WAP Enabled
   - Remote Address Override
   - Send Server Header
   - Accept Context Path in Get Real Path
   - HTTP Max Message Size
   - Enable Tunneling
   - Tunneling Client Ping
   - Tunneling Client Timeout

   For more information about these fields, see Configuration Options.
7. Click **Save**.

2.1.18.3 Configure server jCOM settings

To configure Java to COM (jCOM) protocol settings for a server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Protocols**.
5. Select the **jCOM** page.
6. From the jCOM page, you can define jCOM protocol settings for this server instance, such as:
   - Enable COM
   - NT Authentication Host

- Enable Native Mode
- Verbose Logging Enabled
- Enable Memory Logging
- Prefetch Enumeration
- Apartment Threaded

For more information about these fields, see Configuration Options.

7. Click **Save**.

2.1.18.4 Configure server IIOP settings

To configure Internet Inter-ORB Protocol (IIOP) settings for a server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Protocols**.
5. Select the **IIOP** page.
6. From the IIOP page, you can enable IIOP for this server instance.

   For more information about these fields, see Configuration Options.

   Optionally, expand **Advanced** to define advanced configuration settings for this server instance.
7. Click **Save**.

2.1.18.5 Monitor server network channel settings

To monitor network channel protocol settings for a server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Protocols**.
5. Select the **Channels** page.
6. The Channels table displays information about each network channel that has been configured for this server instance, such as:
   - Name
   - Protocol
   - Enabled
   - Listen Address
   - Listen Port
   - Public Address

- Public Port

For more information about these fields, see Configuration Options.

7. Click **Save**.

2.1.18.6 Configure server network channel settings

To configure network channel protocol settings for a server instance:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** dropdown menu, select **Administration**, then select **Protocols**.
5. Select the **Channels** page.
6. In the Channels table, select the name of the channel you want to configure.
7. Select **Configuration**.
8. From the **General** page, you can define general configuration settings for the network channel, such as:
    - Name
    - Protocol
    - Listen Port
    - Listen Address
    - External Listen Address
    - External Listen Port
    - Enabled

    For more information about these fields, see Configuration Options.

    Optionally, expand **Advanced** to define advanced configuration settings for this network channel.
9. Click **Save**.
10. From the **Security** page, you can define security configuration options for the network channel, such as:
    - Two Way SSL Enabled
    - Client Certificate Enforced

    For more information about these fields, see Configuration Options.

    Optionally, expand **Advanced** to define advanced configuration settings for this network channel.
11. Click **Save**.

> **Note:** For information about configuring server template network channel settings, see Configure server template network channel settings.

2.1.19 Create server notes

To create notes for server configuration:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** menu, select **Administration**, then select **Notes**.
5. On the Notes page, enter your notes.
6. Click **Save**.

For more information, see Configuration Options.

2.1.20 Configure the server template for a dynamic server

To view or change the server template configuration for a dynamic server:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **Target Navigation** pane, select the server instance you want to configure.
4. From the **WebLogic Server** menu, select **Administration**, then select **Server Template**.
5. To view or change the configuration for the server template associated with this dynamic server, click the **Server Template** link.

For more information, see Configuration Options.

3

WebLogic Server Clusters

A WebLogic Server cluster consists of multiple WebLogic Server server instances running simultaneously and working together to provide increased scalability and reliability.

A cluster appears to clients to be a single WebLogic Server instance. The server instances that constitute a cluster can run on the same machine, or be located on different machines. You can increase a cluster's capacity by adding additional server instances to the cluster on an existing machine, or you can add machines to the cluster to host the incremental server instances. Each server instance in a cluster must run the same version of WebLogic Server.

3.1 Configure clusters

This section describes how to configure clusters. This section includes the following tasks:

- Configure general cluster settings
- Configure cluster JTA settings
- Configure cluster messaging
- Configure server instances in cluster
- Configure cluster replication
- Configure cluster migration
- Configure cluster singleton services
- Configure cluster job scheduling
- Configure cluster overload settings
- Configure cluster health monitoring
- Configure cluster HTTP settings
- Configure cluster Coherence cluster settings
- Create cluster notes

3.1.1 Configure general cluster settings

To configure general settings for a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.

2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **General Settings**.

   From the General Settings page you can define the general settings for a cluster, such as:

   - Name
   - Default Load Algorithm
   - Cluster Address
   - Number of Servers in Cluster Address
   - Enable Transaction Affinity

   For more information about these fields, see Configuration Options.

   Optionally, expand **Advanced** to define advanced settings for this cluster.
5. Click **Save**.

3.1.2 Configure cluster JTA settings

To configure JTA for a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Java Transaction API (JTA)**.

   From the Java Transaction API (JTA) page you can define the JTA settings for a cluster, such as:

   - Timeout Seconds
   - Abandon Timeout Seconds
   - Before Completion Iteration Limit
   - Max Transactions
   - Max Unique Name Statistics
   - Checkpoint Interval Seconds
   - Forget Heuristics
   - Unregister Resource Grace Period
   - Execute XA Calls in Parallel
   - Enable Two Phase Commit
   - Enable Tightly Coupled Transactions
   - Enable Cluster-Wide Recovery

   For more information about these fields, see Configuration Options.

Optionally, expand **Advanced** to define advanced settings for this cluster.

5. Click **Save**.

3.1.3 Configure cluster messaging

Clusters use messaging for sharing session, load balancing and failover, JMS, and other information between cluster members. Clusters can use either unicast or multicast messaging. Multicast is a simple broadcast technology that enables multiple applications to subscribe to a given IP address and port number and listen for messages, but requires hardware configuration and support. Unicast does not have these requirements.

To configure messaging for a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Messaging**.

   From the Messaging page, you can define the messaging settings for your cluster, such as:

   - Messaging Mode
   - Unicast Broadcast Channel
   - Multicast Address
   - Multicast Port

   For more information about these fields, see Configuration Options.

   Optionally, expand **Advanced** to define advanced settings for this cluster.
5. Click **Save**.

3.1.4 Configure server instances in cluster

To configure server instances in a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Servers**.

   From the Servers page, you can define settings for server instances in the cluster, such as:

   - Server Template
   - Maximum Number of Servers
   - Server Name Prefix
   - Enable Calculated Listen Ports
   - Enable Calculated Machine Associations
   - Machine Name Match Expression

For more information about these fields, see Configuration Options.

5. Click **Save**.

3.1.5 Configure cluster replication

To configure replication for a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Replication**.

   From the Replication page, you can configure how WebLogic Server will replicate HTTP session state across a cluster, including settings such as:

   - Cross-cluster Replication Type
   - Remote Cluster Address
   - Replication Channel
   - Data Source for Session Persistence
   - Persist Sessions on Shutdown
   - Secure Replication Enabled

   For more information about these fields, see Configuration Options.

   Optionally, expand **Advanced** to define advanced settings for this cluster.
5. Click **Save**.

3.1.6 Configure cluster migration

If a clustered server fails, Node Manager can automatically restart the server instance and its services on another machine. You can specify the machines where Node Manager can restart migratable servers and also the data source used during server migration.

To configure server migration in a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Migration**.

   From the Migration page, you can specify settings related to cluster migration, such:

   - Candidate Machines for Migratable Servers
   - Migration Basis
   - Data Source for Automatic Migration
   - Auto Migration Table Name
   - Member Death Detector Enabled

- Member Discovery Timeout
- Leader Heartbeat Period
- Additional Migration Attempts
- Pause Time Between Migration Attempts

For more information about these fields, see Configuration Options.

5. Click **Save**.

3.1.7 Configure cluster singleton services

This section describes how to configure and control singleton services.

This section includes the following sections:

- Configure cluster singleton services general settings
- Configure cluster singleton services migration
- Create singleton service notes

3.1.7.1 Configure cluster singleton services general settings

To configure general settings for a cluster singleton service:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Singleton Services**.

   In the Singleton Services table, select the singleton service you want to configure.
5. Select **Configuration**, then select **General**.

   From the General page, you can define settings for the class associations of this singleton service, such as:

   - Name
   - Class Name
   - Additional Migration Attempts
   - Sleep Time Between Attempts

   For more information about these fields, see Configuration Options.
6. Click **Save**.

3.1.7.2 Configure cluster singleton services migration

To configure migration for a cluster singleton service:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.

4. From the **WebLogic Cluster** menu, select **Administration**, then select **Singleton Services**.

   In the Singleton Services table, select the singleton service you want to configure.

5. Select **Configuration**, then select **Migration**.

   From the Migration page, you can define migration settings for this singleton service, such as:

   - Name
   - User Preferred Server
   - Constrained Candidate Servers

   For more information about these fields, see Configuration Options.

6. Click **Save**.

3.1.7.3 Create singleton service notes

To create notes for a singleton service:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Singleton Services**.

   In the Singleton Services table, select the singleton service you want to configure.

5. Select **Notes**.
6. On the Notes page, enter your notes.

   For more information about these fields, see Configuration Options.

7. Click **Save**.

3.1.8 Configure cluster job scheduling

Job scheduling makes Java CommonJ timers cluster-aware and provides the ability to execute jobs periodically anywhere in a cluster without dependency on a particular server instance.

To configure job scheduling in a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Scheduling**.

   From the Scheduling page, you can define configuration settings that specify how information is shared across servers in a cluster.

5. In the **Data Source for Job Scheduler** field, select the data source to use.
6. In the **Job Scheduler Table Name** field, enter a table name to use for storing timers active with the job scheduler.

7. Click **Save**.

   For more information, see Configuration Options.

3.1.9 Configure cluster overload settings

To configure overload for a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Overload**.

   From the Overload page, you can configure the cluster-wide defaults that control how WebLogic Server instances in this cluster should react in the case of an overload or failure condition. The settings you can define include:

   - Shared Capacity for Work Managers
   - Failure Action
   - Panic Action
   - Free Memory Percent High Threshold
   - Free Memory Percent Low Threshold
   - Max Stuck Thread Time
   - Stuck Thread Count

   For more information about these fields, see Configuration Options.
5. Click **Save**.

3.1.10 Configure cluster health monitoring

WebLogic Server provides a self-health monitoring capability to improve the reliability and availability of servers in a WebLogic Server domain. Selected subsystems within each server instance monitor their health status based on criteria specific to the subsystem

To configure health monitoring characteristics for a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Health Monitoring**.

   From the Health Monitoring page, you can configure health monitoring characteristics for this cluster, such as:

   - Inter-Cluster Comm Link Health Check Interval
   - Health Check Interval
   - Health Check Periods Until Fencing
   - Fencing Grace Period

For more information about these fields, see Configuration Options.

5. Click **Save**.

3.1.11 Configure cluster HTTP settings

To configure HTTP settings for a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **HTTP**.

   From the HTTP page, you can define the HTTP settings for this cluster, such as:

   - Frontend Host
   - Frontend HTTP Port
   - Frontend HTTPS Port

   For more information about these fields, see Configuration Options.
5. Click **Save**.

3.1.12 Configure cluster Coherence cluster settings

To configure Coherence settings for a cluster:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Coherence**.

   From the Coherence page, you can select the Coherence cluster you want to use for this cluster and define settings such as:

   - Local Storage Enabled
   - Coherence Web Local Storage Enabled

   For more information about these fields, see Configuration Options.
5. Click **Save**.

3.1.13 Create cluster notes

To create notes for cluster configuration:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. In the **Target Navigation** pane, select the cluster you want to configure.
4. From the **WebLogic Cluster** menu, select **Administration**, then select **Notes**.
5. On the Notes page, enter your notes.

6. Click **Save**.

For more information, see Configuration Options.

4 WebLogic Server Machines

A machine is the logical representation of the computer that hosts one or more WebLogic Server instances. Each Managed Server must be assigned to a machine. The Administration Server uses the machine definition in conjunction with Node Manager to start remote servers.

Node Manager is a WebLogic Server utility that enables you to start, shut down, and restart Administration Server and Managed Server instances from a remote location. Although Node Manager is optional, Oracle recommends using it if your WebLogic Server environment hosts applications with high availability requirements because Node Manager allows you to control the running state of distributed server instances from a centralized location.

The Java implementation of Node Manager is configured by default to control all server instances belonging to the same domain, a per domain Node Manager. The server instances need not reside on the same machine.

4.1 Monitor machines

To monitor the machines configured in the current domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware.**
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic Domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Machines**.

   The Machines table displays information about the machines that have been configured in the current WebLogic Server domain, such as:

   - Name
   - Type

   For more information about these fields, see Configuration Options.

4.2 Configure machines

This section describes how to configure machines. This section includes the following tasks:

- Configure general machine settings
- Configure general Unix machine settings
- Configure Node Manager settings

- Configure machine server settings
- Add or remove server instances
- Create machine notes

4.2.1 Configure general machine settings

To monitor general configuration settings for a machine:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Machines**.

   The Machines table displays information about the machines that have been configured in the current WebLogic Server domain.

   For more information, see Configuration Options.

   Optionally, select **View** to access the following table options:

   - Columns: add or remove the columns displayed in the table
   - Detach: detach the table (viewing option)
   - Query by Example
4. In the table, select the name of the machine for which you want to view general configuration settings.
5. Select **Configuration**, then select **General**.

   From the General page, you can monitor the following configuration information for a machine:

   - Name
   - Type

   For more information about these fields, see Configuration Options.

4.2.2 Configure general Unix machine settings

To monitor general configuration settings for a Unix machine:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Machines**.

   The Machines table displays information about the machines that have been configured in the current WebLogic Server domain.

   For more information, see Configuration Options.

   Optionally, select **View** to access the following table options:

   - Columns: add or remove the columns displayed in the table
   - Detach: detach the table (viewing option)

- Query by Example

4. In the table, select the name of the Unix machine for which you want to view general configuration settings.
5. Select **Configuration**, then select **General**.

   From the General page, you can configure the following attributes for a Unix machine:

   - Type
   - Name
   - Enable Post-Bind UID
   - Post-Bind UID
   - Enable Post-Bind GID
   - Post-Bind GID

   For more information about these fields, see Configuration Options.

4.2.3 Configure Node Manager settings

To configure Node Manager settings for a machine:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Machines**.

   The Machines table displays information about the machines that have been configured in the current WebLogic Server domain.

   For more information, see Configuration Options.

   Optionally, select **View** to access the following table options:

   - Columns: add or remove the columns displayed in the table
   - Detach: detach the table (viewing option)
   - Query by Example
4. In the table, select the name of the machine for which you want to configure Node Manager settings.
5. Select **Configuration**, then select **Node Manager**.

   From the Node Manager page, you can configure Node Manager settings, such as:

   - Type
   - Listen Address
   - Listen Port
   - Node Manager Home
   - Shell Command
   - Debug Enabled

   For more information about these fields, see Configuration Options.

4.2.4 Configure machine server settings

To configure server settings for a machine:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Machines**.

   The Machines table displays information about the machines that have been configured in the current WebLogic Server domain.

   For more information, see Configuration Options.

   Optionally, select **View** to access the following table options:

   - Columns: add or remove the columns displayed in the table
   - Detach: detach the table (viewing option)
   - Query by Example
4. In the table, select the name of the machine for which you want to view server configuration information.
5. Select **Configuration**, then select **Servers**.

   From the Servers page, you can add or remove servers from the machine. You can also view configuration information for the servers associated with the machine, such as:

   - Name
   - Type
   - Cluster
   - Machine
   - State
   - Health
   - Listen Port

   For more information about these fields, see Configuration Options.

4.2.5 Add or remove server instances

To add or remove server instances from a machine:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Machines**.

   The Machines table displays information about the machines that have been configured in the current WebLogic Server domain.

   For more information, see Configuration Options.

   Optionally, select **View** to access the following table options:

- Columns: add or remove the columns displayed in the table
- Detach: detach the table (viewing option)
- Query by Example

4. In the table, select the name of the machine for which you want to add or remove server instances.
5. Select **Configuration**, then select **Servers**.
6. Click **Add/Remove**.
7. From the Add/Remove Server page, you can add or remove server instances from this machine by moving server instances between the Available and Chosen lists. Only server instances that are shut down and statically configured may be added or removed from a machine using this page.

   For more information, see Configuration Options.
8. Click **Save**.

4.2.6 Create machine notes

To create notes for machines:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Machines**.
4. In the Machines table, select the name of the machine for which you want to create notes.
5. Select **Notes**.
6. On the Notes page, enter your notes.
7. Click **Save**.

For more information, see Configuration Options.

5 WebLogic Server JDBC Data Sources

This section describes how to create, monitor, control, and configure data sources deployed to the current domain. Java Database Connectivity (JDBC) enables you to configure database connectivity through JDBC data sources in your WebLogic domain. A data source is a Java EE standard method of configuring connectivity to a database. Each WebLogic data source contains a pool of database connections.

Applications look up the data source on the JNDI tree or in the local application context and then reserve a database connection with the `getConnection` method. Data sources and their connection pools provide connection management processes that help keep your system running efficiently.

This section includes the following topics:

- Create JDBC data sources
- Monitor JDBC data sources
- Control JDBC data sources
- Configure JDBC data sources

5.1 Create JDBC data sources

You must create a data source for each database to which you want to connect. If you need more than one set of configuration options for a database, you can create more than one data source that includes connections to the same database. This section includes the following tasks:

- Create JDBC generic data sources
- Create JDBC GridLink data sources
- Create JDBC multi data sources
- Create a JDBC data source from an existing data source

5.1.1 Create JDBC generic data sources

Before you begin

Make sure that the JDBC drivers that you want to use to create database connections are installed on all server instances on which you want to deploy the data source. Some JDBC drivers are installed with WebLogic Server, including Oracle Type 4 JDBC drivers for DB2, Informix, MS SQL Server, and Sybase.

To create a JDBC generic data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.

2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.

   The JDBC Data Sources page is displayed.
4. Click **Create**, then select **Generic Data Source**.
5. Define the configuration options for your JDBC data source on each of the following pages:
   - Data Source Properties
   - Connection Properties
   - Transaction Properties
   - Select Targets
   - Review
6. Click **Create** to save the JDBC data source configuration and deploy the data source to the targets that you selected.

5.1.1.1 Data Source Properties

On the **Data Sources Properties** page, define the general configuration options for this JDBC data source.

- **Data Source Name**: enter a name for this JDBC data source. This name is used in the configuration file (`config.xml`) and whenever referring to this data source.
- **Driver Class Name**: select the JDBC driver you want to use to connect to the database. The list includes common JDBC drivers for the selected DBMS.

> **Note:** You must install JDBC drivers before you can use them to create database connections. Some JDBC drivers are installed with WebLogic Server, but many are not installed.

- **JNDI Name**: enter the JNDI path to where this JDBC data source will be bound. Applications look up the data source on the JNDI tree by this name when reserving a connection.
- **Row Prefetch Enabled**: select to enable multiple rows to be "prefetched" (that is, sent from the server to the client) in one server access.
- **Row Prefetch Size**: if you enabled row prefetching, specify the number of result set rows to prefetch for a client.
- **Stream Chunk Size**: specify the data chunk size for steaming data types.

For more information, see Configuration Options.

5.1.1.2 Connection Properties

On the **Connection Properties** page:

1. On the **Connection Properties** page, enter values for the following properties in the **Database Connection Information** section:
   - **Database URL**: enter the URL of the database to connect to. The format of the URL varies by JDBC driver.

- **Password**: enter the database account password to use to create database connections.
- **Test Table Name or SQL Statement**: enter the name of the database table or SQL statement to use to test physical database connections.

  This name is required when you specify a **Test Frequency** and enable **Test Reserved Connections**. The default SQL code used to test a connection is `select count(*) from TestTableName`. If the Test Table Name begins with `SQL`, then the rest of the string following that leading token will be taken as a literal SQL statement that will be used to test connections instead of the standard query. For example: `SELECT 1 FROM DUAL`.

2. In the **Properties** section, click **Add** and enter the properties that are required by the data source class. For example: `server=dbserver1`.
3. In the **System Properties** section, click **Add** and enter the system properties that are required by the data source class. For example: `server=dbserver1`.
4. In the **Connection Pool Properties** section, enter values for the following properties:
   - **Initial Capacity**: enter the number of physical connections to create when creating the connection pool.
   - **Maximum Capacity**: enter the maximum number of physical connections that this connection pool can contain.
   - **Capacity Increment**: enter the number of connections created when new connections are added to the connection pool.
   - **Statement Cache Type**: select the algorithm used for maintaining the prepared statements stored in the statement cache.
   - **Statement Cache Size**: enter the number of prepared and callable statements stored in the cache. (This may increase server performance.)
5. In the **Advanced** section, enter values for the following properties:
   - **Test Connections On Reserve**: select whether to enable WebLogic Server to test a connection before giving it to a client. (Requires that you specify a **Test Table Name**.)
   - **Test Frequency (seconds)**: enter the number of seconds between when WebLogic Server tests unused connections. (Requires that you specify a **Test Table Name**.) Connections that fail the test are closed and reopened to re-establish a valid physical connection. If the test fails again, the connection is closed.
   - **Seconds to Trust an Idle Pool Connection**: enter the number of seconds within a connection use that WebLogic Server trusts that the connection is still viable and will skip the connection test, either before delivering it to an application or during the periodic connection testing process.
   - **Shrink Frequency (seconds)**: enter the number of seconds to wait before shrinking a connection pool that has incrementally increased to meet demand.
   - **Init SQL**: enter the SQL statement to execute that will initialize newly created physical database connections. Start the statement with SQL followed by a space.
   - **Connection Creation Retry Frequency (seconds)**: enter the number of seconds between attempts to establish connections to the database.

- **Login Delay (seconds)**: enter the number of seconds to delay before creating each physical database connection. This delay supports database servers that cannot handle multiple connection requests in rapid succession.
- **Inactive Connection Timeout (seconds)**: enter the number of inactive seconds on a reserved connection before WebLogic Server reclaims the connection and releases it back into the connection pool.
- **Maximum Waiting for Connection**: enter the maximum number of connection requests that can concurrently block threads while waiting to reserve a connection from the data source's connection pool.
- **Connection Reserve Timeout (seconds)**: enter the number of seconds after which a call to reserve a connection from the connection pool will timeout.
- **Statement Timeout**: enter the time after which a statement currently being executed will time out.
- **Ignore In-Use Connections**: enables the data source to be shutdown even if connections obtained from the pool are still in use.
- **Pinned-To-Thread**: can improve performance by enabling execute threads to keep a pooled database connection even after the application closes the logical connection.
- **Remove Infected Connections Enabled**: specifies whether a connection will be removed from the connection pool after the application uses the underlying vendor connection object.
- **Wrap Data Types**: specifies whether wrapping is enabled.
- **Fatal Error Codes**: specifies a comma-separated list of error codes that are treated as fatal errors.
- **Connection Labeling Callback**: enter the class name of the connection labeling callback.
- **Connection Harvest Max Count**: enter the maximum number of connections that may be harvested when the connection harvesting occurs.
- **Connection Harvest Trigger Count**: specifies the number of available connections (trigger value) used to determine when connection harvesting occurs.
- **Connection Count of Refresh Failures Till Disable**: specifies the number of reconnect failures allowed before WebLogic Server disables a connection pool to minimize the delay in handling the connection request caused by a database failure.
- **Count of Test Failures Till Flush**: specifies the number of test failures allowed before WebLogic Server closes all unused connections in a connection pool to minimize the delay caused by further database testing.

For more information, see Configuration Options.

5.1.1.3 Transaction Properties

On the **Transaction Properties** page, follow these steps. Depending on the driver you selected on the **JDBC Data Source Properties** page, you may not need to specify any of these options.

Supports Global Transactions: select this check box (the default) to enable global transaction support in this data source. Clear this check box to disable (ignore) global transactions in this data source. In most cases, you should leave the option selected.

If you selected **Supports Global Transactions**, select an option for transaction processing (available options vary depending on whether you select an XA driver or a non-XA driver):

- **One-Phase Commit**: select this option to enable the non-XA connection to participate in a global transaction as the only transaction participant. This option is only available when you select a non-XA JDBC driver to make database connections.
- **Emulate Two-Phase Commit**: enables a non-XA JDBC connection to emulate participation in distributed transactions using JTA. Select this option only if your application can tolerate heuristic conditions. This option is only available when you select a non-XA JDBC driver to make database connections.
- **Logging Last Resource**: select this option to enable a non-XA JDBC connection to participate in global transactions using the Logging Last Resource (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit. This option is only available when you select a non-XA JDBC driver to make database connections.

For more information, see Configuration Options.

5.1.1.4 Select Targets

On the **Select Targets** page, select the server instances and clusters on which you want to deploy the data source.

For more information, see Configuration Options.

5.1.1.5 Review

On the **Review** page, review the configuration for this JDBC data source.

For more information, see Configuration Options.

5.1.2 Create JDBC GridLink data sources

Configure database connectivity with your Oracle RAC installation by adding a JDBC GridLink data source to your WebLogic domain. Data sources and their connection pools provide connection management processes that help keep your system running efficiently.

To create a JDBC GridLink data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.

   The JDBC Data Sources page is displayed.
4. Click **Create**, then select **GridLink Data Source**.
5. Define the configuration options for your JDBC GridLink data source on each of the following pages:

   - Data Source Properties
   - Connection Properties
   - Transaction Properties
   - ONS Properties

- Select Targets
- Review

6. Click **Create** to save the JDBC GridLink data source configuration and deploy the GridLink data source to the targets that you selected.

5.1.2.1 Data Source Properties

On the **Data Sources Properties** page, define the general configuration options for this JDBC data source.

- **Data Source Name**: enter a name for this JDBC data source. This name is used in the configuration file (`config.xml`) and whenever referring to this data source.
- **Driver Class Name**: select the JDBC driver you want to use to connect to the database. The list includes common JDBC drivers for the selected DBMS.

> **Note:** You must install JDBC drivers before you can use them to create database connections. Some JDBC drivers are installed with WebLogic Server, but many are not installed.

- **JNDI Name**: enter the JNDI path to where this JDBC data source will be bound. Applications look up the data source on the JNDI tree by this name when reserving a connection.
- **Row Prefetch Enabled**: select to enable multiple rows to be "prefetched" (that is, sent from the server to the client) in one server access.
- **Row Prefetch Size**: if you enabled row prefetching, specify the number of result set rows to prefetch for a client.
- **Stream Chunk Size**: specify the data chunk size for steaming data types.

For more information, see Configuration Options.

5.1.2.2 Connection Properties

On the **Connection Properties** page:

1. On the **Connection Properties** page, enter values for the following properties in the **Database Connection Information** section:
   - **Database URL**: enter the URL of the database to connect to. The format of the URL varies by JDBC driver.
   - **Password**: enter the database account password to use to create database connections.
   - **Test Table Name or SQL Statement**: enter the name of the database table or SQL statement to use to test physical database connections.

     This name is required when you specify a **Test Frequency** and enable **Test Reserved Connections**. The default SQL code used to test a connection is `select count(*) from TestTableName`. If the Test Table Name begins with `SQL`, then the rest of the string following that leading token will be taken as a literal SQL statement that will be used to test connections instead of the standard query. For example: `SELECT 1 FROM DUAL`.
2. In the **Properties** section, click **Add** and enter the properties that are required by the data source class. For example: `server=dbserver1`.

3. In the **System Properties** section, click **Add** and enter the system properties that are required by the data source class. For example: `server=dbserver1`.

4. In the **Connection Pool Properties** section, enter values for the following properties:

   - **Initial Capacity**: enter the number of physical connections to create when creating the connection pool.
   - **Maximum Capacity**: enter the maximum number of physical connections that this connection pool can contain.
   - **Capacity Increment**: enter the number of connections created when new connections are added to the connection pool.
   - **Statement Cache Type**: select the algorithm used for maintaining the prepared statements stored in the statement cache.
   - **Statement Cache Size**: enter the number of prepared and callable statements stored in the cache. (This may increase server performance.)

5. In the **Advanced** section, enter values for the following properties:

   - **Test Connections On Reserve**: select whether to enable WebLogic Server to test a connection before giving it to a client. (Requires that you specify a **Test Table Name**.)
   - **Test Frequency (seconds)**: enter the number of seconds between when WebLogic Server tests unused connections. (Requires that you specify a **Test Table Name**.) Connections that fail the test are closed and reopened to re-establish a valid physical connection. If the test fails again, the connection is closed.
   - **Seconds to Trust an Idle Pool Connection**: enter the number of seconds within a connection use that WebLogic Server trusts that the connection is still viable and will skip the connection test, either before delivering it to an application or during the periodic connection testing process.
   - **Shrink Frequency (seconds)**: enter the number of seconds to wait before shrinking a connection pool that has incrementally increased to meet demand.
   - **Init SQL**: enter the SQL statement to execute that will initialize newly created physical database connections. Start the statement with SQL followed by a space.
   - **Connection Creation Retry Frequency (seconds)**: enter the number of seconds between attempts to establish connections to the database.
   - **Login Delay (seconds)**: enter the number of seconds to delay before creating each physical database connection. This delay supports database servers that cannot handle multiple connection requests in rapid succession.
   - **Inactive Connection Timeout (seconds)**: enter the number of inactive seconds on a reserved connection before WebLogic Server reclaims the connection and releases it back into the connection pool.
   - **Maximum Waiting for Connection (seconds)**: enter the maximum number of connection requests that can concurrently block threads while waiting to reserve a connection from the data source's connection pool.
   - **Connection Reserve Timeout (seconds)**: enter the number of seconds after which a call to reserve a connection from the connection pool will timeout.

- **Statement Timeout**: enter the time after which a statement currently being executed will time out.
- **Ignore In-Use Connections**: enables the data source to be shutdown even if connections obtained from the pool are still in use.
- **Pinned-To-Thread**: can improve performance by enabling execute threads to keep a pooled database connection even after the application closes the logical connection.
- **Remove Infected Connections Enabled**: specifies whether a connection will be removed from the connection pool after the application uses the underlying vendor connection object.
- **Wrap Data Types**: specifies whether wrapping is enabled.
- **Fatal Error Codes**: specifies a comma-separated list of error codes that are treated as fatal errors.
- **Connection Labeling Callback**: enter the class name of the connection labeling callback.
- **Connection Harvest Max Count**: enter the maximum number of connections that may be harvested when the connection harvesting occurs.
- **Connection Harvest Trigger Count**: specifies the number of available connections (trigger value) used to determine when connection harvesting occurs.
- **Connection Count of Refresh Failures Till Disable**: specifies the number of reconnect failures allowed before WebLogic Server disables a connection pool to minimize the delay in handling the connection request caused by a database failure.
- **Count of Test Failures Till Flush**: specifies the number of test failures allowed before WebLogic Server closes all unused connections in a connection pool to minimize the delay caused by further database testing.

For more information, see Configuration Options.

5.1.2.3 Transaction Properties

On the **Transaction Properties** page, follow these steps. Depending on the driver you selected on the **JDBC Data Source Properties** page, you may not need to specify any of these options.

Supports Global Transactions: select this check box (the default) to enable global transaction support in this data source. Clear this check box to disable (ignore) global transactions in this data source. In most cases, you should leave the option selected.

If you selected **Supports Global Transactions**, select an option for transaction processing (available options vary depending on whether you select an XA driver or a non-XA driver):

- **One-Phase Commit**: select this option to enable the non-XA connection to participate in a global transaction as the only transaction participant. This option is only available when you select a non-XA JDBC driver to make database connections.
- **Emulate Two-Phase Commit**: enables a non-XA JDBC connection to emulate participation in distributed transactions using JTA. Select this option only if your application can tolerate heuristic conditions. This option is only available when you select a non-XA JDBC driver to make database connections.

- **Logging Last Resource**: select this option to enable a non-XA JDBC connection to participate in global transactions using the Logging Last Resource (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit. This option is only available when you select a non-XA JDBC driver to make database connections.

For more information, see Configuration Options.

5.1.2.4 ONS Properties

On the **ONS Properties** page, enter values for the following properties:

- Select **Fan Enabled** to subscribe to Oracle Fan Events.
- Under **ONS Nodes**, click **Add** and enter the ONS host and port for each ONS node.
- To test individual nodes, click **Test ONS Node** for an ONS host and port.
- Optionally, configure an ONS wallet file if you want ONS to use SSL protocol.

For more information, see Configuration Options.

5.1.2.5 Select Targets

On the **Select Targets** page, select the server instances and clusters on which you want to deploy the data source.

For more information, see Configuration Options.

5.1.2.6 Review

On the **Review** page, review the configuration for this JDBC GridLink data source.

For more information, see Configuration Options.

5.1.3 Create JDBC multi data sources

Multi data sources provide failover and load balancing for connection requests between two or more data sources. Before you create a multi data source, you should create the data sources that the multi data source will manage, and deploy them to the same targets to which you want to deploy the multi data source.

To create a JDBC multi data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.

   The JDBC Data Sources page is displayed.
4. Click **Create**, then select **Multi Data Source**.
5. Define the configuration options for your JDBC multi data source on each of the following pages:
   - Configure Data Source Properties
   - Select Targets
   - Select Data Source Type
   - Add Data Sources

   - Review

6. Click **Create** to save the JDBC multi data source configuration and deploy the multi data source to the targets that you selected.

5.1.3.1 Configure Data Source Properties

On the **Configure Data Sources Properties** page, define the general configuration options for this JDBC multi data source.

- **Data Source Name**: enter a name for this JDBC multi data source. This name is used in the configuration files (`config.xml` and the JDBC module) and whenever referring to this multi data source.
- **JNDI Name**: enter the JNDI path to where this JDBC multi data source will be bound. Applications look up the data source on the JNDI tree by this name when reserving a connection.
- **Algorithm Type**: select an algorithm option:
  - **Failover**: The multi data source routes connection requests to the first data source in the list; if the request fails, the request is sent to the next data source in the list, and so forth.
  - **Load-Balancing**: The multi data source distributes connection requests evenly to its member data sources.

For more information, see Configuration Options.

5.1.3.2 Select Targets

On the **Select Targets** page, select the server instances or clusters on which you want to deploy this JDBC multi data source.

The targets you select will limit the data sources that you can select as part of the multi data source. You can only select data sources that are deployed to the same targets as the multi data source.

For more information, see Configuration Options.

5.1.3.3 Select Data Source Type

On the **Select Data Source Type** page, select one of the following options:

- **XA Driver**: The multi data source will only use data sources that use an XA JDBC driver to create database connections.
- **Non-XA Driver**: The multi data source will only use data sources that use a non-XA JDBC driver to create database connections.

The option you select limits the data sources that you can select as part of the multi data source in a later step. Limiting data sources by JDBC driver type enables the WebLogic Server transaction manager to properly complete or recover global transactions that use a database connection from a multi data source.

For more information, see Configuration Options.

5.1.3.4 Add Data Sources

On the **Add Data Sources** page, select the data sources that you want the multi data source to use to satisfy connection requests.

For more information, see Configuration Options.

5.1.3.5 Review

On the **Review** page, review the configuration for this JDBC multi data source.

For more information, see Configuration Options.

5.1.4 Create a JDBC data source from an existing data source

You can create a new JDBC data source to have the same configuration settings as an existing JDBC data source.

To create a new JDBC data source from an existing JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the row of the data source instance you want to use to create your new data source.

   The Create Like option is displayed above the table.
5. Click **Create Like**.

   The Creating New JDBC Data Source assistant is displayed with the same configuration settings as the existing JDBC data source you modeled.
6. Review each page of the Creating New JDBC Data Source assistant to ensure the property values match the desired configuration for your new JDBC data source.
7. Click **Create** to save the JDBC data source configuration and deploy the new data source to the targets that you selected.

5.2 Monitor JDBC data sources

After you create a JDBC data source, you can monitor it to look for unusual activity, such as an abnormal number of requests waiting for a connection. You can also test the connection between a data source and the database. This section includes the following tasks:

- Monitor JDBC data sources
- Monitor a JDBC data source
- Monitor a JDBC GridLink data source
- Monitor GridLink data source details
- Monitor a JDBC multi data source
- Test JDBC data sources

5.2.1 Monitor JDBC data sources

You can monitor a variety of statistics for each data source instance in your domain, such as the current number of database connections in the connection pool, current number of connections in use, the longest wait time for a database connection, and so forth.

To monitor the activity of the JDBC data source instances deployed to the current domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.

   The JDBC Data Sources table displays statistics about the JDBC data source instances deployed to the current domain, such as:

   - Name
   - Type
   - Resource
   - Server Name
   - State

   For more information about these fields, see Configuration Options.

   Optionally, select **View** to access the following table options:

   - Columns: add or remove the columns displayed in the table
   - Detach: detach the table (viewing option)
   - Reorder: change the order of the columns displayed
   - Query by Example
4. In the table, select the name of the JDBC data source for which you want to view configuration information.

5.2.2 Monitor a JDBC data source

To monitor the activity of a specific JDBC data source instance deployed to the current domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to monitor.
5. Select **Monitoring**.
6. The Monitor JDBC Data Source page displays statistics about this JDBC data source instance, such as:

   - Type
   - Resource
   - Scope
   - Server Name
   - State

   For more information about these fields, see Configuration Options.

   Optionally, select **View** to access the following table options:

   - Columns: add or remove the columns displayed in the table

- Detach: detach the table (viewing option)
- Reorder: change the order of the columns displayed
- Query by Example

5.2.3 Monitor a JDBC GridLink data source

To monitor the activity of a specific JDBC GridLink data source instance deployed to the current domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC GridLink data source you want to monitor.
5. Select **Monitoring**.

   The Monitoring page displays statistics about this JDBC GridLink data source.

5.2.3.1 Monitor GridLink data source details

To monitor details associated with a GridLink data source instance or ONS client:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC GridLink data source you want to monitor.
5. Select **Monitoring**.

   The Monitoring page displays statistics about this JDBC GridLink data source.
6. In the Monitoring table, select the instance or ONS client you want to monitor.

   The Instances page displays statistics about this GridLink data source instance, such as:

   - Instance Name
   - State
   - Current Capacity
   - Number Available
   - Active Connections Current Count

   For more information about these fields, see Configuration Options.

   The ONS page displays statistics about this GridLink data source ONS client, such as:

   - Host
   - Port
   - Status

For more information about these fields, see Configuration Options.

5.2.4 Monitor a JDBC multi data source

To monitor the activity of a specific JDBC multi data source instance deployed to the current domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC multi data source you want to monitor.
5. Select **Monitoring**.
6. The Monitor JDBC Multi Data Source page displays statistics about all the JDBC data sources participating in this JDBC multi data source, such as:

   - Name
   - Type
   - Resource
   - Server Name
   - State

   For more information about these fields, see Configuration Options.

   Optionally, select **View** to access the following table options:

   - Columns: add or remove the columns displayed in the table
   - Detach: detach the table (viewing option)
   - Reorder: change the order of the columns displayed
   - Query by Example

5.2.5 Test JDBC data sources

You can manually test individual instances of a data source. When you test a data source, WebLogic Server reserves a connection from the data source, tests it using the standard testing query or the query specified in Test Table Name, and then returns the database connection to the pool of connections. Test results are displayed at the top of the page.

The manual connection test relies on the Test Reserved Connections and Test Table Name attributes of the data source. Those attributes are set by default. However, if you changed either attribute, the changes will affect the database connection test.

To test a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to monitor.

5. Select **Monitoring**.
6. Click **Test Data Source**. Test results are displayed.

5.3 Control JDBC data sources

After you create a JDBC data source, you can perform administrative tasks on instances of the data source, including resetting all database connections, suspending use of the data source, and shutting down the data source. This section includes the following tasks:

- Start JDBC data sources
- Stop JDBC data sources
- Resume suspended JDBC data sources
- Suspend JDBC data sources
- Shrink JDBC data source connection pools
- Reset JDBC data source connections
- Clear JDBC data source statement caches
- Delete JDBC data sources
- Control a JDBC data source
- Control a JDBC multi data source

5.3.1 Start JDBC data sources

You can manually start data source instances that have a health state of `Shutdown`.

To start a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to control.
5. Select **Control**.
6. In the Control table, select the row of the data source instance you want to start.

   The control options are displayed above the table.
7. Click **Start**, then select **Yes** to confirm the action.

5.3.2 Stop JDBC data sources

You can manually stop individual instances of a data source. When you stop a data source, behavior depends on the type of stop that you select:

- **Stop:** shuts down a data source that has a health state of `Running`. If any connections from the data source are currently in use, the `Shutdown` operation fails and the health state remains `Running`.
- **Force Stop**: shuts down a data source that has a health state of `Running`, including forcing the disconnection of all current connection users.

To stop a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to control.
5. Select **Control**.
6. In the Control table, select the row of the data source instance you want to stop.

   The control options are displayed above the table.
7. Click **Stop** or **Force Stop**, and then select **Yes** to confirm the action.

5.3.3 Resume suspended JDBC data sources

You can manually resume data source instances that are in a `Suspended` state.

To resume a suspended JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to control.
5. Select **Control**.
6. In the Control table, select the row of the data source instance you want to resume.

   The control options are displayed above the table.
7. Click **Resume**, then select **Yes** to confirm the action.

5.3.4 Suspend JDBC data sources

You can manually suspend individual instances of a data source. When you suspend a data source, applications can no longer get a database connection from the data source. For connections that are already reserved by an application, behavior depends on the type of suspension that you select:

- **Suspend:** marks the data source as disabled and blocks any new connection requests. If there are any reserved connections, the operation will wait for `InactiveTimeout` seconds, if configured. Otherwise, the operation waits 60 seconds before suspending all connections. If successful, the health state is set to `Suspended`.
- **Force Suspend:** marks the data source as disabled, blocks any new requests for a connection from the connection pool, and closes and recreates connections currently in use.

Most connections in a suspended data source remain intact. The connections are not recreated when you resume the data source, except for the connections in use when the data source is Force Suspended.

To suspend a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.

2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to control.
5. Select **Control**.
6. In the Control table, select the row of the data source instance you want to suspend.

   The control options are displayed above the table.
7. Click **Suspend** or **Force Suspend**, then select **Yes** to confirm the action.

5.3.5 Shrink JDBC data source connection pools

You can manually shrink the pool of database connections in individual instances of a data source to the initial capacity or the current number of connections in use, whichever is greater.

To shrink the connection pool in a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to control.
5. Select **Control**.
6. In the Control table, select the row of the data source instance for which you want to shrink the connection pool.

   The control options are displayed above the table.
7. Click **Shrink**, then select **Yes** to confirm the action.

5.3.6 Reset JDBC data source connections

When you reset the database connections in a JDBC data source, WebLogic Server closes and recreates all available database connections in the pool of connections in the data source.

To reset database connections in a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to control.
5. Select **Control**.
6. In the Control table, select the row of the data source instance for which you want to reset connections.

   The control options are displayed above the table.
7. Click **Reset**, then select **Yes** to confirm the action.

5.3.7 Clear JDBC data source statement caches

If statement caching is enabled for a data source, WebLogic Server caches prepared and callable statements that are used in each connection in the data source. Each connection has its own cache, but the caches for each connection are configured and managed as a group. When you clear the statement cache for a data source, you clear the statement cache for all connections in the instance of the data source you select.

To clear the statement cache in a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to control.
5. Select **Control**.
6. In the Control table, select the row of the data source instance for which you want to clear the statement cache.

   The control options are displayed above the table.
7. Click **Clear Statement Cache**, then select **Yes** to confirm the action.

5.3.8 Delete JDBC data sources

Before you begin

Ensure that the data source you want to delete is not used by a multi data source. If the data source you want to delete is used by a multi data source, you must remove the data source from the multi data source before the data source can be deleted. The delete operation will fail if the data source you are attempting to delete is used by a multi data source.

To delete a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the row of the JDBC data source you want to delete.

   The Delete option is displayed above the table.
5. Click **Delete**, then select **Yes** to confirm the action.

5.3.9 Control a JDBC data source

To control a specific JDBC data source instance deployed to the current domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to control.

5. Select **Control**.
6. From the Control JDBC Data Source page you can perform the following actions:
   - Start
   - Stop
   - Force Stop
   - Resume
   - Suspend
   - Force Suspend
   - Shrink Connection Pools
   - Reset
   - Clear Statement Cache

   For more information about these fields, see Configuration Options.

   Optionally, select **View** to access the following table options:
   - Columns: add or remove the columns displayed in the table
   - Detach: detach the table (viewing option)
   - Reorder: change the order of the columns displayed
   - Query by Example

5.3.10 Control a JDBC multi data source

To control a specific JDBC multi data source instance deployed to the current domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC multi data source you want to control.
5. Select **Control**.
6. From the Control JDBC Multi Data Source page you can perform the following actions:
   - Start
   - Stop
   - Force Stop
   - Resume
   - Suspend
   - Force Suspend
   - Shrink Connection Pools
   - Reset
   - Clear Statement Cache

For more information about these fields, see Configuration Options.

Optionally, select **View** to access the following table options:

- Columns: add or remove the columns displayed in the table
- Detach: detach the table (viewing option)
- Reorder: change the order of the columns displayed
- Query by Example

5.4 Configure JDBC data sources

When you create a JDBC data source, most data source attributes are configured so that the data source will work in your environment. However, you may need to modify a data source configuration to enable or disable specific features or to tune performance.

This section includes the following tasks:

- Define general configuration settings
- Bind a JDBC data source to the JNDI tree with multiple names
- Configure connection pool properties
- Configure JDBC data source testing options
- Configure statement cache
- Configure connection pool capacity
- Enable connection requests to wait for a connection
- Configure Oracle parameters
- Configure ONS client parameters
- Configure SSL for the ONS client using a Oracle wallet file
- Configure global transaction options
- Configure JDBC data source diagnostic profiling
- Configure JDBC data source identity options
- Enable identity-based connection pooling
- Target JDBC data sources
- Configure a JDBC multi data source

5.4.1 Define general configuration settings

Applications connect to databases from a data source by looking up the data source on the Java Naming and Directory Interface (JNDI) tree and then requesting a connection. The data source provides the connection to the application from its pool of data base connections.

To define general configuration settings for a specific JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.

4. In the JDBC Data Source table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **General**.
6. From the General Configuration page, you can define configuration settings for this JDBC data source, such as:
   - Data Source Name
   - Type
   - Database Type
   - Driver Class Name
   - JNDI Name
   - Row Prefetch Enabled
   - Row Prefetch Size
   - Stream Chunk Size

   For more information about these fields, see Configuration Options.
7. Click **Save**.

After you finish

After you activate your changes, you will need to redeploy the data source or restart your server before the changes will take effect.

5.4.1.1 Bind a JDBC data source to the JNDI tree with multiple names

Before you begin

You can configure a data source so that it binds to the JNDI tree with multiple names. You can use a multi-JNDI-named data source in place of legacy configurations that included multiple data sources that pointed to a single JDBC connection pool. You must either restart the system after making your change or undeploy the data source before making the change, and then redeploy it after making the change.

To add JNDI names to an existing JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **General**.
6. In the **JNDI Name** field, enter the names you want to use to bind the data source to the JNDI tree with each name on a separate line.

   For more information about these fields, see Configuration Options.
7. Click **Save**.

After you finish

After you activate your changes, you will need to redeploy the data source or restart your server before the changes will take effect.

5.4.2 Configure connection pool properties

The connection pool within a JDBC data source contains a group of JDBC connections that applications reserve, use, and then return to the pool. The connection pool and the connections within it are created when the connection pool is registered, usually when starting WebLogic Server or when deploying the data source to a new target.

To configure the connection pool for a specific JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **Connection Pool**.
6. From the Connection Pool page, you can define connection properties for this JDBC data source, such as:
   - Data Source Name
   - Type
   - Driver Class Name
   - Database URL
   - Password
   - Confirm Password
   - Properties
   - System Properties
   - Initial Capacity
   - Maximum Capacity
   - Capacity Increment
   - Statement Cache Type
   - Statement Cache Size

   For more information about these fields, see Configuration Options.
7. Optionally, expand **Advanced** to define advanced connection properties for this JDBC data source.
8. Click **Save**.

5.4.2.1 Configure JDBC data source testing options

You can set database connection testing options in a data source to make sure that the database connections remain healthy, which helps keep your applications running properly.

To configure testing options for a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.

4. In the JDBC Data Sources table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **Connection Pool**.
6. Expand **Advanced** to display the advanced connection pool options.
7. Select one or more of the following options:

   - **Test Connections on Reserve**: select this check box to test the database connection before giving it to your application when your application requests a connection from the data source.
   - **Test Frequency**: enable periodic background connection testing by entering the number of seconds between periodic tests.

   You can use these options to achieve the right mix of performance and fault tolerance for your system.
8. In the **Test Table Name** field, enter the name of a small table to use in a query to test database connections. The standard query is `select 1 from table_name`. If you prefer to use a different query as a connection test, enter `SQL` followed by a space and the SQL code you want to use to test database connections.
9. Optionally, in the **Seconds to Trust an Idle Pool Connection** field, enter the number of seconds within which, if the database connection has been used or tested, WebLogic Server will skip the connection test. This option can help reduce the overhead of connection testing and improve application performance.

   For more information about these fields, see Configuration Options.
10. Click **Save**.

5.4.2.2 Configure statement cache

To improve performance, WebLogic Server can cache prepared and callable statements used in your applications (enabled by default). When an application or EJB calls any of the statements stored in the cache, WebLogic Server reuses the statement stored in the cache. Each database connection in a data source has its own statement cache.

To configure the statement cache for a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **Connection Pool**.
6. In the **Statement Cache Type** field, select one of the following options:

   - **LRU**: after the Statement Cache Size is met, the Least Recently Used statement is removed when a new statement is used.
   - **Fixed**: the first Statement Cache Size number of statements is stored and stay fixed in the cache. No new statements are cached unless the cache is manually cleared or the cache size is increased.
7. In the **Statement Cache Size** field, enter the number of statements to cache per connection per data source instance.

   For more information about these fields, see Configuration Options.
8. Click **Save**.

5.4.2.3 Configure connection pool capacity

You can configure the initial and maximum capacity for a JDBC connection pool.

To configure the connection capacity for a JDBC connection pool:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **Connection Pool**.
6. In the **Initial Capacity** field, enter the number of physical connections to create when creating the connection pool.
7. In the **Maximum Capacity** field, enter the maximum number of physical connections that this connection pool can contain.

   > **Note:** An easy way to boost performance of JDBC in WebLogic Server applications is to set the value of **Initial Capacity** equal to the value for **Maximum Capacity** when configuring connection pools in your data source.

   For more information about these fields, see Configuration Options.
8. Click **Save**.

5.4.2.4 Enable connection requests to wait for a connection

To enable connection requests to wait for a connection from a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **Connection Pool**.
6. Expand **Advanced** to display the advanced connection pool options.
7. In the **Maximum Waiting for a Connection** field, enter the maximum number of connection requests that can wait for a connection from the connection pool while blocking threads.
8. In the **Connection Reserve Timeout** field, enter the number of seconds that connection requests can wait for a connection.

   For more information about these fields, see Configuration Options.
9. Click **Save**.

5.4.3 Configure Oracle parameters

Before you begin

Additional configuration may be required to support Oracle parameters.

To configure Oracle parameters:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC GridLink data source you want to configure.
5. Select **Configuration**, then select **Oracle**.
6. From the Oracle Parameters page, you can define Oracle parameters for this JDBC GridLink data source, such as:
   - Oracle Optimize UTF8 Conversion
   - Connection Initialization Callback
   - Oracle Proxy Session
   - Use Database Credentials
   - Replay Initiation Timeout
   - Active GridLink Data Source
   - Affinity Policy

   For more information about these fields, see Configuration Options.
7. Click **Save**.

5.4.4 Configure ONS client parameters

Before you begin

Additional configuration may be required to support ONS client parameters.

To configure ONS client parameters:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC GridLink data source you want to configure.
5. Select **Configuration**, then select **ONS Client**.
6. From the ONS Client Configuration page, you can define ONS configuration options, such as:
   - Fan Enabled
   - ONS Nodes
   - ONS Wallet File Directory
   - ONS Wallet Password
   - Confirm ONS Wallet Password

   For more information about these fields, see Configuration Options.

7. Click **Save**.

5.4.4.1 Configure SSL for the ONS client using a Oracle wallet file

Before you begin

A wallet file is only required when the ONS client is configured to communicate with ONS daemons using SSL. Additional configuration is required to support this feature.

To configure an Oracle wallet file when using SSL:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC GridLink data source you want to configure.
5. Select **Configuration**, then select **ONS Client**.
6. On the ONS Client Configuration page, configure the following attributes:
   - **ONS Wallet File**: enter the location of the Oracle wallet file in which the SSL certificates are stored.
   - **ONS Wallet Password**: enter and confirm the ONS wallet password.

   For more information about these fields, see Configuration Options.
7. Click **Save**.

5.4.5 Configure global transaction options

The transaction protocol for a JDBC data source determines how connections from the data source are handled during transaction processing.

> **Note:** If the data source uses an XA JDBC driver to create database connections, connections from the data source will support the two-phase commit transaction protocol only. No other transaction options are available for data sources that use an XA JDBC driver.

To configure transaction options for a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **Transaction**.
6. From the Transaction Options page, select the **Supports Global Transactions** checkbox to enable global transaction support in this data source. Clear this checkbox to disable (ignore) global transactions in this data source. In most cases, you should select this option.

   If you select Supports Global Transactions, select an option for transaction processing.

- **One-Phase Commit**: Select this option to enable the non-XA connection to participate in a global transaction as the only transaction participant.
- **Emulate Two-Phase Commit**: Enables a non-XA JDBC connection to emulate participation in distributed transactions using JTA. Select this option only if your application can tolerate heuristic conditions.
- **Logging Last Resource**: Select this option to enable a non-XA JDBC connection to participate in global transactions using the Logging Last Resource (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit.

For more information about these fields, see Configuration Options.

7. Click **Save**.

5.4.6 Configure JDBC data source diagnostic profiling

If the monitoring statistics indicate that there is a problem in your WebLogic Server domain, you can configure any data source to collect profile information to help you pinpoint the source of the problem. The collected profile information is stored in records in the WLDF Archive.

To configure diagnostic profiling for a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **Diagnostics**.
6. From the Diagnostic Profiling Options page, you can configure diagnostic profiling options, such as:
   - Profile Connection Usage
   - Profile Connection Reservation Wait
   - Profile Connection Leak
   - Profile Connection Reservation Failed
   - Profile Statement Cache Entry
   - Profile Statement Usage
   - Profile Connection Last Usage
   - Profile Connection Multithreaded Usage
   - Profile Connection Unwrap
   - Profile Harvest Frequency Seconds
   - Driver Interceptor

   For more information about these fields, see Configuration Options.
7. Click **Save.**

5.4.7 Configure JDBC data source identity options

You can choose the security option you want to use when mapping WebLogic Server user credentials to database user credentials. This section includes the following tasks:

- Enable credential mapping
- Enable identity-based connection pooling

5.4.7.1 Enable credential mapping

When an application requests a database connection from the data source, WebLogic Server determines the current WebLogic Server user ID and then sets the mapped database ID as a lightweight client ID on the database connection.

> **Note:** This feature relies on features in the JDBC driver and DBMS. It is only supported for use with Oracle and DB2 databases and with the Oracle Thin and DB2 UDB JDBC drivers, respectively.

To enable credential mapping for a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to configure.
5. Select **Configuration**, then select **Identity Options**.
6. Select the **Set Client ID On Connection** checkbox.

   > **Note:** **Set Client ID On Connection** and **Enable Identity Based Connection Pooling** are mutually exclusive. If you think you need both mechanisms to pass security credentials in your application environment, create separate data sources—one for use with **Set Client ID On Connection** and one for use with **Enable Identity Based Connection Pooling**.

   For more information about this field, see Configuration Options.
7. Click **Save**.

5.4.7.2 Enable identity-based connection pooling

Identity-based connection pooling allows applications to use a JDBC connection with a specific DBMS credential based on the end user application by pooling physical connections.

To enable identity-based connection pooling for a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to configure.

5. Select **Configuration**, then select **Identity Options**.
6. Select the **Enable Identity Based Connection Pooling** checkbox.

> **Note:** **Set Client ID On Connection** and **Enable Identity Based Connection Pooling** are mutually exclusive. If you think you need both mechanisms to pass security credentials in your application environment, create separate data sources—one for use with **Set Client ID On Connection** and one for use with **Enable Identity Based Connection Pooling**.

   For more information about this field, see Configuration Options.

7. Click **Save**.

5.4.8 Target JDBC data sources

Before you begin

Ensure that the JDBC drivers you want to use to create database connections are installed on all server instances on which you want to deploy the data source. Some JDBC drivers are installed with WebLogic Server, including WebLogic Type 4 JDBC drivers for DB2, Informix, MS SQL Server, and Sybase.

When you target a JDBC data source, a new instance of the data source is created on the target. When you select a server as a target, an instance of the data source is created on the server. When you select a cluster as a target, an instance of the data source is created on all member server instances in the cluster.

To target a JDBC data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source you want to target.
5. Select **Targets**.
6. On the Targets page, select the server instances or clusters on which you want to deploy the data source.
7. Click **Save** to save the JDBC data source configuration and deploy the data source to the targets that you selected.

For more information, see Configuration Options.

5.4.9 Create JDBC data sources notes

To create notes for JDBC data source configuration:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC data source for which you want to create notes.

5. Select **Notes**.
6. On the Notes page, enter your notes.
7. Click **Save**.

For more information, see Configuration Options.

5.4.10 Configure a JDBC multi data source

Multi data sources provide failover and load balancing for connection requests between two or more data sources. Before you create a multi data source, you should create the data sources that the multi data source will manage, and deploy them to same targets on which you want to deploy the multi data source. Note that the underlying databases must have some kind of data synchronization or replication. WebLogic Server does not handle that replication.

To configure a specific JDBC multi data source:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **JDBC Data Sources**.
4. In the JDBC Data Sources table, select the JDBC multi data source you want to configure.
5. To configure general settings, select **Configuration**, then select **General**.

   From the General Configuration page, you can define configuration options for this JDBC multi data source, such as:

   - Name
   - JNDI Name
   - Algorithm Name
   - Failover Request if Busy
   - Failover Callback Handler
   - Test Frequency Seconds

   For more information about these fields, see Configuration Options.
6. To select the JDBC data sources that you would like to include as part of this JDBC multi data source, select **Configuration**, then select **Data Sources**.
7. To deploy this JDBC multi data source to targeted server instances, select **Targets**.
8. To monitor this JDBC multi data source, see Monitor a JDBC multi data source.
9. To control this JDBC multi data source, see Control a JDBC multi data source.
10. To create notes that describe the configuration of this JDBC data source, select **Notes**.
11. Click **Save**.

6 WebLogic Server Templates

A server template contains common, non-default settings and attributes that you can apply to a set of server instances, which then inherit the template configuration.

Server templates enable you to easily manage configuration for a group of server instances in one centralized location. You define common configuration attributes in a server template and then apply the template to other server instances without having to manually configure each one. If you need to update an attribute across all server instances, you can simply change the value in the server template and the new value takes effect in all of the server instances that use the server template.

6.1 Monitor server templates

To monitor the status of all server templates configured in a domain:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain, such as:

   - Name
   - Cluster
   - Machine
   - Listen Port
   - Listen Address

   For more information about these fields, see Configuration Options.

6.2 Configure server templates

This section describes how to configure server templates. This section includes the following tasks:

- Configure server template general settings
- Configure server template cluster settings
- Configure server template services

- Configure server template keystores
- Change a server template keystore configuration
- Configure server template SSL settings
- Change a server template identity and trust location
- Configure server template Federation Services
- Configure server template deployment settings
- Configure server template tuning
- Configure server template overload settings
- Configure server template health monitoring
- Configure server template startup
- Configure server template Web services
- Configure server template Coherence cluster settings
- Configure server template protocols
- Create server template notes

6.2.1 Configure server template general settings

Changes to your general settings for a template will apply to every server that references the template unless explicitly overridden.

To configure general settings for a sever template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **General Settings**.
6. From the General Settings page, you can configure general features of this server template, such as:
   - Machine
   - Cluster
   - Listen Address
   - Listen Port Enabled
   - Listen Port
   - SSL Listen Port Enabled
   - SSL Listen Port
   - Client Cert Proxy Enabled

- Java Compiler
- Diagnostic Volume

For more information about these fields, see Configuration Options.

Optionally, expand **Advanced** to define advanced settings for this server template.

7. Click **Save**.

6.2.2 Configure server template cluster settings

A WebLogic Server cluster is a group of servers that work together to provide a scalable and reliable application platform.

To configure cluster configuration settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **Cluster**.
6. From the Cluster page, you can define a cluster configuration for your server template, such as:
   - Replication Group
   - Preferred Secondary Group
   - Cluster Weight
   - Interface Address
   - Replication Ports

   For more information about these fields, see Configuration Options.
7. Click **Save**.

6.2.3 Configure server template services

To configure service settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.

5. Select **Configuration**, then select **Services**.
6. From the Services page, you can set WebLogic service configuration settings, such as:

   - JMS
   - Default Store
   - Transaction Log
   - Messaging Bridge
   - XML Services

   For more information about these fields, see Configuration Options.
7. Click **Save**.

6.2.4 Configure server template keystores

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). You can view and define various keystore configurations which help you manage the security of message transmissions.

To configure keystore settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **Keystores**.
6. To change your keystore configuration:
   1. Click **Change**.
   2. From the Keystores dropdown menu, select the keystore configuration you want to use.
   3. Click **Save**.
7. From the Keystore page, you can define the keystore configuration settings for this server template, such as:

   - Identity settings
   - Trust settings

   For more information about these fields, see Configuration Options.
8. Click **Save**.

6.2.5 Change a server template keystore configuration

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). You can view and define various keystore configurations which help you manage the security of message transmissions.

To change the keystore configuration for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **Keystores**.
6. Click **Change**.
7. From the Keystores dropdown menu, select the keystore configuration you want to use.

   For more information about these fields, see Configuration Options.
8. Click **Save**.

6.2.6 Configure server template SSL settings

You can view and define various Secure Sockets Layer (SSL) settings for a server template, which help you manage the security of message transmissions.

To configure SSL settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **SSL**.
6. To change your identity and trust location:
   1. Click **Change**.
   2. From the Identity and Trust Locations dropdown menu, select the location you want to use.
   3. Click **Save**.
7. From the SSL page, you can define the SSL configuration settings for this server template, such as:

- Identity and Trust Locations
- Private Key Location
- Private Key Alias
- Private Key Passphrase
- Certificate Location
- Trusted Certificate Authorities

For more information about these fields, see Configuration Options.

Optionally, expand **Advanced** to define advanced configuration settings for this server template.

8. Click **Save**.

6.2.7 Change a server template identity and trust location

You can view and define various Secure Sockets Layer (SSL) settings for a server template, which help you manage the security of message transmissions.

To change your identity and trust location:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **SSL**.
6. Click **Change**.
7. From the Identity and Trust Locations dropdown menu, select the location you want to use.

   For more information about these fields, see Configuration Options.
8. Click **Save**.

6.2.8 Configure server template Federation Services

You can configure a WebLogic Server instance to function as a producer or as a consumer of SAML assertions that can be used for the following:

- Web single sign-on between online business partners
- Exchange of identity information in Web services security

The general process of configuring Federation Services depends upon the version of SAML you are using. WebLogic Server supports both SAML 1.1 and SAML 2.0.

To configure WebLogic Server to serve as a SAML 1.1 federated partner:

- Configure SAML 1.1 source services
- Configure SAML 1.1 destination services

To configure WebLogic Server to serve as a SAML 2.0 federated partner:

- Configure SAML 2.0 general services
- Configure SAML 2.0 Identity Provider services
- Configure SAML 2.0 Service Provider services

6.2.8.1 Configure SAML 1.1 source services

Before you begin

You must first configure a SAML Credential Mapper V2 security provider in the server's security realm.

You can configure a WebLogic Server instance to function as a SAML source site. A SAML source site is a site that provides an Intersite Transfer Service (ITS). A source site generates assertions that are conveyed to a destination site using one of the single sign-on profiles.

To configure a server as a SAML source site:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates.**
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**.
6. Select **Federation Services**, then select **SAML 1.1 Source Site**.
7. Select the **Source Site Enabled** attribute to cause this server to act as a source for SAML assertions.
8. From the SAML 1.1 Source Site page, you can also define other configuration settings for this server, such as:
   - Source Site URL
   - Signing Key Alias
   - Intersite Transfer URIs
   - ITS Requires SSL
   - Assertion Retrieval URIs
   - ARS Requires SSL
   - ARS Requires Two-Way SSL Authentication
   - Assertion Store Class Name
   - Assertion Store Properties

   For more information about these fields, see Configuration Options.
9. Click **Save**.

6.2.8.2 Configure SAML 1.1 destination services

Before you begin

You must first configure a SAML Identity Asserter V2 security provider in the server's security realm.

You can configure a WebLogic Server instance to function as a SAML destination site. A destination site can receive SAML assertions and use them to authenticate local subjects.

To configure a server as a SAML destination site:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**.
6. Select **Federation Services**, then select **SAML 1.1 Destination Site**.
7. Select the **Destination Site Enabled** attribute to enable the Assertion Consumer Service.
8. From the SAML 1.1 Destination Site page, you can also define other desired configuration settings for this server, such as:

   - Assertion Consumer URIs
   - ACS Requires SSL
   - SSL Client Identity Alias
   - POST Recipient Check Enabled
   - POST One-Use Check Enabled
   - Used Assertion Cache Class Name
   - Used Assertion Cache Properties

   For more information about these fields, see Configuration Options.
9. Click **Save**.

6.2.8.3 Configure SAML 2.0 general services

You can configure general SAML 2.0 services for a server. If you are configuring SAML 2.0 Web single sign-on services with your federated partners, the site information you configure is published in a metadata file that you send to your federated partners.

To configure the general SAML 2.0 properties of this server:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.
4. In the Server Templates table, select the name of the server you want to configure.
5. Select **Configuration**.
6. Select **Federation Services**, then select **SAML 2.0 General**.

7. Select the **Replicated Cache Enabled** attribute to use the persistent cache for storing SAML 2.0 artifacts.

   This option is required if you are configuring SAML 2.0 services in two or more WebLogic Server instances in your domain. For example, if you are configuring SAML 2.0 services in a cluster, you must enable this option in each Managed Server instance individually.

   > **Note:** If you are configuring SAML 2.0 services in two or more WebLogic Server instances in your domain, you must configure the RDBMS security store. The embedded LDAP server is not supported in these configurations.

8. In the **Site Info** section, enter the following information about your SAML 2.0 site:

   - Contact person details
   - Your organization's name and URL
   - The Published Site URL, which is the top-level URL for your site's SAML 2.0 service endpoints. This URL must be appended with the string `/saml2`, which will be automatically combined with constant suffixes to create full endpoint URLs.

9. In the **Bindings** section, enter the common binding information to be used by this SAML 2.0 server instance.

   If you do not specify a Transport Layer Security key alias and passphrase, the server's configured SSL private key alias and passphrase from the server's SSL configuration is used for the TLS alias by default.

10. If the Artifact binding is enabled for any SAML 2.0 security provider hosted on this server instance, define the Artifact Resolution Service settings in the **Artifact Resolution Service** section.

11. In the **Single Sign-on** section, enter the keystore alias and passphrase for the key to be used for signing documents sent to federated partners.

    If you do not specify a single sign-on signing key alias and passphrase, the server's configured SSL private key alias and passphrase from the server's SSL configuration is used by default.

12. Click **Save**.

For more information, see Configuration Options.

After you finish

After you have configured this server's general SAML 2.0 services, select the **SAML 2.0 Identity Provider** page or the **SAML 2.0 Service Provider** page to configure this server as an Identity Provider or Service Provider, respectively. For more information, see Configure SAML 2.0 Identity Provider services and Configure SAML 2.0 Service Provider services.

6.2.8.4 Configure SAML 2.0 Identity Provider services

You can configure a server in the role of SAML 2.0 Identity Provider. A SAML 2.0 Identity Provider creates, maintains, and manages identity information for principals, and provides principal authentication to other Service Provider partners within a federation by generating SAML 2.0 assertions for those partners.

To configure a server as a SAML 2.0 Identity Provider:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**.
6. Select **Federation Services**, then select **SAML 2.0 Identity Provider**.
7. Select the **Enabled** attribute to activate this server's SAML 2.0 services in the role of Identity Provider.
8. Select **Only Accept Signed Authentication Requests** if you want to ensure that any incoming authentication requests must be signed.
9. If you are using a custom login Web application to which unauthenticated requests are directed:
   - Select **Login Customized**.
   - Enter the URL of the custom login Web application.
   - Enter the login return query parameter

     The query parameter is a unique string that the SAML 2.0 services uses to hold the login return URL for the local single sign-on service servlet. (Note that, as an alternative, the login return URL can also be specified in the login Web application.)
10. Set the SAML bindings for which this server is enabled, and select the preferred binding type.
11. Click **Save**.

For more information, see Configuration Options.

After you finish

Coordinate with your federated partners to ensure that the SAML bindings you have enabled for this SAML authority, as well as your requirements for signed documents, are compatible with your partners.

6.2.8.5 Configure SAML 2.0 Service Provider services

You can configure a WebLogic Server instances as a SAML 2.0 Service Provider. A Service Provider is a SAML authority that can receive SAML assertions and extract identity information from those assertions. The identity information can then be mapped to local Subjects, and optionally groups as well, that can be authenticated.

To configure a server as a SAML 2.0 Service Provider:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**.
6. Select **Federation Services**, then select **SAML 2.0 Service Provider**.
7. Select the **Enabled** attribute to activate SAML 2.0 services in this server in the role of Identity Provider.
8. Set the configuration options for the local SAML 2.0 Service Provider services as appropriate. Note the following:
   - Choose options for **Always Sign Authentication Requests** and **Only Accept Signed Assertions** as desired and in a manner that is coordinated with your federated partners so that authentication requests and assertions are accepted.
   - Communicate the SAML bindings settings for this server with your federated partners to ensure compatibility.
9. Click **Save**.

For more information, see Configuration Options.

After you finish

Coordinate with your federated partners to ensure that the SAML bindings you have enabled for this SAML authority, as well as your requirements for signed documents, are compatible with your partners.

6.2.9 Configure server template deployment settings

To configure deployment settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates.**

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **Deployment**.
6. From the Deployment page, you can define the default deployment staging configuration for a server template, such as:
   - Staging Mode
   - Staging Directory Name
   - Upload Directory Name

   For more information about these fields, see Configuration Options.
7. Click **Save**.

6.2.10 Configure server template tuning

To configure tuning settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **Tuning**.
6. From the Tuning page, you can define tuning performance and functionality settings of this server template, such as:
   - Enable Native IO
   - JavaSocketMuxer Socket Readers
   - Enable Gathered Writes
   - Enable Scattered Reads
   - Maximum Open Sockets
   - Stuck Thread Max Time
   - Stuck Thread Timer Interval
   - Accept Backlog
   - Login Timeout
   - SSL Login Timeout
   - Reverse DNS Allowed

   For more information about these fields, see Configuration Options.

   Optionally, expand **Advanced** to define advanced settings for this server template.
7. Click **Save**.

6.2.11 Configure server template overload settings

To configure overload settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **Overload**.
6. From the Overload page, you can configure how WebLogic Server should react in the case of an overload or failure condition, including settings such as:

- Shared Capacity for Work Managers
- Failure Action
- Panic Action
- Free Memory Percent High Threshold
- Free Memory Percent Low Threshold
- Max Stuck Thread Time
- Stuck Thread Count

For more information about these fields, see Configuration Options.

7. Click **Save**.

6.2.12 Configure server template health monitoring

WebLogic Server provides a self-health monitoring capability to improve the reliability and availability of servers in a WebLogic Server domain. Selected subsystems within each server monitor their health status based on criteria specific to the subsystem.

To configure health monitoring settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **Health Monitoring**.
6. From the Health Monitoring page, you can configure the frequency of a server's automatic health checks and the frequency with which the Node Manager application (optional) checks the server's health state by defining settings such as:

   - Health Check Interval
   - Auto Kill if Failed
   - Auto Restart
   - Restart Interval
   - Max Restarts Within Interval
   - Restart Delay Seconds

   For more information about these fields, see Configuration Options.
7. Click **Save**.

6.2.13 Configure server template startup

Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions.

To configure startup settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **Server Start**.
6. From the Server Start page, you can configure the startup settings that Node Manager will use to start this server on a remote machine, including settings such as:
   - Java Home
   - Java Vendor
   - BEA Home
   - Root Directory
   - Class Path
   - Arguments
   - Security Policy File
   - User Name
   - Password
   - Confirm Password

   For more information about these fields, see Configuration Options.
7. Click **Save**.

6.2.14 Configure server template Web services

This section describes how to configure Web services for a server template.

This section includes the following tasks:

- Configure server template buffering settings
- Configure server template reliable messaging
- View server template logical stores

6.2.14.1 Configure server template buffering settings

To configure messaging buffering settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.

4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**.
6. Select **Web Services**, then select **Buffering**.
7. From the Buffering page, you can define message buffering settings for Web services on a server template, such as:
   - Retry Count
   - Retry Delay
   - Request Queue Enabled
   - Request Queue Connection Factory JNDI Name
   - Request Queue Transaction Enabled
   - Response Queue Enabled
   - Response Queue Connection Factory JNDI Name
   - Response Queue Transaction Enabled

   For more information about these fields, see Configuration Options.
8. Click **Save**.

6.2.14.2 Configure server template reliable messaging

Web service reliable messaging is a framework that enables an application running on one application server to reliably invoke a Web service running on another application server, assuming that both servers implement the WS-Reliable Messaging specification. Reliable is defined as the ability to guarantee message delivery between the two Web services.

To configure reliable messaging settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**.
6. Select **Web Services**, then select **Reliable Message**.
7. From the Reliable Messaging page, you can customize reliable messaging configuration settings on the Web service endpoint, such as:
   - Base Retransmission Interval
   - Enable Retransmission Exponential Backoff
   - Non-buffered Source

- Non-buffered Destination
- Acknowledgement Interval
- Inactivity Timeout
- Sequence Expiration

For more information about these fields, see Configuration Options.

8. Click **Save**.

6.2.14.3 View server template logical stores

A logical store is a named unit of storage that provides the business configuration requirements and connects the Web service to the physical and buffering queue.

To view logical stores configured for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to view.
5. Select **Configuration**.
6. Select **Web Services**, then select **Logical Stores**.
7. From the Logical Stores page, you can view information about each logical store that has been configured in the current WebLogic domain, such as:

   - Name
   - Persistence Strategy
   - Request Buffering Queue JNDI Name
   - Response Buffering Queue JNDI Name
   - Default

   For more information about these fields, see Configuration Options.

   Optionally, select **View** to access the following table options:

   - Columns: add or remove the columns displayed in the table
   - Detach: detach the table (viewing option)
   - Reorder: change the order of the columns displayed
   - Query by Example

6.2.15 Configure server template Coherence cluster settings

To configure Coherence settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.

2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Configuration**, then select **Coherence**.
6. From the Coherence page, you can select the Coherence cluster you want to use in this server template.

   For more information, see Configuration Options.
7. Click **Save**.

6.2.16 Configure server template protocols

This section describes how to configure server template protocols.

This section includes the following tasks:

- Configure server template protocol general settings
- Configure server template HTTP settings
- Configure server template jCOM settings
- Configure server template IIOP settings
- Monitor server template network channel settings
- Configure server template network channel settings

6.2.16.1 Configure server template protocol general settings

To configure general protocol settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Protocols**, then select **General Settings**.
6. From the General Settings page, you can define connections settings for various communication protocols that this server template uses, such as:
   - Complete Message Timeout
   - Idle Connection Timeout
   - Enable Tunneling

- Tunneling Client Ping
- Tunneling Client Timeout
- Maximum Message Size

For more information about these fields, see Configuration Options.

7. Click **Save**.

6.2.16.2 Configure server template HTTP settings

To configure HTTP protocol settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Protocols**, then select **HTTP**.
6. From the HTTP page, you can define the HTTP settings for this server template, such as:

   - Default WebApp Context Root
   - Post Timeout
   - Max Post Size
   - Enable Keepalives
   - Duration
   - HTTPS Duration
   - Frontend Host
   - Frontend HTTP Port
   - Frontend HTTPS Port
   - WAP Enabled
   - Remote Address Override
   - Send Server Header
   - Accept Context Path in Get Real Path
   - HTTP Max Message Size
   - Enable Tunneling
   - Tunneling Client Ping
   - Tunneling Client Timeout

   For more information about these fields, see Configuration Options.
7. Click **Save**.

6.2.16.3 Configure server template jCOM settings

To configure Java to COM (jCOM) protocol settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Protocols**, then select **jCOM**.
6. From the jCOM page, you can define jCOM protocol settings for this server template, such as:
   - Enable COM
   - NT Authentication Host
   - Enable Native Mode
   - Verbose Logging Enabled
   - Enable Memory Logging
   - Prefetch Enumeration
   - Apartment Threaded

   For more information about these fields, see Configuration Options.
7. Click **Save**.

6.2.16.4 Configure server template IIOP settings

To configure Internet Inter-ORB Protocol (IIOP) settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Protocols**, then select **IIOP**.
6. From the IIOP page, you enable IIOP for this server template.

   For more information, see Configuration Options.

   Optionally, expand **Advanced** to define advanced configuration settings for this server template.
7. Click **Save**.

6.2.16.5 Monitor server template network channel settings

To monitor network channel protocol settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.

   The Server Templates table displays information about each server template that has been configured in the current WebLogic Server domain.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Protocols**, then select **Channels**.
6. The Channels table displays information about each network channel that has been configured for this server template, such as:
   - Name
   - Protocol
   - Enabled
   - Listen Address
   - Listen Port
   - Public Address
   - Public Port

   For more information about these fields, see Configuration Options.
7. Click **Save**.

6.2.16.6 Configure server template network channel settings

To configure network channel protocol settings for a server template:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** dropdown menu, select **Environment**, then select **Server Templates**.
4. In the Server Templates table, select the name of the server template you want to configure.
5. Select **Protocols**, then select **Channels**.
6. In the Channels table, select the name of the channel you want to configure.
7. Select **Configuration**.
8. From the **General** page, you can define general configuration settings for the network channel, such as:
   - Name
   - Protocol
   - Listen Port

- Listen Address
- External Listen Address
- External Listen Port
- Enabled

For more information about these fields, see Configuration Options.

Optionally, expand **Advanced** to define advanced configuration settings for this network channel.

9. Click **Save**.
10. From the **Security** page, you can define security configuration options for the network channel, such as:
    - Two Way SSL Enabled
    - Client Certificate Enforced

    For more information about these fields, see Configuration Options.

    Optionally, expand **Advanced** to define advanced configuration settings for this network channel.
11. Click **Save**.

6.2.17 Create server template notes

To create notes for server configuration:

1. From the **Welcome Page**, select **Targets**, then select **Middleware**.
2. In the Targets table, select your WebLogic domain. If prompted, enter your user credentials to log in to the WebLogic domain.
3. From the **WebLogic Domain** menu, select **Environment**, then select **Server Templates**.
4. In the Server Templates table, click the name of the server template you want to configure.
5. Select **Notes**.
6. On the Notes page, enter your notes.
7. Click **Save**.

For more information, see Configuration Options.