

Oracle® Fusion Middleware

Administering Oracle ADF Applications

12c (12.1.3)

E41272-02

November 2015

Documentation for Oracle Application Development Framework (Oracle ADF) administrators that describes how to deploy, monitor, and configure ADF applications

Oracle Fusion Middleware Administering Oracle ADF Applications 12c (12.1.3)

E41272-02

Copyright © 2009, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Alyona Stashkova (Lead), Liza Rekadze, Himanshu Marathe

Contributing Author: Odile Sullivan-Tarazi, Cammy Moore

Contributors: Lynn Munsinger, Duncan Mills, Dipankar Bajpai, Harry Hsu, Ray Maslinski

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Contents

Preface	xi
Audience	xi
Documentation Accessibility	xi
Related Documents	xi
Conventions	xii
What's New in This Guide	xiii
New and Changed Features for Release 12c (12.1.3).....	xiii
Other Significant Changes in this Document for Release 12c (12.1.3).....	xiii
Part I Understanding Oracle ADF	
1 Introduction to Oracle ADF Administration	
1.1 Introducing Oracle ADF	1-1
1.2 Oracle ADF Architecture	1-1
1.2.1 ADF Business Components.....	1-2
1.2.2 ADF Model	1-2
1.2.3 ADF Controller.....	1-2
1.2.4 ADF Faces Rich Client.....	1-2
1.2.5 ADF Desktop Integration	1-3
1.3 Administering Oracle ADF Applications.....	1-3
Part II Basic Administration	
2 Getting Started with Managing Oracle ADF	
2.1 Overview of ADF Administration Tools.....	2-1
2.2 Getting Started Using Fusion Middleware Control to Manage ADF	2-1
2.3 Using the WebLogic Scripting Tool	2-4
2.4 Using the Oracle WebLogic Administration Console	2-4
2.5 Using the Fusion Middleware Control MBean Browsers	2-5
3 Monitoring and Configuring ADF Applications Using Fusion Middleware Control	
3.1 Introduction to ADF Application Monitoring and Configuration	3-1

3.2	Monitoring Performance Using Enterprise Manager Fusion Middleware Control.....	3-2
3.2.1	Understanding the Home Page	3-2
3.2.2	Finding Version Information of ADF Runtime JARs	3-3
3.2.3	Viewing Application Performance.....	3-3
3.2.4	Viewing Application Module Pool Performance.....	3-3
3.2.5	Viewing ADF Task Flow Performance	3-4
3.3	Configuring Application Properties Using Fusion Middleware Control.....	3-4
3.3.1	Modifying ADF Business Components Parameters	3-4
3.3.2	Modifying Connection Configurations	3-5
3.4	Configuring Application Properties Using the MBean Browser	3-6
3.4.1	Modifying ADF Application Configurations Using MBeans.....	3-6
3.4.2	Modifying ADF Connections Using MBeans	3-7
3.4.3	Modifying ADF Business Components Configurations Using MBeans	3-8
3.4.4	Modifying MDS Configuration Using MBeans.....	3-9
3.4.5	Modifying Active Data Service Configuration Using MBeans	3-10
3.5	Editing Credentials Deployed with the Application	3-11

4 Deploying ADF Applications

4.1	Introduction to Deploying ADF Applications.....	4-1
4.2	Preparing the Standalone Application Server for Deployment	4-2
4.2.1	Installing the ADF Runtime to the Application Server Installation.....	4-4
4.2.1.1	Installing the ADF Runtime into an Existing WebLogic Server Installation Using the Oracle Fusion Middleware Infrastructure Installer 4-4	
4.2.2	Installing Relevant Patches.....	4-5
4.2.3	Creating and Extending Oracle WebLogic Server Domains	4-5
4.2.3.1	Creating an Oracle WebLogic Server Domain for Oracle ADF	4-5
4.2.3.2	Extending the Oracle WebLogic Server Domain for Oracle ADF	4-6
4.2.3.3	Setting Up Remote WebLogic Managed Servers for Oracle ADF.....	4-6
4.2.4	Creating a JDBC Data Source for Oracle WebLogic Server	4-8
4.3	Deploying Using Oracle Enterprise Manager Fusion Middleware Control	4-9
4.4	Deploying Using Scripting Commands.....	4-9
4.5	Deploying Using Scripts and Ant.....	4-10
4.6	Deploying Using the Application Server Administration Tool	4-10

5 Administering ADF Desktop Integration

5.1	Installing and Upgrading the Runtime Edition of ADF Desktop Integration From a Shared File Server and Web Server 5-1	
5.1.1	Making the Runtime Edition of ADF Desktop Integration Available to Multiple End Users 5-2	
5.1.2	Upgrading the Runtime Edition of ADF Desktop Integration	5-2
5.1.3	Installing the Runtime Edition of ADF Desktop Integration From a Web Server	5-3
5.1.4	Upgrading the Runtime Edition of ADF Desktop Integration From a Web Server ..	5-4
5.1.5	Patching the Runtime Edition of ADF Desktop Integration	5-5
5.1.6	What Happens When Runtime Edition of ADF Desktop Integration Is Installed and Upgraded 5-5	
5.2	ADF Desktop Integration Logs	5-5
5.2.1	Server-Side Logging	5-6

5.2.2	Using the Oracle Diagnostics Log Analyzer to Analyze ADF Desktop Integration Servlet Requests	5-6
5.3	Security in ADF Desktop Integration.....	5-6
5.3.1	End User Authentication	5-7
5.3.2	What You May Need to Know About Configuring Security in a Fusion Web Application	5-7
5.3.3	What You May Need to Know About Resource Grants for Web Pages.....	5-8
5.4	Verifying the Client Version of ADF Desktop Integration	5-8
5.4.1	Configuring Client-Server Version Precision	5-9
5.4.2	What You May Need to Know About Client-Server Version Verification	5-10
5.5	Verifying Integrated Excel Workbook Metadata	5-10
5.5.1	Disabling the Metadata Tamper-Check in the Fusion Web Application	5-11
5.5.2	Allowing Missing Entries in the ADF Desktop Integration Client Registry.....	5-12
5.6	Common ADF Desktop Integration Error Messages and Problems	5-13
5.7	Localizing the ADF Desktop Integration Installer	5-14

6 Diagnostic Tools

6.1	Introduction to Monitoring and Diagnostic Tools	6-1
6.2	Click History	6-1
6.2.1	How to Obtain Click History	6-2
6.2.2	How to Enable Click History for an Application.....	6-2
6.2.3	How to Enable and Disable Click History in WebLogic Server	6-2
6.2.4	How to Adjust Click History Capacity.....	6-2
6.2.5	How to Dump Click History Data Explicitly.....	6-3
6.2.6	How to Interpret Click History Records	6-4
6.2.7	How to Get Click History DMS Metrics.....	6-6
6.3	ADF Logs	6-6
6.4	Diagnosing Problems using the Diagnostic Framework	6-6
6.5	Finding ADF Runtime Version Information.....	6-7
6.5.1	How to Find Version Information by Starting the Application.....	6-7
6.5.2	How to Find Version Information by Using Shell Scripts in Offline Mode.....	6-8
6.5.3	How to Find Version Information by Using the WLST Scripting Console.....	6-8
6.5.4	How to Find Version Information Using Fusion Middleware Control.....	6-8
6.5.5	How to Find Version Information by Inspecting Diagnostic Dump Files	6-8
6.6	Troubleshooting Oracle ADF for High Availability	6-9

7 WLST Command Reference for ADF Applications

7.1	Overview of Custom WSLT Commands for Oracle ADF.....	7-1
7.2	ADF-Specific WLST Commands.....	7-1
7.2.1	adf_createFileURLConnection	7-2
7.2.1.1	Description	7-2
7.2.1.2	Syntax	7-2
7.2.1.3	Example.....	7-2
7.2.2	adf_createHttpURLConnection	7-2
7.2.2.1	Description	7-2
7.2.2.2	Syntax	7-3

7.2.2.3	Example.....	7-3
7.2.3	adf_setURLConnectionAttributes	7-3
7.2.3.1	Description	7-3
7.2.3.2	Syntax	7-3
7.2.3.3	Example.....	7-3
7.2.4	adf_listURLConnection	7-3
7.2.4.1	Description	7-3
7.2.4.2	Syntax	7-4
7.2.4.3	Example.....	7-4
7.2.5	getADFMArchiveConfig	7-4
7.2.5.1	Description	7-4
7.2.5.2	Syntax	7-4
7.2.5.3	Example.....	7-5
7.2.6	exportJarVersions	7-6
7.2.6.1	Description	7-6
7.2.6.2	Syntax	7-6
7.2.6.3	Example.....	7-6
7.2.7	exportApplicationJarVersions	7-6
7.2.7.1	Description	7-6
7.2.7.2	Syntax	7-6
7.2.7.3	Example.....	7-7
7.2.8	exportApplicationSelectedJarVersions	7-7
7.2.8.1	Description	7-7
7.2.8.2	Syntax	7-7
7.2.8.3	Example.....	7-7
7.2.9	createWebServiceConnection.....	7-8
7.2.9.1	Description	7-8
7.2.9.2	Syntax	7-8
7.2.9.3	Example.....	7-8
7.2.10	listWebServiceConnection.....	7-8
7.2.10.1	Description	7-8
7.2.10.2	Syntax	7-9
7.2.10.3	Example.....	7-9
7.2.11	deleteWebServiceConnection	7-9
7.2.11.1	Description	7-9
7.2.11.2	Syntax	7-9
7.2.11.3	Example.....	7-9

Part III Appendices

A Configuring GlassFish Server

A.1	About Configuring GlassFish	A-1
A.2	Obtaining GlassFish Server and Oracle ADF Runtime	A-1
A.2.1	Obtaining GlassFish Server	A-2
A.2.2	Obtaining Oracle ADF Runtime	A-2
A.3	Configuring GlassFish with ADF Runtime Libraries	A-2
A.3.1	Installing ADF Share Libraries Manually	A-3

A.4	Additional Configuration Tasks	A-4
A.4.1	Using the GlassFish Administration Console.....	A-4
A.4.2	Creating a Datasource for GlassFish.....	A-4
A.4.3	Configuring the JVM Cache	A-4
A.5	Deploying an ADF Application to GlassFish	A-5

B ADF Runtime Libraries

B.1	Using JDeveloper to Find the ADF Runtime Library	B-1
B.2	adf.oracle.domain.webapp.war Library	B-1
B.3	adf.oracle.domain.ear Library	B-4
B.4	adf.desktopintegration.war Library	B-6
B.5	adf.desktopintegration.model.ear Library	B-7
B.6	System Classpath	B-7

C Audit Reference for Oracle Application Development Framework

C.1	About Custom and Standard Audit Reports	C-1
C.2	Attributes of ADF Audit Events	C-1
C.3	Audit Events in Oracle ADF View	C-2
C.4	Audit Events in Oracle ADF Business Components.....	C-3
C.5	Audit Events in Oracle ADF Model	C-3
C.6	Audit Events in Oracle ADF Controller	C-4
C.7	Audit Events in Jedi.....	C-4

Preface

Welcome to *Administering Oracle ADF Applications*.

Audience

This document is intended for system administrators who need to deploy, manage, monitor, and configure Oracle ADF applications using the Oracle Application Development Framework (Oracle ADF).

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.

Related Documents

For more information, see the following documents:

Developing Fusion Web Applications with Oracle Application Development Framework

Administering Oracle Fusion Middleware

Developing Web User Interfaces with Oracle ADF Faces

Developing Applications with Oracle ADF Desktop Integration

Securing Applications with Oracle Platform Security Services

WLST Command Reference for WebLogic Server

High Availability Guide

Planning an Installation of Oracle Fusion Middleware

Installing and Configuring the Oracle Fusion Middleware Infrastructure

Installing Oracle JDeveloper

[Creating WebLogic Domains Using the Configuration Wizard](#)
[Creating Templates and Domains Using the Pack and Unpack Commands](#)
[Patching with OPatch](#)
[Creating Schemas with the Repository Creation Utility](#)
[Deploying Applications to Oracle WebLogic Server](#)
[Oracle JDeveloper Online Help](#)
[Oracle JDeveloper Release Notes](#), included with your JDeveloper installation, and on Oracle Technology Network
[Oracle Fusion Middleware Java API Reference for Oracle ADF Model](#)
[Oracle Fusion Middleware Java API Reference for Oracle ADF Controller](#)
[Oracle Fusion Middleware Java API Reference for Oracle ADF Lifecycle](#)
[Oracle Fusion Middleware Java API Reference for Oracle ADF Faces](#)
[Oracle Fusion Middleware JavaScript API Reference for Oracle ADF Faces](#)
[Oracle Fusion Middleware Java API Reference for Oracle ADF Data Visualization Components](#)
[Oracle Fusion Middleware Java API Reference for Oracle ADF Share](#)
[Oracle Fusion Middleware Java API Reference for Oracle ADF Business Components Browser](#)
[Oracle Fusion Middleware Java API Reference for Oracle Generic Domains](#)
[Oracle Fusion Middleware interMedia Domains Java API Reference for Oracle ADF Business Components](#)
[Oracle Fusion Middleware Java API Reference for Oracle Metadata Service \(MDS\)](#)

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.
<i>italic</i>	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.

What's New in This Guide

The following topics introduce the new and changed features of Oracle JDeveloper and Oracle Application Development Framework (Oracle ADF) and other significant changes, which are described in this guide. This document is the new edition of the document formerly titled *Oracle Fusion Middleware Administrator's Guide for Oracle Application Development Framework*.

New and Changed Features for Release 12c (12.1.3)

Oracle Fusion Middleware Release 12c (12.1.3) of Oracle JDeveloper and Oracle Application Development Framework (Oracle ADF) includes the following new and changed development features, which are described in this guide.

- Added a new chapter to describe how to use diagnostic tools such as Click History, logs, and diagnostic framework. See [Chapter 6, "Diagnostic Tools."](#)

Other Significant Changes in this Document for Release 12c (12.1.3)

For Release 12c (12.1.3), this document has been updated in several ways. Following are the sections that have been added or changed.

- Added new WLST commands for creating, listing, and editing Web Service connections for ADF Applications. See [Chapter 7, "WLST Command Reference for ADF Applications."](#)

Part I

Understanding Oracle ADF

Part I contains the following chapters:

- [Chapter 1, "Introduction to Oracle ADF Administration"](#)

Introduction to Oracle ADF Administration

This chapter describes the administrative tasks you can perform and the tools you can use to deploy, manage, monitor, and configure applications developed for the Oracle Application Development Framework (Oracle ADF). Some of the tools you will be using are the Oracle Enterprise Manager Fusion Middleware Control, Oracle JDeveloper, and WebLogic Scripting Tool (WLST).

This chapter includes the following sections:

- [Section 1.1, "Introducing Oracle ADF"](#)
- [Section 1.2, "Oracle ADF Architecture"](#)
- [Section 1.3, "Administering Oracle ADF Applications"](#)

1.1 Introducing Oracle ADF

The Oracle Application Development Framework (Oracle ADF) builds on Java Platform, Enterprise Edition (Java EE) standards and open-source technologies to provide a complete framework for implementing service-oriented applications. You can use this framework to provide enterprise solutions across different platforms. You can build applications that search, display, create, modify, and validate data for web, web services, desktop, or mobile interfaces.

You use Oracle JDeveloper with Oracle ADF to develop applications with an environment that supports the full development lifecycle of design, test, and deployment. For more information about ADF development, see *Developing Fusion Web Applications with Oracle Application Development Framework*.

After you have developed and tested your ADF application in test environments, you can deploy your application to production environments using the tools described in this book. You can monitor the performance of applications as they are running. You can also manage and configure properties and attributes.

1.2 Oracle ADF Architecture

Oracle ADF supports the industry-standard model-view-controller architecture to achieve separation of business logic, navigation, and user interface. The MVC architecture provides:

- A model layer that represents the data values
- A view layer that contains the UI components
- A controller layer that handles input and navigation
- A business service layer that encapsulates business logic

The Fusion web application technology stack components are:

- ADF Model, for accessing declarative data binding metadata
- ADF Business Components, for building business services
- ADF Faces rich client, for AJAX-enabled UI components for web applications built with JavaServer Faces (JSF)
- ADF Controller, for input processing, navigation, and reusable task flows

1.2.1 ADF Business Components

ADF Business Components are application objects you can use to implement service-oriented Java EE applications. You implement ADF Business Components for clients to query, insert, update, and delete business data. You can apply business rules to the Business Components to enforce proper usage. The key components of ADF Business Components are the entity object, the view object, and the application module.

An *entity object* represents a row in a database table. It uses data manipulation language (DML) operations to modify data. Entity objects are used with others to reflect relationships in the database schema.

A *view object* represents a SQL query. You use the SQL Language to query the database to obtain the results. You can also link a view object with other entity objects to create master-detail hierarchies.

An *application module* is the transactional component that allows UI components to access data. It presents a data model and methods to perform certain tasks.

1.2.2 ADF Model

ADF Model implements a service abstraction called *data control*. Data control uses metadata interfaces to abstract business services. This metadata is used to describe data collections, properties, methods, and types. In JDeveloper, data controls appear in the Data Controls panel. When you drag and drop attributes, collections, and methods onto a page, JDeveloper automatically creates the bindings from the page to the associated services.

1.2.3 ADF Controller

ADF Controller provides a navigation and state management model that works with JSF. You can create navigational flows called task flows that encapsulate a specific task sequence.

1.2.4 ADF Faces Rich Client

ADF Faces provides over 100 rich components that can be used out of the box to create web applications. ADF Faces components provide built-in AJAX functionality to allow requests to be sent to the server without fully rendering the page. JSF provides server-side control to reduce the dependency on JavaScript. The components support skinning, internationalization, and accessibility options.

ADF Faces has a large set of components, including tables, trees, dialogs, accordions, and a variety of layout components. It also includes ADF Data Visualization components, which are Flash- and SVG-enabled, for displaying graphs, charts, and gauges.

1.2.5 ADF Desktop Integration

ADF Desktop Integration provides a framework for Oracle ADF developers to extend the functionality provided by a Fusion web application to desktop applications, such as Excel, even when they are disconnected from the network.

Using Excel's familiar user interface, end users can undertake information management tasks, such as performing complex calculations or uploading a large amount of data.

1.3 Administering Oracle ADF Applications

You can perform a variety of administration tasks on ADF applications. You can deploy ADF applications using Enterprise Manager Fusion Middleware Control, WLST commands, the `ojdeploy` command, scripts, or the WebLogic Administration Console.

After the ADF application has been deployed, you can configure application properties using Enterprise Manager Fusion Middleware Control. You can also configure some properties using the MBean Browser to change values in the ADF MBeans. For example, you can use Enterprise Manager Fusion Middleware Control to change the URL connection or WebService connection endpoints or seed the production credentials.

When you run the application, you can monitor performance data on the application modules, application module pooling, and task flows.

Part II

Basic Administration

Part II contains the following chapters:

- [Chapter 2, "Getting Started with Managing Oracle ADF"](#)
- [Chapter 3, "Monitoring and Configuring ADF Applications Using Fusion Middleware Control"](#)
- [Chapter 4, "Deploying ADF Applications"](#)
- [Chapter 5, "Administering ADF Desktop Integration"](#)
- [Chapter 6, "Diagnostic Tools"](#)
- [Chapter 7, "WLST Command Reference for ADF Applications"](#)

Getting Started with Managing Oracle ADF

This chapter describes how to use Oracle Enterprise Fusion Middleware Control to perform ADF application configuration, monitor performance, and setting up logs. It also describes how to use the ADF-specific WebLogic Scripting Tool (WLST).

This chapter contains the following sections:

- [Section 2.1, "Overview of ADF Administration Tools"](#)
- [Section 2.2, "Getting Started Using Fusion Middleware Control to Manage ADF"](#)
- [Section 2.3, "Using the WebLogic Scripting Tool"](#)
- [Section 2.4, "Using the Oracle WebLogic Administration Console"](#)

2.1 Overview of ADF Administration Tools

You can use Enterprise Manager Fusion Middleware Control, WLST commands, or Oracle WebLogic Administration Console to manage and configure ADF applications.

Oracle offers the following primary tools for managing your Oracle Fusion Middleware installations:

- Oracle Enterprise Manager Fusion Middleware Control. See [Section 2.2](#).
- The Oracle Fusion Middleware command-line tools. See [Section 2.3](#).
- Oracle WebLogic Server Administration Console. See [Section 2.4](#).
- The Fusion Middleware Control MBean Browser. See [Section 2.5](#).

2.2 Getting Started Using Fusion Middleware Control to Manage ADF

For information about navigating within Fusion Middleware Control and a description of its elements and main menus, see "Navigating Within Fusion Middleware Control" in the *Administering Oracle Fusion Middleware*.

Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer Oracle Fusion Middleware.

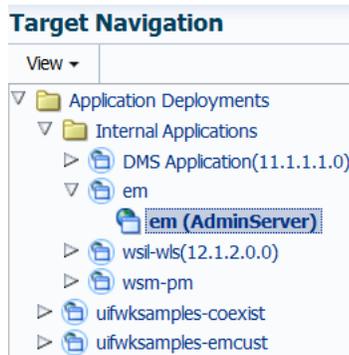
To navigate to Oracle Application Development Framework pages in Fusion Middleware Control:

1. Enter the Fusion Middleware Control URL, which includes the name of the host and the port number assigned to Fusion Middleware Control during the installation. The following shows the format of the URL

```
http://hostname.domain:port/em
```

The port number is the number of the Administration Server of Oracle WebLogic Server. By default, the port number is 7001.

2. Enter the Oracle Fusion Middleware administrator user name and password and click **Login**.
3. From the navigation pane, expand the farm and then Application Deployments, and select the ADF application.

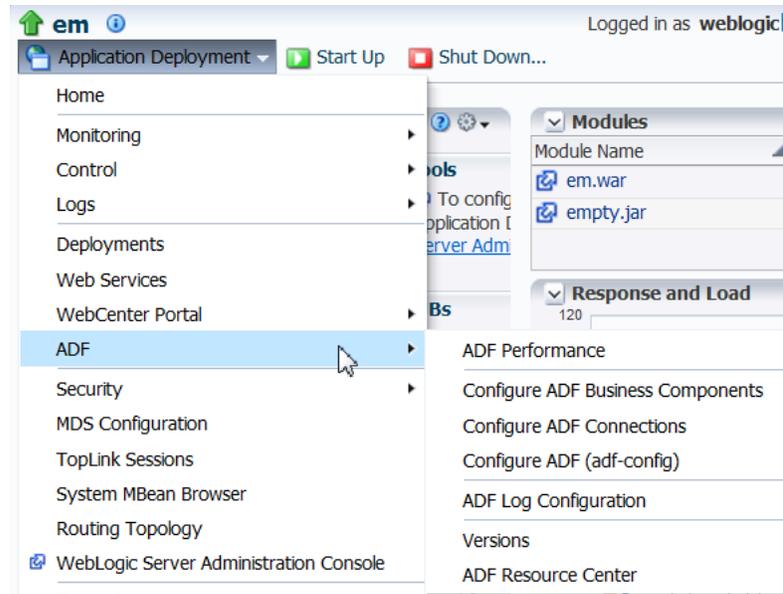


This image shows the Target Navigation pane.

- Application Deployments
 - Internal Applications
 - em
 - em (AdminServer)
 - wsil-wls (12.1.2.0.0)
 - wsm-pm
 - uifwksamples-coexist
 - uifwksamples-emcust

The ADF home page displays.

4. Select the **Application Deployment** menu and then the **ADF** menu.



This image shows the ADF menu. The options are described in the table that follows.

The ADF menu displays the following options.

Menu Option	Description
ADF Performance	Displays the Application Module Pool performance tab and the ADF Taskflows tab. For more information, see Section 3.2.4, "Viewing Application Module Pool Performance," and Section 3.2.5, "Viewing ADF Task Flow Performance."
Configure ADF Business Components	Configure the ADF Business Components. You can select configuration tabs for: <ul style="list-style-type: none"> ■ Pooling and Scalability - Application Pool Properties ■ Pooling and Scalability - Connection Pool Properties ■ Core ■ Database Properties ■ Security Properties For more information, see Section 3.3.1, "Modifying ADF Business Components Parameters."
Configure ADF Connections	Allows you create and edit connections that are available to ADF applications. For more information, see Section 3.3.2, "Modifying Connection Configurations."
Configure ADF (adf-config)	Allows you to configure ADF application properties using the System MBean Browser. For more information, see Section 3.4.1, "Modifying ADF Application Configurations Using MBeans."
ADF Log Configuration	Allows you to configure the logging levels for ADF Loggers. For more information, see "Managing Log Files and Diagnostic Data" chapter in <i>Administering Oracle Fusion Middleware</i> .

Menu Option	Description
Versions	Allows you to find the version information of ADF runtime JAR files. For more information, see Section 3.2.2, "Finding Version Information of ADF Runtime JARs."
ADF Resource Center	Provides links to ADF resources such as documentation, knowledge base, forums, demos, samples, and software downloads.

2.3 Using the WebLogic Scripting Tool

You can also use Oracle WebLogic Scripting Tool (WLST) to create, manage, and monitor Oracle WebLogic Server domains. WLST is a command-line based on Jython. ADF provides a set of custom WLST commands that can use to perform functions specifically for ADF applications.

For more information about using WLST, see "Using the WebLogic Scripting Tool" in *Understanding the WebLogic Scripting Tool* and "Getting Started Using the Oracle WebLogic Scripting Tool (WLST)" in *Administering Oracle Fusion Middleware*.

For reference information about the ADF WLST commands, see [Chapter 7, "WLST Command Reference for ADF Applications."](#)

To access the ADF-specific WLST commands:

- Go to the Oracle Common home directory for your installation, for example `/home/Oracle/Middleware/oracle_common`.

For information about the Oracle Common home directory and installing Oracle Fusion Middleware, see *Planning an Installation of Oracle Fusion Middleware*.
- Start Oracle WebLogic Server.
- Start WLST using the `wlst.sh/cmd` command located in the `oracle_common/common/bin` directory. For example:
 - `/home/Oracle/Middleware/oracle_common/common/bin/wlst.sh` (UNIX)
 - `C:\Oracle\Middleware\oracle_common\common\bin\wlst.cmd` (Windows)
- Connect to the running WebLogic Server instance using the `connect()` command. For example, the following command connects WLST to the Admin Server at the URL `myAdminServer.example.com:7001` using the `username/password` credentials `my_username/my_password`:


```
connect("my_username", "my_password", "t3://myAdminServer.example.com:7001")
```

For reference information about the ADF WLST commands, see [Chapter 7, "WLST Command Reference for ADF Applications."](#)

2.4 Using the Oracle WebLogic Administration Console

You can use the Oracle WebLogic Administration Console to manage Oracle WebLogic Server domains. The console runs in a Web browser and has a graphical user interface.

For more information, see "Getting Started Using Oracle WebLogic Server Administration Console" in *Administering Oracle Fusion Middleware*.

2.5 Using the Fusion Middleware Control MBean Browsers

Fusion Middleware Control provides a set of MBean browsers that allow to you view the MBeans for an application or for an Oracle WebLogic Server. You can also use the MBean browser to perform monitoring and configuration tasks.

For more information, see "Getting Started Using the Fusion Middleware Control MBean Browsers" in *Administering Oracle Fusion Middleware*.

Monitoring and Configuring ADF Applications Using Fusion Middleware Control

This chapter describes how to view ADF application performance. It also describes how to configure an ADF application's properties after it has been deployed to Oracle WebLogic Server. It describes how to use Oracle Enterprise Manager Fusion Middleware Control and its System MBean Browser to perform monitoring and configuration tasks. It also describes diagnosing problems with the Diagnostic Framework and monitoring metrics using DMS spy.

This chapter includes the following sections:

- [Section 3.1, "Introduction to ADF Application Monitoring and Configuration"](#)
- [Section 3.2, "Monitoring Performance Using Enterprise Manager Fusion Middleware Control"](#)
- [Section 3.3, "Configuring Application Properties Using Fusion Middleware Control"](#)
- [Section 3.4, "Configuring Application Properties Using the MBean Browser"](#)
- [Section 3.5, "Editing Credentials Deployed with the Application"](#)

3.1 Introduction to ADF Application Monitoring and Configuration

After you have deployed an ADF application to Oracle WebLogic Server, you can view the application performance and configure application properties on the server. You can use Enterprise Manager Fusion Middleware Control to perform these tasks.

Enterprise Manager Fusion Middleware Control offers a user interface for the performance tasks. Some configuration tasks can be performed either from a user interface or by configuring an MBean, as listed in [Table 3–1](#).

Table 3–1 Configuration Tasks Using Fusion Middleware Control

Configuration tasks	Fusion Middleware Control UI	Fusion Middleware Control MBean Browser
ADF Business Components	Section 3.3.1, "Modifying ADF Business Components Parameters"	Section 3.4.3, "Modifying ADF Business Components Configuration Using MBeans"
ADF connections	Section 3.3.2, "Modifying Connection Configurations"	Section 3.4.2, "Modifying ADF Connections Using MBeans"

Table 3–1 (Cont.) Configuration Tasks Using Fusion Middleware Control

Configuration tasks	Fusion Middleware Control UI	Fusion Middleware Control MBean Browser
ADF application configuration		Section 3.4.1, "Modifying ADF Application Configurations Using MBeans"
Metadata Services (MDS)		Section 3.4.4, "Modifying MDS Configuration Using MBeans"
Active Data Service (ADS)		Section 3.4.5, "Modifying Active Data Service Configuration Using MBeans"

By default, the post-deployment changes made using MBeans are stored in MDS with a layer name of `adfshare` and a layer value of `adfshare`. You can provide a specific layer name by specifying the `adfAppUID` property in the application's `adf-config.xml`.

[Example 3–1](#) shows the `adf-properties-child` code in `adf-config.xml`.

Example 3–1 MDS Layers in the `adf-config.xml` File

```
<adf:adf-properties-child xmlns="http://xmlns.oracle.com/adf/config/properties">
  <adf-property name="adfAppUID" value="DeptApp.myApp" />
</adf:adf-properties-child>
```

If you are moving data between MDS repositories (for example, from a test to a production system), use the `MDS exportMetadata` and `importMetadata` commands as described in the chapter on managing the Oracle metadata repository in the *Administering Oracle Fusion Middleware* and in the chapter on Metadata Services custom WLST commands in the *WLST Command Reference for WebLogic Server*.

3.2 Monitoring Performance Using Enterprise Manager Fusion Middleware Control

You can monitor the performance of Oracle ADF applications using the Fusion Middleware Control, as described in the following topics:

- [Section 3.2.1, "Understanding the Home Page"](#)
- [Section 3.2.2, "Finding Version Information of ADF Runtime JARs"](#)
- [Section 3.2.3, "Viewing Application Performance"](#)
- [Section 3.2.4, "Viewing Application Module Pool Performance"](#)
- [Section 3.2.5, "Viewing ADF Task Flow Performance"](#)

3.2.1 Understanding the Home Page

You can view performance information about application module pools and ADF task flows. Application module components can be used to support a unit of work which spans multiple browser pages.

You can:

1. View application module pool performance.
2. View task flow performance.

3.2.2 Finding Version Information of ADF Runtime JARs

You can use find the version information for ADF runtime JAR files and display the results in a table or export the information to an Excel file. You can narrow down your results by specifying a filter for each column.

To finding version information of runtime JARs:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to view the runtime JARs.
2. From the Application Deployment menu, choose **ADF > Versions**.

The Log Configuration page displays.

3. You can filter the results for each column by entering the filter criteria into the input field above each column. For instance, if you want to display only JAR files associated with specification 2.2, enter "2.2" in the field above the **Specification Version** column and press Enter.
4. If you want to export the results table into an Excel file, click **Export to Excel**.

3.2.3 Viewing Application Performance

You can view performance information about application modules. Application module components can be used to support a unit of work which spans multiple browser pages.

To view application performance:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to view.
2. From the Application Deployment menu, choose **ADF > ADF Performance**.

The ADF Performance page displays. It contains subtabs for viewing performance information about active application module pools and task flows.

3.2.4 Viewing Application Module Pool Performance

An *application module pool* is a collection of instances of a single application module type which are shared by multiple application clients. One application module pool is created for each root application module used by an ADF web application (ADF Business Components, ADF Controller, or ADF Faces) in each Java virtual machine where a root application module of that type is used by the ADF Controller layer.

To view application module pool performance:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to view pool performance.
2. From the Application Deployment menu, choose **ADF > ADF Performance**.
3. Click the **Application Module Pools** tab.
4. In the **Module** column, select an application module to display its details in the Application Module Pools table.

No Data Available displays in the Module column if an application has never run.

5. Click a module to display additional informations about the module, for example, Lifetime, State Management, Pool Use, and Application Module Pools Page.

Use the Application Module Pools page to display active application module pools, a collection of application module instances of the same type. The Application Module Pools page:

- Displays size and performance information about pool connections
- Specifies settings that affect how application module pools behave
- Specifies credential information for the application module pools

3.2.5 Viewing ADF Task Flow Performance

You can view performance information about task flows. Task flows provide a modular and transactional approach to navigation and application control. Task flows mostly contain pages that will be viewed, but they also can contain activities that call methods on managed beans, evaluate an EL expression, or call another task flow, all without invoking a particular page.

To view task flow performance:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to view task flow performance.
2. From the Application Deployment menu, choose **ADF > ADF Performance**.
3. Click the **Task Flows** tab.

By default, Task Flow Performance charts on the tab display data for the preceding 15 minutes. To set a different interval, click the time at the top of the page or move the slider to another interval, for example, from 08:00 AM to 08:30 AM.

4. Click **TF Charts**.
 - **Request Processing Time** displays the average request processing time for all ADF task flows that execute during the selected interval.
 - **Active Task Flows** displays the number of active instances of each ADF task flow during the selected interval.

3.3 Configuring Application Properties Using Fusion Middleware Control

You can use Enterprise Manager Fusion Middleware Control to configure ADF application configuration parameters. These configuration parameters are stored in ADF MBeans. Fusion Middleware Control provides a user interface to configure the ADF Business Components and ADF Connections MBeans. You can also use the System MBean Browser to directly access the underlying MBeans and configure their values. For more information about accessing the underlying MBeans, see [Section 3.4](#).

You can use Fusion Middleware Control to configure ADF parameters, as described in the following topics:

- [Section 3.3.1, "Modifying ADF Business Components Parameters"](#)
- [Section 3.3.2, "Modifying Connection Configurations"](#)

3.3.1 Modifying ADF Business Components Parameters

You control the runtime behavior of an application module pool by setting appropriate configuration parameters. Fusion Middleware Control provides a UI to configure ADF Business Components, as described in this section. You can also configure the ADF Business Components MBeans directly using the generic MBean Browser, as described in [Section 3.4.3](#)

To modify Business Components parameters:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to configure Business Components.
2. From the Application Deployment menu, choose **ADF > Configure ADF Business Components**.
3. Click an **Application Module**.
4. Click the **Pooling and Scalability**, **Core**, **Database**, or **Security** tabs to update configuration parameters.

If the application module uses data sources, you can configure the data sources by clicking **Edit Datasource** from the **Core** tab.

The ADF Business Components configurations page is arranged with the following sections or tabs:

- Application Modules section
- Pooling and Scalability tab - Application Pool Properties
- Pooling and Scalability tab - Connection Pool Properties
- Core tab
- Database Properties tab
- Security Properties tab

3.3.2 Modifying Connection Configurations

A connection configuration contains information that a client application uses to identify the ADF application module's deployment scenario. You use Oracle Enterprise Manager Fusion Middleware Control to:

- Register and manage back-end services such as mail, discussion forums servers, and so on
- Register and manage external applications that users need access to while working with applications
- Register and manage any portlet producers that the application uses or that users may need access to

Fusion Middleware Control provides a UI to configure ADF connections, as described in this section. You can also configure the ADF connections MBean directly using the System MBean Browser, as described in [Section 3.4.2](#).

You must already have deployed an Oracle ADF application and have Enterprise Manager Fusion Middleware Control available to access the application.

You must have MDS configured in your application before you can modify the ADF application and connection configurations. ADF connection attributes are persisted to MDS.

If you deployed an application to several nodes within a cluster, any ADF connection changes to a single node will be propagated to all the other nodes. MDS will store a single set of connection information for all versions of an application.

To modify connection configurations:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to configure connections.

2. From the Application Deployment menu, choose **ADF > Configure ADF Connections**.
3. In the **Connection Type** drop-down list, choose the type of connection you want to configure:
 - ADF BC Service
 - Enterprise Scheduling Service
 - URL
 - Web Service

You cannot create an Essbase connection, however, you can edit an existing Essbase connection that was deployed with the application.

4. In the **Connection Name** field, enter a unique name for the connection configuration.
5. Click **Create Connection**.

The Connection Configuration page updates with a section where you can specify options for the connection type you chose.

3.4 Configuring Application Properties Using the MBean Browser

You can use the Fusion Middleware Control System MBean Browser to access and modify the values in ADF MBeans deployed with the ADF application into Oracle WebLogic Server.

You can use the Fusion Middleware Control System MBean Browser to perform configuration tasks, as described in the following topics:

- [Section 3.4.1, "Modifying ADF Application Configurations Using MBeans"](#)
- [Section 3.4.2, "Modifying ADF Connections Using MBeans"](#)
- [Section 3.4.3, "Modifying ADF Business Components Configurations Using MBeans"](#)
- [Section 3.4.4, "Modifying MDS Configuration Using MBeans"](#)
- [Section 3.4.5, "Modifying Active Data Service Configuration Using MBeans"](#)

3.4.1 Modifying ADF Application Configurations Using MBeans

You can modify ADF application configurations MBeans using the MBean Browser.

You must have MDS configured in your application before you can modify the ADF application and connection configurations. ADF application attributes are persisted to MDS.

If you deployed an application to several nodes within a cluster, any ADF application configuration changes to a single node via an MBean will be propagated to all the other nodes. MDS will store a single set of ADF application configuration information for all versions of an application.

To modify ADF application configurations using the System MBean Browser:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to configure.
2. From the Application Deployment menu, choose **ADF > Configure ADF (adf-config)**.

- In the left pane of the System MBean Browser, expand the parent ADF MBean **ADFConfig** and then the **ADFConfig** folder to expose the child ADF MBeans. You may see the child ADF MBeans **ADFCConfiguration** and **MDSAppConfig**.
- In the left pane, select the **ADFCConfiguration** MBean, and in the right pane, select the attribute you want to view or modify.

The screenshot shows the System MBean Browser interface. The left pane displays a tree view of the MBean hierarchy, with the following path expanded: `oracle.adf.share.config > Server: AdminServer > Application: uifwksamples-emcust > Application: em > ADFConfig > ADFConfig > ADFCConfiguration`. The right pane, titled "Application Defined MBeans: ADFCo...", shows a table of attributes for the selected MBean.

Name	Description
1 AdfScopeHaSupport	Causes the ADF Controller to and to replicate those change
2 AllowUnboundedTaskFlowCustomizat	Controls whether a session le store unbounded task flow r
3 ConfigMBean	If true, it indicates that this M
4 DetectMetadataChanges	Allows a runtime optimization or customization class chang
5 EnableGrammarValidation	Enables validation of ADFc m additional validation rules enfr
6 EnableImplicitSavepoints	Enables creation of implicit sa
7 eventProvider	If true, it indicates that this M JSR-77.
8 eventTypes	All the event's types emitted

- Change the attribute value and click **Apply**.
- In the left pane, select the parent ADF MBean **ADFConfig**.
- In the right pane, click the **Operations** tab and click **save**.

The new values you have edited are written to MDS after you click **save** from the parent MBean.

3.4.2 Modifying ADF Connections Using MBeans

You can modify ADF connection configurations MBean using the MBean Browser.

You can also modify ADF connections using the Fusion Middleware UI described in [Section 3.3.2](#).

You must have MDS configured in your application before you can modify the ADF application and connection configurations. ADF application attributes are persisted to MDS.

If you deployed an application to several nodes within a cluster, any ADF connection changes to a single node via an MBean will be propagated to all the other nodes. MDS will store a single set of ADF application configuration information for all versions of an application.

To modify ADF connections configurations using the System MBean Browser:

- From the navigation pane, expand **Application Deployments**, then click the application that you want to configure.
- From the Application Deployment menu, choose **System MBean Browser**.

3. In the left pane of the System MBean Browser, navigate to the **ADFConnections** MBean. The MBean should be in **oracle.adf.share.connections > server name > application name**.
4. In the left pane, select the ADF Connections MBean, and in the right pane, select the attribute you want to view or modify.



5. Change the attribute value and click **Apply**.
6. In the right pane, click the **Operations** tab and click **save**.

The new values you have edited are written to MDS after you click **save**.

3.4.3 Modifying ADF Business Components Configurations Using MBeans

You can modify ADF Business Components configurations MBeans using the MBean Browser. ADF Business Component configuration information are stored in MBeans that are specific for each application. Unlike ADF connections and ADF application configuration information which you can configure once for all versions of the same application, you will need to configure ADF Business Components for each version of the application.

You can also modify ADF Business Components configuration information using the Fusion Middleware UI described in [Section 3.3.1](#).

You must have MDS configured in your application before you can modify the ADF application and connection configurations. ADF application attributes are persisted to MDS.

If you deployed an application to several nodes within a cluster, any ADF Business Components changes to a single node via MBeans will be propagated to all the other nodes. MDS will store a single set of ADF application configuration information for all versions of an application.

To modify ADF Business Components configurations using the System MBean Browser:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to configure.
2. From the Application Deployment menu, choose **System MBean Browser**.
3. In the left pane of the System MBean Browser, navigate to the BC4J MBeans. These MBeans should be in `oracle.bc4j.mbean.share > server name > application name`.
4. In the left pane, select the ADF Connections MBean, and in the right pane, select the attribute you want to view or modify.
5. Change the attribute value and click **Apply**.

3.4.4 Modifying MDS Configuration Using MBeans

You can use the MBean Browser to perform advanced configuration of MDS parameters. For more information about configuring MDS using MBeans, see *Administering Oracle Fusion Middleware*.

You must already have deployed an Oracle ADF application and have Enterprise Manager Fusion Middleware Control available to access the application.

To modify MDS configurations using the System MBean Browser:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to configure.
2. From the Application Deployment menu, choose **MDS Configuration**.
3. Click **Configuration MBean Browser** or **Runtime MBean Browser**.
4. Select the MBean and the attribute you want to view or modify.

The screenshot shows the System MBean Browser interface. The left pane displays a tree view of MBeans. The right pane shows the 'Application Defined MBeans: ADFCo...' window with a table of attributes.

Name	Description
1 AppMetadataRepositoryInfo	Metadata repository partition where
2 AutoPurgeTimeToLive	Automatically purge versions of me given time interval specified in seco
3 ConfigMBean	If true, it indicates that this MBean i
4 DeployTargetRepository	The repository where the applicatio
5 eventProvider	If true, it indicates that this MBean i JSR-77.
6 eventTypes	All the event's types emitted by thi
7 ExternalChangeDetection	Enables the application to detect ap performed external to the applicati
8 ExternalChangeDetectionInter	The maximum time interval in seco detect external metadata changes. ExternalChangeDetection is enable
9 MaximumCacheSize	The maximum metadata cache size

5. Change the value and click **Apply**.
6. In the left pane, select the parent ADF MBean **ADFConfig**.
7. In the right pane, click the **Operations** tab and click **save**.

The new values you have edited are written to MDS after you click **save** from the parent MBean.

3.4.5 Modifying Active Data Service Configuration Using MBeans

You can use Active Data Service (ADS) framework to control the runtime behavior of an Oracle ADF application and qualifying ADF Faces components so that whenever data changes on the server, the ADF Model layer notifies the component and the component rerenders the changed data.

You must already have deployed an Oracle ADF application and have Enterprise Manager Fusion Middleware Control available to access the application.

Note that the ADF Faces components of your application must be configured to use ADS. Additionally, if your application services do not support ADS, then your application must define a service proxy so that the components can display the data as it updates in the source. For details about ADS, see the "Using the Active Data Service" section in *Developing Fusion Web Applications with Oracle Application Development Framework*.

To modify Active Data Service configurations using the System MBean Browser:

1. From the navigation pane, expand **Application Deployments**, then click the application that you want to configure.
2. From the Application Deployment menu, choose **ADF > Configure ADF (adf-config)**.
3. In the left pane of the System MBean Browser, expand the parent ADF MBean **ADFConfig** and then the **ADFConfig** folder to expose the child ADF MBeans.

You may see the child ADF MBeans **ActiveDataConfiguration** and **MDSAppConfig**.

System MBean Browser

Application Defined MBeans: ADFConfig:ActiveDataCo... [Apply] [Revert]

Show MBean Information

Attributes		Notifications
Name	Description	
1 ConfigMBean	If true, it indicates that this MBean is a Config MBean.	
2 eventTypes	All the event's types emitted by this MBean.	
3 KeepAliveInterval	Frequency (in ms) for sending keep-alive messages when no events are generated	
4 LatencyThreshold	Latency threshold (in ms): Active Data messages with network delays greater than this threshold will be treated as being 'late'	
5 MaxReconnectAttemptTime	Maximum period of time (in ms) a client will keep attempting to reconnect to server upon getting disconnected	
6 PollingInterval	Frequency (in ms) of the poll request; only used when client are set to use polling	
7 ReadOnly	If true, it indicates that this MBean is a read only MBean.	
8 ReconnectWaitTime	Time interval (in ms) to wait between reconnect attempts	
9 RestartNeeded	Indicates whether a restart is needed.	
10 SystemMBean	If true, it indicates that this MBean is a System MBean.	
11 Transport	Active Data transport mechanism; can be polling, long-pollin or streaming.	
12 UsePolling	This attribute has been deprecated. Flag to indicate whether clients should use polling	

4. In the left pane, select the **ActiveDataConfiguration** MBean, and in the right pane, select the attribute you want to view or modify.

Attribute	Description
Transport	The method by which data will be delivered to the client. Value values are: <ul style="list-style-type: none"> ■ streaming (default) ■ polling ■ long-polling For more information, see "What You May Need to Know About Configuring an ADS Transport Mode" in <i>Developing Fusion Web Applications with Oracle Application Development Framework</i> .
LatencyThreshold	Latency threshold in milliseconds. Active data messages with network delays greater than this threshold will be treated as being "late".
KeepAliveInterval	Frequency in milliseconds for sending keep-alive messages when no events are generated.
PollingInterval	When transport set to polling, frequency in milliseconds of the poll request.
MaxReconnectAttemptTime	Maximum period of time in milliseconds a client will attempt to reconnect the push channel upon getting disconnected
ReconnectWaitTime	Time interval in milliseconds to wait between reconnect attempts.

5. Change the attribute value and click **Apply**.
6. In the left pane, select the parent ADF MBean **ADFConfig**.
7. In the right pane, click the **Operations** tab and click **save**.

The new values you have edited are written to MDS after you click **save** from the parent MBean.

3.5 Editing Credentials Deployed with the Application

You can use Enterprise Manager Fusion Middleware Control to edit credentials that were deployed with an ADF application to the credential store. You can also create new credentials and delete existing credentials.

For ADF applications, the following considerations apply:

- The **Map** name is typically the `adfAppUIId` property defined in the application's `adf-config.xml` file.
- The **Key** name is typically in the format `anonymous#connection`, where `connection` is the connection name.
- The **Credential Type** is **Generic** and it is modeled as a hash map of key-value pairs.

For more information, see "Managing Credentials with Fusion Middleware Control" in the *Securing Applications with Oracle Platform Security Services*.

Deploying ADF Applications

This chapter describes how to deploy Oracle ADF applications packaged as an EAR file to a target application server. The focus is on deploying ADF applications for production and later stage testing. It shows you how to prepare the application server for deployment by installing the ADF runtime. It also describes some of the tools that can be used for deployment including Oracle Enterprise Manager Fusion Middleware Control, scripts, and Ant.

This chapter includes the following sections:

- [Section 4.1, "Introduction to Deploying ADF Applications"](#)
- [Section 4.2, "Preparing the Standalone Application Server for Deployment"](#)
- [Section 4.3, "Deploying Using Oracle Enterprise Manager Fusion Middleware Control"](#)
- [Section 4.4, "Deploying Using Scripting Commands"](#)
- [Section 4.5, "Deploying Using Scripts and Ant"](#)
- [Section 4.6, "Deploying Using the Application Server Administration Tool"](#)

For information about deploying ADF applications for development, see *Developing Fusion Web Applications with Oracle Application Development Framework*.

4.1 Introduction to Deploying ADF Applications

Deployment is the process of packaging application files and artifacts and transferring them to a target application server to be run. During application development using JDeveloper, developers can test the application using the Integrated WebLogic Server that is built into the JDeveloper installation, or they can use JDeveloper to directly deploy to a standalone application server.

After the application has been developed, administrators can deploy the application to production application servers. The tools that the administrators use for production-level deployment are:

- Oracle Enterprise Manager Fusion Middleware Control
- WebLogic Scripting Tool (WLST) commands
- Command scripts and Ant scripts
- Oracle WebLogic Administration Console

This chapter describes the tools and methods that administrators use to deploy ADF applications. For information about deploying ADF applications for development and

testing purposes using JDeveloper, see *Developing Fusion Web Applications with Oracle Application Development Framework*.

If your application uses customizations, you may need to set up the MDS repository in the application server. For more information about MDS, see *Administering Oracle Fusion Middleware*.

Note: Developers, Test, and QA personnel may also use these tools and the methods in this chapter to deploy ADF applications to staging application servers.

4.2 Preparing the Standalone Application Server for Deployment

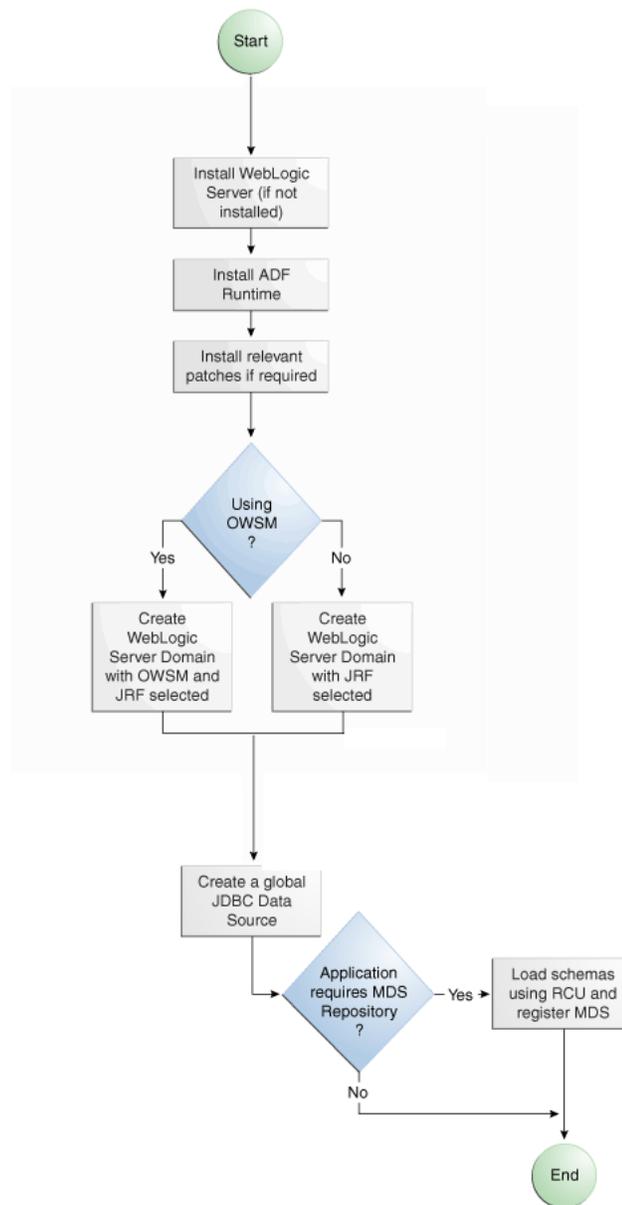
To run ADF applications, you must install the standalone application server with the ADF runtime. You can include the ADF runtime during a new application server installation or you can install the ADF runtime into an existing application server installation.

[Figure 4-1](#) shows the flow diagram for preparing a standalone application server for deployment. Note the following definitions used in the diagram:

- OWSM: Oracle Web Services Manager
- JRF: Java Required Files
- RCU: Repository Creation Utility
- MDS: Metadata Store

The following diagram contains clickable links.

Figure 4–1 Preparing the Application Server Flow Diagram



Preparing the standalone application server for deployment involves the following tasks:

- Install the application server (if not already installed)
- [Section 4.2.1, "Installing the ADF Runtime to the Application Server Installation"](#)
- [Section 4.2.2, "Installing Relevant Patches"](#)
- [Section 4.2.3, "Creating and Extending Oracle WebLogic Server Domains"](#).
- [Section 4.2.4, "Creating a JDBC Data Source for Oracle WebLogic Server"](#)
- Create and Register the MDS Repository (if needed)

For WebLogic Server, the following points apply:

- After WebLogic Server has the ADF runtime installed, you can create a new WebLogic Server domain or you can extend an existing WebLogic Server domain for Oracle ADF.
- If the Managed Servers are on a different host than the Administration Server, you must perform additional configuration tasks for the Managed Servers to enable them to host ADF applications.
- An ADF application will use either a JDBC data source or a JDBC URL to access its data. You can configure WebLogic Server with the data source using the Oracle WebLogic Server Administration Console.
- For additional information about OPatch for patching your existing Oracle Fusion Middleware environment, see *Patching with OPatch*.
- For additional information about creating and loading the schemas that are necessary for many of the Oracle Fusion Middleware components, see *Creating Schemas with the Repository Creation Utility*.
- For the latest information about Oracle Application Development Framework and Oracle JDeveloper, see the *Oracle JDeveloper Release Notes* on Oracle Technology Network.

4.2.1 Installing the ADF Runtime to the Application Server Installation

The application server requires the ADF runtime to run ADF applications.

Installing the ADF runtime is not required if you are using JDeveloper to run applications in Integrated WebLogic Server.

For WebLogic Server, you can install the ADF runtime using the following installers:

- Oracle Fusion Middleware Application Developer Infrastructure Installer: Installs the ADF runtime and Oracle Enterprise Manager. You should use the Oracle Fusion Middleware Infrastructure Installer if you want to use Oracle Enterprise Manager to manage standalone ADF applications (without Oracle SOA Suite or Oracle WebCenter Portal components). For more information, see *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Note: The Oracle JDeveloper Installer can also be used to install the ADF runtime to the application server installation. However, it does not include all the components that are typically needed for production and full test environments. Therefore, this installer should not be used for anything other than for development purposes. For information about the obtaining and using the installer, see *Installing Oracle JDeveloper*.

4.2.1.1 Installing the ADF Runtime into an Existing WebLogic Server Installation Using the Oracle Fusion Middleware Infrastructure Installer

You can use the Oracle Fusion Middleware Infrastructure Installer to install the ADF runtime and Enterprise Manager.

Install Oracle WebLogic Server. You must also have obtained the Oracle Fusion Middleware Infrastructure Installer.

Use the instructions in *Planning an Installation of Oracle Fusion Middleware* to obtain the software, start the installer, and to complete the installation.

In the installer you will perform several tasks including:

- Adding any software updates
- Selecting the WebLogic Server directory for installation
- Verifying installation information

After you have installed the ADF runtime, follow the instructions in [Section 4.2.3](#) to use the Oracle Fusion Middleware Configuration Wizard to create or extend the Oracle WebLogic Server domain.

4.2.2 Installing Relevant Patches

You may need to install patches and other updates to your software distribution. For more information, see *Patching with OPatch*.

4.2.3 Creating and Extending Oracle WebLogic Server Domains

You need to create and configure the Oracle WebLogic Server domain to accept ADF applications. If you do not already have a domain, you need to create one. If you already have a domain, you must extend the domain before it can run ADF applications.

If you are using Managed Servers to run your applications, you may need to configure your Managed Server. For more information about configuring a Managed Server on Oracle WebLogic Server, see *Creating WebLogic Domains Using the Configuration Wizard*.

If you are setting up Managed Servers for ADF where the Managed Servers are on the same host as the Administration Server, follow the instructions described in this section.

If you are setting up to deploy to Managed Servers that are on a different host than the Administration Server, perform the additional steps described in [Section 4.2.3.3](#).

4.2.3.1 Creating an Oracle WebLogic Server Domain for Oracle ADF

It may be helpful to have an understanding of the options that are available to you when you create a WebLogic Server domain for Oracle ADF. For more information, see [Section 4.2](#).

To create a new Oracle WebLogic Server domain:

1. Start the Oracle Fusion Middleware Configuration wizard as described in the "Configuring Your WebLogic Domain" chapter of the *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Follow the directions as described in that guide but consider the following steps.

2. In the Welcome page, select **Create a New Domain** and click **Next**.
3. In the Templates page, select **Create Domain Using Product Templates**.

The option **Basic WebLogic Server Domain - 12.1.2.0 [wlserver]** is already selected.

Select **Oracle JRF - 12.1.2.0 [oracle_common]**. If you are using Oracle Web Services, select **Oracle WSM Policy Manager 12.1.2.0 [oracle_common]** and click **Next**.

4. Continue to follow the directions in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

4.2.3.2 Extending the Oracle WebLogic Server Domain for Oracle ADF

It may be helpful to have an understanding of the options that are available to you when you extend the WebLogic Server domain for Oracle ADF. For more information, see [Section 4.2](#).

You will need to create an Oracle WebLogic Server domain with the ADF runtime installed.

To extend an Oracle WebLogic Server domain for ADF:

1. Start the Oracle Fusion Middleware Configuration wizard as described in the "Configuring Your WebLogic Domain" chapter of the *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Follow the directions as described in that guide but consider the following steps.

2. In the Welcome page, select **Update an existing domain**.
3. Select the location of the domain you want to update for Oracle ADF, and click **Next**.
4. In the Templates page, select **Update Domain Using Product Templates**.

The option **Basic WebLogic Server Domain - 12.1.2.0 [wlserver]** is already selected.

Select **Oracle JRF - 12.1.2.0 [oracle_common]**. If you are using Oracle Web Services, select **Oracle WSM Policy Manager 12.1.2.0 [oracle_common]** and click **Next**.

5. Continue to follow the directions in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

This configures the rest of the runtime .jar files using the manifest file.

Note: Your application's EAR file must have a `weblogic-application.xml` file containing a reference to the `adf.oracle.domain` shared library.

You can now start Oracle WebLogic Server by running the command-line script `ORACLE_HOME\user_projects\domains\domain_name\bin\startWebLogic.cmd`, and you can stop the server using the `stopWebLogic.cmd` script in the same directory. For Linux platforms, use `\bin\startWebLogic.sh` and `stopWebLogic.sh` respectively.

Access the Oracle WebLogic Server Administration Console using the URL `http://localhost:7001/console`.

4.2.3.3 Setting Up Remote WebLogic Managed Servers for Oracle ADF

If the WebLogic Managed Servers are on a different host than the Administration Server, you need to perform additional steps.

You will need to set up Managed Servers for Oracle ADF on the host with the Administration Server, pack the JRF template, copy it to the remote host, and unpack the template.

It may be helpful to have an understanding of the options that are available to you when you create remote WebLogic Managed Servers for Oracle ADF. For more information, see [Section 4.2](#).

For more information about using `pack` and `unpack` commands to set up managed servers, see *Creating Templates and Domains Using the Pack and Unpack Commands*.

You will need to complete this task:

1. Set up Managed Servers for Oracle ADF on the host with the Administration Server.
2. Pack the JRF template.
3. Copy the template to the remote host.
4. Unpack the template

To set up remote Managed Servers for Oracle ADF:

1. Use the Oracle Installer for JDeveloper to install Oracle WebLogic Server installations on both the local and remote hosts, if not already installed. If you are not installing JDeveloper Studio, you need to select the **Application Development Framework Runtime** option in the installer. The local host is the host with the Administration Server.

Or, if there are existing Weblogic Server installations, use the Oracle Installer for JDeveloper to install the ADF runtime into the WebLogic Server installations on both hosts by selecting the **Application Development Framework Runtime** option. For more information on installation, see [Section 4.2.1](#).

2. Run the Oracle Fusion Middleware Configuration Wizard to create a new Oracle WebLogic Server domain. In the wizard, select the **Oracle JRF** option, as described in [Section 4.2.3.1](#).
3. On the local host, run the Oracle Fusion Middleware Configuration Wizard to create Managed Servers.
4. On the local host, start the Administration Server and the Managed Server.

For example,

```
cd ORACLE_HOME/user_projects/domain/base_domain/bin
./startWeblogic.sh
./startManagedWebLogic.sh ManagedServer_1 http://localhost:7001
```

5. On the local host, pack the Managed Server configuration information into a JAR and then copy the JAR to the remote host. This JAR contains the JRF template information.

For example,

```
cd ORACLE_HOME/oracle_home/common/bin

./pack.sh -managed=true -domain=../../../../user_projects/domains/base_domain
         -template=../../../../base_domain_managed.jar -template_name=
         "Base Managed Server Domain"

cp ../../../../../../base_domain_managed.jar remote_machine_ORACLE_HOME/
```

6. On the remote host, unpack the Managed Server configuration JAR.

For example,

```
cd ORACLE_HOME/oracle_common/common/bin
./unpack.sh -domain=../../../../user_projects/domains/base_domain
           -template=../../../../base_domain_managed.jar
```

If the Managed Server was created after the domain was, you must delete the entire domain configuration directory of the Managed Server before running unpack.

7. On the remote host, start the Node Manager.

For example,

```
cd ORACLE_HOME/wlserver_10.3/server/bin
./startNodeManager.sh
```

8. On the remote host, if the Managed Server was not created with the JRF template applied, run the `applyJRF WLST` command to extend the Managed Server with the JRF template.

Also, if the Managed Server was created after the domain was, you must delete the entire domain configuration directory of the Managed Server before running `applyJRF`.

9. On the both hosts, start the Managed Servers.

For example,

```
cd ORACLE_HOME/user_projects/domains/base_domain/bin./startManagedWebLogic.sh
ManagedServer_2 http://<adminServerHost>:7001
```

4.2.4 Creating a JDBC Data Source for Oracle WebLogic Server

Use the Oracle WebLogic Server Administration Console to set up a JDBC data source in the WebLogic Server instance for your applications.

It may be helpful to have an understanding of the options that are available to you when you create a JDBC data source. For more information, see [Section 4.2](#).

To configure Oracle WebLogic Server for a JDBC data source:

1. Start Oracle WebLogic Server (if not already started) by choosing **Oracle Fusion Middleware > User Projects > Domain > Start Admin Server for WebLogic Server Domain** from the Windows **Start** menu.

For Linux, log in as the root user and navigate to:

```
<ORACLE_HOME>/user_projects/domains/MYSOADomain/bin
```

Run the following command:

```
./startWebLogic.sh
```

Or, from the Application Server Navigator, right-click an Oracle WebLogic Server instance and choose **Launch Admin Console**.

2. Start the Oracle WebLogic Server Administration Console by choosing **Oracle Fusion Middleware > User Projects > Domain > Admin Server Console** from the Windows **Start** menu.
3. Log in to the Oracle WebLogic Server Administration Console.
4. In the WebLogic Server Administration Console page, select **JDBC > Data Sources**.
5. Click **New**.
6. In the JDBC Data Source Properties page:
 - In the **Name** field, enter the name of the JDBC data source.
 - In the **JNDI** field, enter the name of the connection in the form `jdbc/connection DS`.

- For the **Database Type**, select **Oracle**.
 - For the **Database Driver**, select **Oracle Driver (thin)**, and click **Next**.
7. In the Transactions Options page, accept the default options and click **Next**.
 8. In the Connection Properties page:
 - For **Database Name**, enter the Oracle SID. For example, `orcl`.
 - For **Host Name**, enter the machine name of the database.
 - Enter the port number used to access the database.
 - Enter the user name and password for the database and click **Next**.
 9. In the Test Database Connection page, click **Test Configuration** to test the connection.
 10. In the Select Targets page, select the server for which the JDBC data source is to be deployed.
 11. Click **Finish**.

Once the data source has been created in Oracle WebLogic Server, it can be used by an application module.

4.3 Deploying Using Oracle Enterprise Manager Fusion Middleware Control

You can use Oracle Enterprise Manager Fusion Middleware Control to deploy the EAR file created in JDeveloper. Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer a farm. For more information about deploying using Fusion Middleware Control, see *Administering Oracle Fusion Middleware*.

4.4 Deploying Using Scripting Commands

Applications or modules can be deployed from JDeveloper without starting the JDeveloper IDE. You can run WLST commands from the command line or sequence them in scripts to run as a batch.

Before deploying from the command line, there must be deployment profiles for the application (EAR) or project (JAR or WAR). JDeveloper creates these deployment profiles automatically for certain types of applications, but before using commands for deployment, it is important to verify that the deployment profile(s) exist. To verify that the profiles exist, choose the **Deployment** node from either the Application Properties or Project Properties dialogs in JDeveloper. For more information about deployment profiles, see *Developing Fusion Web Applications with Oracle Application Development Framework*.

JDeveloper can also be used to deploy an application's EAR, WAR, or JAR files. The same scripts that are used for deployment via a command line are also used to deploy via JDeveloper, but JDeveloper creates the syntax and provides a user interface for the deployment.

There are specific WLST commands (WebLogic) for working with ADF applications. For a list of these commands, see [Chapter 7, "WLST Command Reference for ADF Applications."](#)

For more information about using WLST scripts, see the *WLST Command Reference for Infrastructure Components*.

4.5 Deploying Using Scripts and Ant

You can deploy the application using commands and scripts. You create a script to deploy the application using the `ojdeploy` command and use the `ojaudit` command to audit projects, workspaces, or source files of the application. You can also set up the script to run automatically, for instance, whenever a developer checks in new changes.

`ojdeploy` scripts and Ant scripts can be used together or separately:

1. Create an `ojdeploy` script to compile, package, and deploy the application.
2. Create an `ojdeploy` script to compile and package the application. Then use an Ant script (such as `WLDeploy`) to deploy the application.
3. Create an Ant script to compile, package, and deploy the application. The Ant does not need to use `ojdeploy`.

For more information about the `ojdeploy` and `ojaudit` commands, see the JDeveloper online help.

You can deploy to most application servers from JDeveloper, or use tools provided by the application server vendor. You may also use Ant to package and deploy applications. The `build.xml` file, which contains the deployment commands for Ant, may vary depending on the target application server.

For deployment to other application servers, see the application server's documentation. If your application server does not provide specific Ant tasks, you may be able to use generic Ant tasks. For example, the generic `ear` task creates an EAR file for you.

For information about Ant, see <http://ant.apache.org>.

4.6 Deploying Using the Application Server Administration Tool

For WebLogic, you can use the Oracle WebLogic Server Administration Console to deploy the EAR file created in JDeveloper. For more information, see *Deploying Applications to Oracle WebLogic Server*.

Administering ADF Desktop Integration

This chapter describes the actions the system administrator performs such as deploying the runtime edition client installer of ADF Desktop Integration to a shared location, and adjusting server configuration settings.

As the system administrator, you are responsible for deployment and maintenance of the Fusion web application using ADF Desktop Integration. Note that before an end user can use the integrated Excel workbook, the runtime edition of ADF Desktop Integration must be installed on the end user's system.

You can install the runtime edition of ADF Desktop Integration in any of the following ways:

- For a single user, install the runtime edition from the local path, for example a CD drive or the local directory, on the end user's system (see the "Installing, Upgrading, and Removing the Runtime Edition of ADF Desktop Integration" section of *Developing Applications with Oracle ADF Desktop Integration*).
- For multiple users, install the runtime edition from a shared file server or a web server (see [Section 5.1](#)).

This chapter includes the following sections:

- [Section 5.1, "Installing and Upgrading the Runtime Edition of ADF Desktop Integration From a Shared File Server and Web Server."](#)
- [Section 5.2, "ADF Desktop Integration Logs"](#)
- [Section 5.3, "Security in ADF Desktop Integration"](#)
- [Section 5.4, "Verifying the Client Version of ADF Desktop Integration"](#)
- [Section 5.5, "Verifying Integrated Excel Workbook Metadata"](#)
- [Section 5.6, "Common ADF Desktop Integration Error Messages and Problems"](#)
- [Section 5.7, "Localizing the ADF Desktop Integration Installer"](#)

5.1 Installing and Upgrading the Runtime Edition of ADF Desktop Integration From a Shared File Server and Web Server

Before end users can install the runtime edition, you must make the client installer files accessible to the end users, and provide them the complete path or the link to the `setup.exe` file.

When the ADF Desktop Integration client installer runs, it verifies whether the required software is installed on the system. For more information about the required

software, see the following sections of *Developing Applications with Oracle ADF Desktop Integration*:

- Required Oracle ADF Modules and Third-Party Software
- Configuring Excel to work with ADF Desktop Integration

5.1.1 Making the Runtime Edition of ADF Desktop Integration Available to Multiple End Users

To install the runtime edition for multiple end users, the client installer must be hosted on a shared file server so that all end users use the same version of runtime edition on their systems, and continue to share the same version when the runtime edition is upgraded.

ADF Desktop Integration uses the Microsoft ClickOnce installer technology that enables users to install and run a Windows application with minimum interaction. For more information about Microsoft ClickOnce installer, see the following:

<http://msdn.microsoft.com/en-us/library/71baz9ah.aspx>

To make the Runtime edition of ADF Desktop Integration available to end users:

1. Navigate to the `MW_HOME\oracle_common\modules\oracle.adf.desktopintegration_12.1.3` directory, where `MW_HOME` is the Middleware Home directory.
2. Extract the contents of `adfdi-excel-runtime-client-installer.zip` to a directory on the shared file server accessible to all end users (for example, `\\test_system\Oracle\ADFDI_Client\`).

Note that the fully qualified path of the directory must be less than 248 characters.

3. Provide the complete directory path, or the link, to the end user.

To install the Runtime edition of ADF Desktop Integration:

1. From the end user's system, navigate to the directory on the shared file server, or use the provided link.
2. Run `setup.exe` from the path provided in Step 3.
3. Follow the instructions that appear in the dialog boxes launched by `setup.exe` to successfully install the required components.
4. If prompted, click **Yes** to restart the system and complete the setup of ADF Desktop Integration.

WARNING: After the runtime edition of ADF Desktop Integration is installed, do not delete the directory where the client installer files are copied.

Note: JDeveloper includes the designer edition of ADF Desktop Integration, which is intended to be used by developers creating workbooks. Generally, the designer edition is not made available to the end users.

5.1.2 Upgrading the Runtime Edition of ADF Desktop Integration

You can upgrade the runtime edition of ADF Desktop Integration in two ways.

- Uninstall the old version, and install the new version.

For more information about installing and uninstalling the runtime edition, see the "Installing, Upgrading, and Removing the Runtime Edition of ADF Desktop Integration" section of *Developing Applications with Oracle ADF Desktop Integration*.

- Upgrade automatically from a shared file server.

If the runtime edition is installed on multiple end users' systems from a shared file server, they are not required to upgrade individually as they will be upgraded automatically when the files on the shared file server are upgraded.

Upgrading the runtime edition of ADF Desktop Integration is an easy process and can be done by copying the new version client installer files over the old version client installer files.

Note: The client installer should be upgraded after the server patches are deployed.

To make the Runtime edition upgrade available to end users:

1. If available, apply the Fusion Middleware patch as per the standard patching instructions.

As the system administrator, you should complete the server patching deployment before making the client upgrade available to end users. For more information, see *Patching with OPatch*.

2. Navigate to the `MW_HOME\oracle_common\modules\oracle.adf.desktopintegration_12.1.3` directory of the upgraded or patched Fusion Middleware installation, where `MW_HOME` is the Middleware Home directory.
3. Extract the contents of `adfdi-excel-runtime-client-installer.zip` to the exact same location on the shared file server that was used for previous installations. Some installation files will be overwritten and the new files will be added.

Ensure that the fully qualified path to the folder to which `adfdi-excel-runtime-client-installer.zip` gets extracted remains the same.

End users will automatically get the updated ADF Desktop Integration client files copied from the shared location when they next launch Excel, provided 24 hours have elapsed since Excel was last launched. For more information, see [Section 5.1.6](#).

5.1.3 Installing the Runtime Edition of ADF Desktop Integration From a Web Server

If you want to provide the runtime edition setup files from a web server instead of a shared file server, you must update the path recorded in the client installer `setup.exe` file before you run it. By default, the installation path in the `setup.exe` file is configured for local installation.

To make the installation files available from a web server:

1. Copy the full set of installation files to the web server, as described in Step 1, 2, and 3 of [Section 5.1.2](#).
2. Run the `setup.exe` file on a Windows system using the following command syntax:

```
setup.exe /url="URL of web server where installation files are copied"
```

For example:

```
setup.exe /url="http://www.myserver/adfdilocation"
```

Note: The `setup.exe` command must be run on the Windows-based system, even though the web server that hosts the installer is based on other operating systems such as Linux. In such a case, after executing Step 2, copy `setup.exe` to the web server.

Changing the path recorded in the client installer `setup.exe` file invalidates its digital signature, and you may see the Unknown Publisher security warning message when you run it. To avoid the error message, acquire a digital certificate for the `setup.exe` file and re-sign it after updating the installation path. For more information, see "How to: Sign Application and Deployment Manifests" on Microsoft Developer Network at <http://msdn.microsoft.com/en-us/library/default.aspx>.

Note: When you update the recorded installation path, only the digital signature of `setup.exe` is invalidated. The signatures of the remaining setup files are not affected.

5.1.4 Upgrading the Runtime Edition of ADF Desktop Integration From a Web Server

If the runtime edition is installed on multiple end users' systems from a web server, they are not required to upgrade individually as they will be upgraded automatically when the files on the web file server are upgraded.

As described in [Section 5.1.3](#), copy the new version client installer files over the old version client installer files on the web server. Excel will check for updates when it is launched and if 24 hours have elapsed since the last check for updates. If a new version is found on the server, ADF Desktop Integration automatically upgrades itself. For more information, see [Section 5.1.6](#).

Note: Before upgrading, if available, apply the Fusion Middleware patch as per the standard patching instructions.

Alternatively, you can also upgrade by uninstalling the old version (see the "Installing, Upgrading, and Removing the Runtime Edition of ADF Desktop Integration" section of *Developing Applications with Oracle ADF Desktop Integration*), and then installing the latest version (see [Section 5.1.3](#)).

If an end user tries to upgrade the Runtime edition of ADF Desktop Integration manually, as described in the "How to Upgrade the Runtime Edition of ADF Desktop Integration On a Local System" section of *Developing Applications with Oracle ADF Desktop Integration*, the **Update Information** link might not run the installer but open as an XML file in the browser of the end user's desktop. To solve this issue, you should configure the MIME mapping setting of the web server that hosts the client and add the following mappings to allow the `vsto`, `application`, `manifest`, and `deploy` file extensions to run:

```
<mime-mapping>
  <extension>vsto</extension>
  <mime-type>application/x-ms-vsto</mime-type>
</mime-mapping>
<mime-mapping>
```

```

    <extension>application</extension>
    <mime-type>application/x-ms-application</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>manifest</extension>
    <mime-type>application/x-ms-manifest</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>deploy</extension>
    <mime-type>application/octet-stream</mime-type>
</mime-mapping>

```

After configuring the MIME mapping setting of the web server, provide instructions to the end user to open the **Update Information** link in Internet Explorer as other browsers are not supported.

5.1.5 Patching the Runtime Edition of ADF Desktop Integration

Patching involves copying a small collection of files over an existing installation. ADF Desktop Integration patch updates are delivered with Oracle Fusion Middleware patch updates. For more information, see *Patching with OPatch*.

To patch an existing ADF Desktop Integration framework of Oracle Fusion Middleware 11g Patch Set 1 (Release 11.1.1.2.0) or Patch Set 2 (Release 11.1.1.3.0) environment to make it a Release 12.1.3.0.0 environment, you must upgrade the JRF domains to accommodate changes to ADF shared libraries. For more information see the chapter on post-patching tasks for JRF Infrastructure and ADF shared libraries in *Patching with OPatch*.

5.1.6 What Happens When Runtime Edition of ADF Desktop Integration Is Installed and Upgraded

When the runtime edition of ADF Desktop Integration is installed on the end user's system, it records a link back to the original install directory. After the installation, each time Excel is launched, Excel periodically checks the original installation directory to see whether a new version is available before activating the ADF Desktop Integration add-in. If a new version is found, Excel automatically upgrades the ADF Desktop Integration add-in.

Note: ADF Desktop Integration is configured to check for a new version once per day. Depending on when Excel is launched, it may take up to one day, or more, for ADF Desktop Integration add-in to upgrade. Remember that ADF Desktop Integration is upgraded only when Excel is launched.

5.2 ADF Desktop Integration Logs

Based on various client and server events, logs are generated by ADF Desktop Integration. For more information about client-side, see the "Generating Log Files for an Integrated Excel Workbook" section in *Developing Applications with Oracle ADF Desktop Integration*.

5.2.1 Server-Side Logging

You configure the generation of server-side log files for ADF Desktop Integration the same way as for other Oracle ADF modules. This involves setting values that specify the verbosity level and output location in a Administering Oracle ADF Applications configuration file named `logging.xml`. You can also use Oracle Diagnostic Logging Configuration of JDeveloper to configure the logging levels specified in the `logging.xml` file. For more information about using the JDeveloper debugging tools and ADF Logger, see the "Using the ADF Logger" section in *Developing Fusion Web Applications with Oracle Application Development Framework*.

Table 5–1 describes the package names that you supply as attribute parameters to the `<logger>` elements in the `logging.xml` file to configure log file generation in ADF Desktop Integration.

Table 5–1 Package Names for Log File Configuration

To generate log file entries for this component...	Enter this package name...
All ADF Desktop Integration server logic	<code>oracle.adf.desktopintegration</code>
ADF Desktop Integration HTTP filter	<code>oracle.adf.desktopintegration.filter</code>

For more information about server-side logging levels, see the "Using the ADF Logger" section in *Developing Fusion Web Applications with Oracle Application Development Framework*.

5.2.2 Using the Oracle Diagnostics Log Analyzer to Analyze ADF Desktop Integration Servlet Requests

Using the Oracle Diagnostics Log Analyzer, you can view a hierarchical breakdown of elapsed time spent performing each ADF Desktop Integration servlet request. The hierarchical breakdown also includes the time spent in other ADF components, such as the ADF model layer. For more information about using the log analyzer for viewing web requests, see the "How to Use the Log Analyzer to View Log Messages" section in *Developing Fusion Web Applications with Oracle Application Development Framework*.

Tip: The hierarchical breakdown can be helpful in identifying performance bottlenecks due to unusually long execution times.

In order to log a complete hierarchy tree of ADF event messages, including ADF Desktop Integration events, specify log level `CONFIG` for the `oracle.adfdiagnostics` package. For more detail about the `oracle.adfdiagnostics` logger, see the "How to Create an Oracle ADF Debugging Configuration" section in *Developing Fusion Web Applications with Oracle Application Development Framework*.

5.3 Security in ADF Desktop Integration

If your Fusion web application enforces authentication, the integrated Excel workbooks also ensure that an authenticated user session is established before data transfer happens between the workbooks and application. For more information, see the "About Security In Your Integrated Excel Workbook" section in *Developing Applications with Oracle ADF Desktop Integration*.

5.3.1 End User Authentication

If end users are not prompted for user credentials while using integrated workbooks and interacting with a secure Fusion web application, you need to investigate the security configuration of the Fusion web application. For more information, see the "Verifying End-User Authentication for Integrated Excel Workbooks" section in *Developing Applications with Oracle ADF Desktop Integration*.

For more information about ADF Desktop Integration security, see the "Oracle ADF Desktop Integration Security whitepaper" on OTN at:

<http://www.oracle.com/technetwork/developer-tools/adf/overview/index-085534.html>

5.3.2 What You May Need to Know About Configuring Security in a Fusion Web Application

You should note the following points before you secure your application:

- For applications running in an environment using Oracle Access Manager, the system administrator should ensure that the URL for the ADF Desktop Integration Remote servlet is configured as a protected resource for Oracle Access Manager.

For more information, see the *Administrator's Guide for Oracle Access Management*.

- For applications running in an environment using WebGate 11g, set the user-defined parameter `filterOAMAuthnCookie` to `False`.

For more information, see the chapter on registering partners (agents and applications) remotely in the *Administrator's Guide for Oracle Access Management*.

- Ensure that applications using ADF Desktop Integration have a security constraint configured in `web.xml` that protects the ADF Desktop Integration remote servlet.

The following code extract from `web.xml` shows an example security constraint protecting the remote servlet:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>adfdiRemote</web-resource-name>
    <url-pattern>/adfdiRemoteServlet</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>valid-users</role-name>
  </auth-constraint>
</security-constraint>
```

- When using Oracle WebGate and a SSL URL to access the Fusion web application (such as `https://...`) it may be necessary to configure WebGate's `mod_wl_ohs.conf` configuration file as follows:

```
<IfModule mod_weblogic.c>
  WLProxySSLPassThrough ON
  WLProxySSL ON
  MatchExpression /TestApp
  WebLogicHost=test.host.com|WebLogicPort=7101|
</IfModule>
```

where `/TestApp` is the context root of your application, `test.host.com` is the host name and domain, and `7101` is the port number for the web application.

- When opening the integrated Excel Workbook, or any Microsoft Office document, directly (without downloading the file) from a link in the Fusion web application, the Windows Login dialog may appear twice asking for user credentials. This happens because Microsoft Office send its own authentication request to the web server, making the Login dialog appear twice. End users may click **Cancel** and ignore the authentication request.
- Applications secured via a digital certificate where clients use https URLs to access the application should ensure that the certificate is valid. Valid certificates have host names that match the host to which they are deployed, have not expired, and have a valid path to a trusted issuing authority. In the case where the certificate is not valid, the client will be prompted during login to accept the invalid certificate.
- ADF uses chunked encoding for some requests to the server. If you have any network devices between Excel and the web application server configured to block requests that do not contain a content length header, you should configure them to allow chunked encoding (no content length header). Some network devices such as content caching servers may have a default configuration that blocks requests with no content length header.

For more information about securing integrated Excel workbooks, see the "What You May Need to Know About Securing an Integrated Excel Workbook" section in *Developing Applications with Oracle ADF Desktop Integration*.

5.3.3 What You May Need to Know About Resource Grants for Web Pages

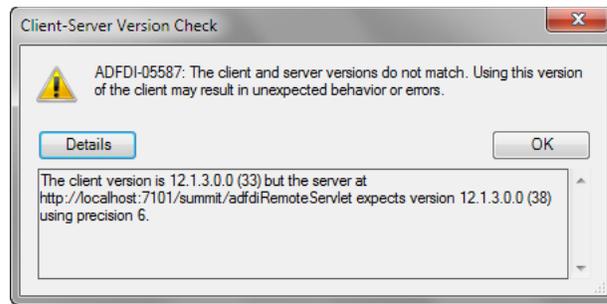
In an integrated Excel workbook, each worksheet is bound to a specific page definition. Users' access to pages may be controlled by resource grants. If an end user is not authorized to work with a page definition, ADF Desktop Integration disallows all data transactions in worksheets bound to that page definition, displays a failure message, and disables those integrated worksheets. The end user can alter any existing data in the worksheet, but cannot download or upload it. The tracking of changes in ADF Table components is also disabled. The end user can continue to use ADF Desktop Integration features in other worksheets in the same workbook, provided those worksheets are bound to page definitions that the end user is authorized to work with.

The worksheet is re-enabled when the end user reopens the workbook and establishes a new session, provided the new session is authorized.

For more information about securing your Fusion web application, see the "Enabling ADF Security in a Fusion Web Application" chapter in *Developing Fusion Web Applications with Oracle Application Development Framework*.

5.4 Verifying the Client Version of ADF Desktop Integration

When an end user establishes a connection with the Fusion web application from the runtime integrated workbook, ADF Desktop Integration verifies whether the client and the server versions are same, and issues a warning dialog if they do not match (see [Figure 5-1](#)).

Figure 5–1 Client-Server Version Check Warning Dialog

The client-server version check succeeds when the following components of the client version match the server version:

- Oracle Release number: It consists of five version elements separated by a period character. For example: 12.1.3.0.0.
- Build number: An integer assigned to the ADF Desktop Integration client and server components during the build process. For example, 33.

If the end user dismisses the warning dialog, ADF Desktop Integration attempts to continue to function normally and no ADF Desktop Integration features are disabled.

Notes:

- The warning dialog appears once per Excel session for a given combination of Fusion web application and client-server mismatch, regardless of how many times the user session is established or re-established.
- When the client version is newer than the server version, an error message (ADFDI-00127) might appear, instead of the warning dialog.

For more information about the error message, see the "Common ADF Desktop Integration Error Messages and Problems" section of *Developing Applications with Oracle ADF Desktop Integration*.

5.4.1 Configuring Client-Server Version Precision

You can configure the precision at which the client-server versions are checked, or disable the feature, before deploying the Fusion web application.

To configure the client-server version precision in the Fusion web application:

1. Open the `web.xml` file of your Fusion web application
2. Add the following initialization parameter to configure the version precision.

Property	Value
Name	Enter the name of the initialization parameter as follows: <code>ClientServerVersionCheck.Precision</code> Note that the name is case-sensitive.

Property	Value
Value	<p>Set the value of <code>ClientServerVersionCheck.Precision</code> to a number between 0 and 6. The default value is 3.</p> <ul style="list-style-type: none"> ■ 0 – Do not verify client-server versions. ■ 1 through 5 – Verify the number of elements of the version from left to right. For example, a value of 4 verifies "12.1.3.0" portion from "12.1.3.0.0". ■ 6 – Verify complete version, including the build number. <p>If the parameter is not specified, ADF Desktop Integration sets the default value for the parameter.</p>

3. Save the `web.xml` file.

The `web.xml` file of your Fusion web application has the following entries:

```
<servlet>
<servlet-name>adfdiRemote</servlet-name>
  <servlet-class>
    oracle.adf.desktopintegration.servlet.DIRemoteServlet
  </servlet-class>
  <init-param>
    <param-name>ClientServerVersionCheck.Precision</param-name>
    <param-value>2</param-value>
  </init-param>
</servlet>
```

4. Rebuild and restart your Fusion web application.

5.4.2 What You May Need to Know About Client-Server Version Verification

Note the following points about client-server version verification in an ADF Desktop Integration project:

- If enabled, the client-server version verification is performed every time when the integrated workbook establishes a session with the Fusion web application.

The verification process is independent of the ADF security settings of the Fusion web application.
- The client-server version verification is not performed in design mode.
- When the client and server version components match, or the feature has been disabled for a particular application, there is no visible effect to end users.
- The ADF Desktop Integration version running on the server can change at any time (for example, server upgrade), but the version verification is performed only when the user session is re-established.

5.5 Verifying Integrated Excel Workbook Metadata

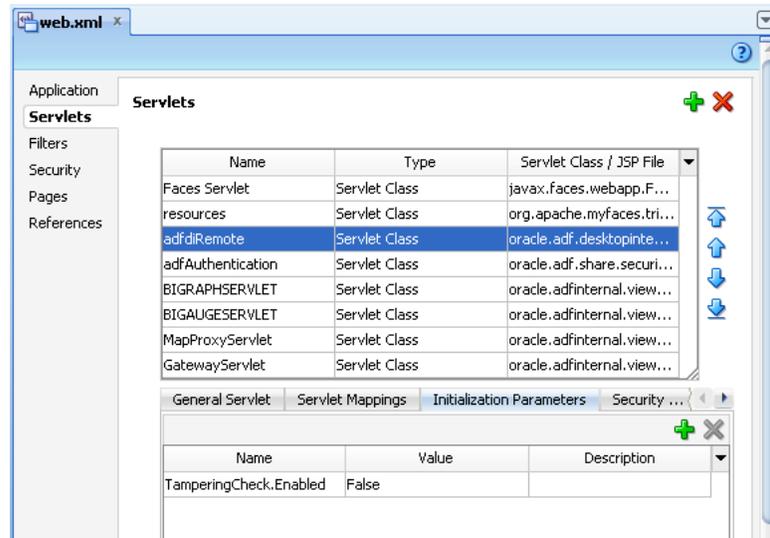
To give end users the confidence that the workbook configuration has not been altered maliciously, ADF Desktop Integration verifies the integrity of the workbook metadata automatically using the *Tamper-Check* feature. For more information, see the "Checking the Integrity of an Integrated Excel Workbook's Metadata" section in *Developing Applications with Oracle ADF Desktop Integration*.

5.5.1 Disabling the Metadata Tamper-Check in the Fusion Web Application

By default, ADF Desktop Integration verifies that the workbook configuration metadata is not tampered with after the workbook developers publish the Excel workbook for end users. You can disable the metadata tamper-check by configuring a parameter in the deployment descriptor file (`web.xml`) of the Fusion web application.

To disable the metadata tamper-check in the Fusion web application:

1. Open the `web.xml` file of your Fusion web application.
2. Add an initialization parameter to the `adfdiRemote` servlet to disable the metadata tamper-check.



Configure the following options to disable metadata tamper-check.

Property	Value
Name	Enter the name of the initialization parameter as follows: <code>TamperingCheck.Enabled</code> Note that the name is case-sensitive.
Value	Set the value of <code>TamperingCheck.Enabled</code> to <code>False</code> . Note that any value other than <code>False</code> will be interpreted as <code>True</code> .

3. Save the `web.xml` file.

The `web.xml` file of your Fusion web application has the following entries:

```
<servlet>
  <servlet-name>adfdiRemote</servlet-name>
  <servlet-class>...</servlet-class>
  <init-param>
    <param-name>TamperingCheck.Enabled</param-name>
    <param-value>False</param-value>
  </init-param>
</servlet>
```

4. Rebuild and restart your Fusion web application.

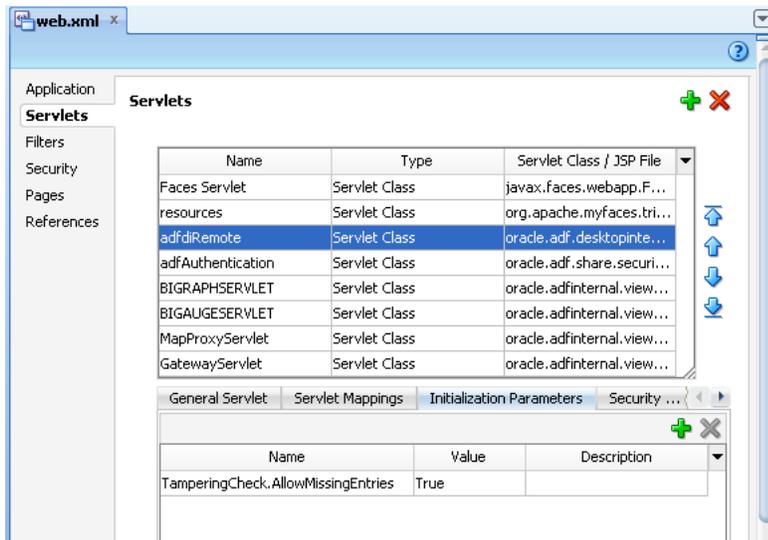
If the `TamperingCheck.Enabled` parameter is not present in `web.xml`, tamper check is enabled. For more information about the `web.xml` file, see the "ADF Desktop Integration Settings in the Web Application Deployment Descriptor" appendix in *Developing Applications with Oracle ADF Desktop Integration*.

5.5.2 Allowing Missing Entries in the ADF Desktop Integration Client Registry

You can configure the metadata tamper-check so that a missing entry for the `WorkbookID` workbook property is allowed in the `adfdi-client-registry.xml` file.

To allow missing entries in the metadata of the Fusion web application:

1. Open the `web.xml` file of your Fusion web application.
2. Add an initialization parameter to the `adfdiRemote` servlet to allow missing entries in the metadata.



Configure the following options to allow missing entries in the metadata.

Property	Value
Name	Enter the name of the initialization parameter as follows: <code>TamperingCheck.AllowMissingEntries</code> Note that the name is case-sensitive.
Value	Set the value of <code>TamperingCheck.AllowMissingEntries</code> to <code>True</code> . Any value other than <code>True</code> will be interpreted as <code>False</code> .

3. Save the `web.xml` file.

The `web.xml` file of your Fusion web application has the following entries:

```
<servlet>
  <servlet-name>adfdiRemote</servlet-name>
  <servlet-class>...</servlet-class>
  <init-param>
    <param-name>TamperingCheck.AllowMissingEntries</param-name>
    <param-value>True</param-value>
  </init-param>
</servlet>
```

4. Rebuild and restart your Fusion web application.

If the `TamperingCheck.AllowMissingEntries` parameter is not present in `web.xml`, missing entries are not allowed. For more information about the `web.xml` file, see the "ADF Desktop Integration Settings in the Web Application Deployment Descriptor" appendix in *Developing Applications with Oracle ADF Desktop Integration*.

5.6 Common ADF Desktop Integration Error Messages and Problems

While using or configuring the ADF Desktop Integration enabled Fusion web application or workbooks, you might see error messages or have some problems. The following list describes the most common error messages, their cause, and solutions.

If you are a system administrator, you should also see the "Common ADF Desktop Integration Error Messages and Problems" section in *Developing Applications with Oracle ADF Desktop Integration*.

Error message: Access to this web server is disabled because it is controlled by basic authentication and does not use Secure Socket Layer (SSL).

Cause: The user downloads and opens a workbook, published using Microsoft Office 2010, from the Fusion web application secured using basic authentication.

Action: By default, Microsoft Office 2010 applications disable basic authentication over a non-SSL connection. Use any of the following methods to resolve the error:

- Use form based authentication instead of basic authentication in the Fusion web application.
- Download and save the workbook before opening it.
- Enable SSL encryption.

If you want to use basic authentication without SSL, see Microsoft Support solution at <http://support.microsoft.com/kb/2123563>.

Error message: UnableToEstablishUnauthenticatedSessionException: ADFDI-00502: The client was unable to establish an unauthenticated session with the web application

Cause: Incorrect security configuration in the Fusion web application.

Action: Review and correct the security configuration. Ensure that the `/adfdiRemoteServlet` URL is protected by a `<security-constraint>` in `web.xml`.

If SSL is used with Oracle WebGate, you might also need to verify the settings in the `mod_wl_ohs.conf` file.

For example:

```
<IfModule mod_weblogic.c>
  WLProxySSLPassThrough ON
  WLProxySSL ON
  MatchExpression /DepthTestingROT WebLogicHost=test.host.com|WebLogicPort=7101|
</IfModule>
```

where `/TestApp` is the context root of your application, `test.host.com` is the host name and domain, and `7101` is the port number for the web application.

Problem: Edit Options dialog appears prompting for WebAppRoot when downloading an integrated Excel workbook from a Fusion web application

Cause: The `adfdiExcelDownload` filter is not properly configured in `web.xml`, and so the filter is not able to set the **WebAppRoot** property on the downloaded workbook.

Action: Ensure that the `adfdiExcelDownload` filter is properly configured in `web.xml`. Verify that the filter is listed in the correct order with respect to the ADF Library Web Application Support, if it is in use. Also verify that the filter mappings for the `adfdiExcelDownload` filter are correct (see the "Configuring the ADF Desktop Integration Excel Download Filter" section in *Developing Applications with Oracle ADF Desktop Integration*.) You should also clear the directory into which browser downloads files.

Problem: Login window does not close after submitting valid credentials in Oracle Access Manager environment

Cause: The `/myApp/adfdiRemoteServlet` was not properly added as a protected resource to the Oracle Access Manager configuration.

Action: Add `/myApp/adfdiRemoteServlet` as a protected resource to the Oracle Access Manager configuration.

For more information, see the chapter on managing policies to protect resources and enable SSO in the *Administrator's Guide for Oracle Access Management*.

Problem: UserSessionRequiredException on login in Oracle Access Manager environment with WebGate 11g

Cause: The user defined parameter `filterOAMAuthnCookie` is not set in the WebGate 11g configuration.

Action: Set `filterOAMAuthnCookie` to `false` in the WebGate 11g configuration.

For more information, see the chapter on registering partners (agents and applications) remotely in the *Administrator's Guide for Oracle Access Management*.

Problem: The DIRemoteServlet returns Invalid XML: unexpected end of response error message

Cause: An exception has occurred in the ADF Model code, custom application module, or in the view object.

Action: Check the server logs for more information.

5.7 Localizing the ADF Desktop Integration Installer

ADF Desktop Integration uses Microsoft ClickOnce and Visual Studio Tools for Office technologies for installing the client. By default, the installer described in the "Installing ADF Desktop Integration" section of the *Developing Applications with Oracle ADF Desktop Integration*, runs in English. To make the installer run in a different language, you should install the respective Visual Studio Tools for Office language pack on all end-users' systems. The language pack can be downloaded and installed from the Microsoft Download Center.

This section assumes that no instance of ADF Desktop Integration is present on the end users system and that the system uses a non-English version of the operating system. If ADF Desktop Integration is present, remove it as described in the "Removing ADF Desktop Integration" section of the *Developing Applications with Oracle ADF Desktop Integration*.

For information about supported operating systems, see the "Required Oracle ADF Modules and Third-Party Software" section of the *Developing Applications with Oracle ADF Desktop Integration*.

To localize the ADF Desktop Integration installer:

1. Download the appropriate language pack (for example, French) for Microsoft Visual Studio 2010 Tools for the Microsoft Office System (Version 4.0 Runtime) Language Pack from the Microsoft Download Center at:
<http://www.microsoft.com/downloads/>
2. Install the language pack that you downloaded in Step 1.
3. Set up ADF Desktop Integration, as described in the "Installing, Upgrading, and Removing the Runtime Edition of ADF Desktop Integration" section of the *Developing Applications with Oracle ADF Desktop Integration*.

Diagnostic Tools

This chapter describes the diagnostic and monitoring tools you can use to help you develop and diagnose Fusion Web applications problems.

This chapter includes the following sections:

- [Section 6.2, "Click History"](#)
- [Section 6.3, "ADF Logs"](#)
- [Section 6.4, "Diagnosing Problems using the Diagnostic Framework"](#)
- [Section 6.5, "Finding ADF Runtime Version Information"](#)
- [Section 6.6, "Troubleshooting Oracle ADF for High Availability"](#)

6.1 Introduction to Monitoring and Diagnostic Tools

Oracle Fusion Middleware and ADF provide tools you can use to monitor and diagnose issues with your application.

The tools and procedures include:

- Click History
- Loggers
- Diagnostics Framework

6.2 Click History

Click History is a Fusion Middleware diagnostics feature that captures ADF Faces events generated by user activities on ADF Faces components (for instance, clicking an ADF command button from a web browser). The data is continuously captured and stored in a circular buffer in memory. The buffer size is adjustable. When a diagnostic incident occurs, the data in the buffer is automatically written to disk as part of the diagnostic data. For more information about diagnostic incidents, see the "Diagnosing Problems" chapter of *Administering Oracle Fusion Middleware*. You do not need to configure Click History to capture the "click" events, it is automatically enabled by default. You can use Click History data to help diagnose problems. It is particularly useful to help determine the last user action that had caused the failure or problem.

The Click History record provides information about the UI component, the view, the region, the user, the Diagnostic Session Id (DSID), and other key application data. Click History captures its data using ADF EUM (End User Monitoring) services. It has a circular buffer that is optimized specifically for EUM data. The data is packed in a single contiguous memory segment in the Java heap. It has minimal impact to Java

Garbage Collection and its memory usage is strictly controlled. The default buffer size of 860 KB can hold approximately 1,500 "click" events, which should be sufficient for incident analysis.

6.2.1 How to Obtain Click History

Click History is delivered as part of the default JRF Template. When you install Fusion Oracle Middleware or Oracle ADF, Click History will be installed as part of the Fusion Middleware framework.

6.2.2 How to Enable Click History for an Application

Click History is installed as part of ADF. However, the actual logging of Click History messages are performed by a separate thread independent of ADF, therefore, Click History imposes insignificant overhead for ADF.

Click History is enabled on an application by application basis. You enable an application for Click History by adding entries to its `weblogic.xml` and `web.xml` configuration files. For instructions, see the "Enabling the Application for RUEI and Click History" section in *Developing Fusion Web Applications with Oracle Application Development Framework*.

6.2.3 How to Enable and Disable Click History in WebLogic Server

After you have enabled Click History for an application, you can enable it for a specific WebLogic server instance by setting the `oracle.clickhistory.EUM` logger to the `NOTIFICATION` level.

You can use a WLST `setLogLevel` command or the Enterprise Manager Cloud Control Support Workbench to set the logging level.

You do not need to restart the server for this to take effect.

To enable Click History using WLST, run the following command:

```
setLogLevel(logger='oracle.clickhistory.EUM', target='server_name', level='INFO', persist='1')
```

To disable Click History using WLST, run the following command:

```
setLogLevel(logger='oracle.clickhistory.EUM', target='server_name', level='OFF', persist='1')
```

For information on how to use Fusion Middleware Cloud Control Support Workbench to set the log level to enable Click History, see the "Managing Log Files and Diagnostic Data" section in *Administering Oracle Fusion Middleware*.

6.2.4 How to Adjust Click History Capacity

The Click History buffer size is defined in the `logging.xml` file. The default value is 860 KB or 880640 bytes. You can use the WLST `configureLogHandler` command to configure the buffer size (in bytes). You will need to restart the server for the change to take effect.

For example:

```
configureLogHandler(name="apps-clickhistory-handler", propertyName="bufferSize", propertyValue="10000000", target="your_server_name")
```

You can find the existing buffer size using the WLST `listLogHandlers` command.

For example:

```
listLogHandlers(name="apps-clickhistory-handler")
```

The command returns a list of information including `bufferSize` in bytes.

For example:

```
bufferSize=880640
```

6.2.5 How to Dump Click History Data Explicitly

When an incident occurs, Click History automatically dumps its data into a file. However, you can also access Click History data at any time using the WLST `executeDump` command.

To dump Click History data explicitly:

```
executeDump(name="odl.quicktrace", args={"handlerName" :
"apps-clickhistory-handler"}, server="CRMCommonServer_1")
```

The Click History output file is a QuickTrace dump file. It has the same format as the ODL log file. This format can be parsed using the ODL LogQuery API as well as using the Enterprise Manager Cloud Control Support Workbench log query feature.

The Click History dump file format convention is:

```
odl_quicktrace<unique ID>_i<incident ID>.app-clickhistory-handler.dmp
```

The suffix is the QuickTrace handler name defined in the server's `logging.xml`.

For example:

```
odl_quicktrace502_i273.app-clickhistory-handler.dmp
```

A sample dump file output is shown in [Example 6-1](#).

Example 6-1 Sample Click History Dump

```
[2013-10-08T14:56:12.664-07:00] [AdminServer] [NOTIFICATION] [SOC]
[oracle.clickhistory.EUM] [tid: 34] [userId: CASH_MANAGER]
[ecid:a3da34f168f6c0de:-7bd21c26:14199e687e9:-8000-00000000000033d3,0]
[CH_RTY: oracle.adf.rich.ComboBox ListOfValues]
[CH_CST: 1381269371002] [DSID: 0000K6PzLBrFS8G5Iz9Dif1IL7HS00000A]
[CH_CID: pt1:USma:0:MAnt2:1:pt1:Manual:0:mrpanel1:mrbankAccountNameId]
[CH_VID: /ManualReconPage] [APP: PayablesApp]
[CH_CMP: oracle.adf.RichInputComboBoxListOfValues]
[CH_CNM: Bank Account]
[CH_TYP: lovInternal]
[CH_WID: 114iztpu6s_6]
[CH_RVD: /ManualReconLocalAreaFlow/ManualReconLocalAreaPF]
[CH_EID: a3da34f168f6c0de:-7bd21c26:14199e687e9:-8000-00000000000033d3]
[CH_FAM: org.apache.myfaces.trinidad.Input]

[2013-10-08T14:56:12.664-07:00] [AdminServer] [NOTIFICATION] [EOC]
[oracle.clickhistory.EUM] [tid: 34] [userId: CASH_MANAGER]
[ecid: a3da34f168f6c0de:-7bd21c26:14199e687e9:-8000-000000000000339c,0]
[CH_CET: 1381269364245]
```

```
[CH_CST: 1381269363617]
[APP: PayablesApp]
[DSID: 0000K6PzLBrFS8G5Iz9Dif1IL7HS00000A]
[CH_TTT: 628]
[CH_EID: a3da34f168f6c0de:-7bd21c26:14199e687e9:-8000-000000000000339c]
```

6.2.6 How to Interpret Click History Records

The Click History dump file is an ordinary ASCII file in the format of a log file. You can use any ASCII viewer to view its contents, but it would be best to use log file viewing tools to take advantage of the search and formatting functions they offer. The Click History messages written to the log file are logged by a separate thread that is independent of ADF.

The log file is located at:

```
${domain.home}/servers/${weblogic.Name}/logs/${weblogic.Name}-clickhistory.log
```

The Click History log data is automatically captured as part of a diagnostic incident. If you have an incident that includes Click History data, you can use the incident's `readme.txt` file to find the ECID (ExecutionContext ID), DSID (Diagnostics Session ID), user ID, and thread ID of the incident. You can also query Click History records based on these diagnostics flags.

The following information are logged:

- Region/View Id
- Component Id
- Server time
- Network time
- Browser render time

By default, a pair of messages are logged for each click. The message Id for the first message is SOC (Start of Click). The second message EOC (End of Click) will not be logged until the subsequent click occurs. Timing details, such as TTT (Total Time), can be found only in the second message.

[Table 6–1](#) lists the Click History attributes and their corresponding ADF `UserActivityInfo` fields.

Table 6–1 Click History Attribute Mapping

Click History Attributes	ADF <code>UserActivityInfo</code> fields
CH_CST	<code>userActivityInfo.getClientStartTime()</code>
CH_CET	<code>userActivityInfo.getClientEndTime()</code>
CH_VID	<code>userActivityInfo.getViewId()</code>
CH_WID	<code>userActivityInfo.getWindowId()</code>
CH_CID	<code>eventInfo.getComponentClientId()</code>
CH_CNМ	<code>eventInfo.getComponentDisplayName()</code>
CH_FAM	<code>eventInfo.getComponentFamily()</code>
CH_CMP	<code>eventInfo.getComponentType()</code>
CH_RVD	<code>eventInfo.getRegionViewId()</code>

Table 6–1 (Cont.) Click History Attribute Mapping

Click History Attributes	ADF UserActivityInfo fields
CH_RNM	eventInfo.getRegionViewName()
CH_RTY	eventInfo.getRendererType()
CH_TYP	eventInfo.getType()
CH_EID	userActivityInfo.getContextId()
CH_PEI	userActivityInfo.getParentContextId()
CH_TTT	Total Time = CH_CET - CH_CST
CH_RRT	userActivityInfo.getResponseReceivedTime()
CH_PRT	userActivityInfo.getParentResponseReceivedTime()
CH_RNT	Rendering Time = CH_CET - CH_RRT

Some of the viewing tools you can use to show the log file contents are:

- Enterprise Manager Cloud Control Support Workbench
You can use Enterprise Manager Cloud Control Support Workbench to view and sort the log information.
- WLST command
You can also use WLST command `displayLog` to view the record
- Any text editor such as `vi`.

For example, search all Click History records with a matching DSID in order to return a list of the recent "click" events of a HTTP session.

For example:

```
[2013-01-21T20:50:16.403-08:00] [ProcurementServer_1] [NOTIFICATION] []
[oracle.clickhistory.EUM] [tid: 44] [userId: FUSION]
[ecid: 004ox_GMXwy6aMGpIwslyF00018d000gvb,0:2:8:2:2]
[CLICK_COMPATYPE: oracle.adf.RichCommandButton] [APP: ProcurementApp#V2.0]
[DSID: 0000JlW_fz5Fw0HpIsO5yf1GzEoI000Dh3]
[CLICK_VIEWID: /adfp-portlet-bridge-container/container-view]
[CLICK_REGIONVIEWID: [/BrowseCategoryFlow/BrowseCategory,
/BrowseCategoryMainAreaFlow/BrowseCategoryMainArea,
/FSGenericTaskFlow/DynamicRegionContainer]]
[CLICK_STARTTIME: 1358830216656]
[CLICK_RENDERTYPE: oracle.adf.rich.Button]
[CLICK_TYPE: action]
[CLICK_COMPCLIENTID: _jpfpcncuivr__ns1588903364_j_id__
ctr0:fragmentRegionStatic:0:dynamicRegion1:0:pt1:Brows1:0:
ManageCategoryHierarchyPanel:ATT1:deleteWarningYES]
[CLICK_WINDOWID: k4coiwdvi_1]
[CLICK_REGIONVIEWNAME: [jpfpcncuivr__ns1588903364_j_id
_ctr0:fragmentRegionStatic:0:dynamicRegion1:0:pt1:Brows1, _jpfpcncuivr
_ns1588903364_j_id_ctr0:fragmentRegionStatic:0:dynamicRegion1, _jpfpcncuivr
_ns1588903364_j_id_ctr0:fragmentRegionStatic]]
[CLICK_COMPDISPLAYNAME: Yes]
```

6.2.7 How to Get Click History DMS Metrics

You can determine Click History's buffer utilization and capacity by inspecting the DMS metrics. You can execute a WLST command to display the metrics. If multiple QuickTrace handlers are configured, look for the handler with this name: `apps-clickhistory-handler`.

For example:

```
wls:/base_domain/serverConfig> displayMetricTables('QuickTraceInfo',
servers=['your_server'])
```

```
QuickTraceInfo
```

```
Host:    xyz.com
Name:    apps-clickhistory-handler
Parent:  /oracle/odl/quicktrace
Process: your_server:7009
ServerName: your_server
bufferElapsed.value: 134719 seconds
bufferRecordCount.value: 1579 records
bufferUsedPercentage.value: 100 percent
oldestTimestamp.value: 2012-06-26T18:38:48.261-0700 time
```

6.3 ADF Logs

Oracle Fusion Middleware provides log files that record events such as error messages, warning messages, server information, and HTTP requests. Most Fusion Middleware log files are in the Oracle Diagnostic Logging (ODL) format. You can use Fusion Middleware Control or WLST commands to search, view, and download log files. You can also use these tools to perform various tasks related to logging, such as configuring settings, changing file locations, setting log levels, setting file format, and configuring tracing.

For more information about Fusion Middleware log files, see the "Managing Log Files and Diagnostic Data" section of *Administering Oracle Fusion Middleware*.

JDeveloper also provides an ADF Logger and a log analyzer that you can use for monitoring and diagnostic purposes. JDeveloper generates the log file in the Oracle Diagnostic Logging (ODL) format that is used throughout Fusion Middleware. In JDeveloper, you use the Oracle Diagnostic Logging Configuration editor to set the log levels and other configuration parameters. Configuration information is stored in the `logging.xml` file.

After the log files have been created, you can use the Oracle Diagnostic Log Analyzer within JDeveloper to view the log entries. You can search and filter the log entries based on log levels, time frames, message content, and other parameters.

For more information about the ADF Loggers, see the "Using the ADF Logger" section of *Developing Fusion Web Applications with Oracle Application Development Framework*.

6.4 Diagnosing Problems using the Diagnostic Framework

Oracle Fusion Middleware provides a Diagnostic Framework to help you detect, diagnose, and resolve problems with your application. When a critical error occurs, the Diagnostic Framework immediately captures diagnostic data and associates the data

and error with an incident number. Using this number, you can retrieve the data for analysis from the Automatic Diagnostic Repository (ADR).

Oracle ADF provides an ADFConfig dump which will execute when an INCIDENT_ERROR message is logged. You can also add code to invoke the dump in the application exception handlers. [Example 6-2](#) show a sample code you can add to your exception handler to invoke the ADFConfig dump.

Example 6-2 Sample Code for Invoking ADFConfig Diagnostic Dump in Exception Handler

```
IllegalArgumentException e = new IllegalArgumentException("test exception");
LoggerFactory.getFrameworkLogger().log(ODLLevel.INCIDENT_ERROR,
    "Test error message", e);
```

For more information about the Diagnostic Framework, see *Administering Oracle Fusion Middleware*.

If you are using the Diagnostic Framework on an IBM WebSphere application server, you need to perform additional tasks. For more information, see the *Oracle Fusion Middleware Third-Party Application Server Guide*.

6.5 Finding ADF Runtime Version Information

Knowing the ADF runtime JARs version numbers can be important information for debugging and diagnosing application problems. This information may be requested by Oracle Support.

You can access My Oracle Support at <https://support.oracle.com>.

You can find which versions of the ADF Runtime JARs are in effect using several methods:

- Retrieving a CSV file that is generated when the application starts.
- Using shell scripts in offline mode.
- Using WLST commands.
- Using Fusion Middleware Control.
- Retrieving the dump files when an incident occurs.

6.5.1 How to Find Version Information by Starting the Application

When a deployed application is started, the runtime JARs version information is dumped into a CSV file located at:

```
<DomainName>/servers/<ServerName>/logs/<MyApp>-Versions.csv
```

Note the following:

- `oracle.adf.share.diagnostics.versions` logger controls generation of runtime JARs versions at application startup. JARs versions are generated when the logger is set to `level="FINEST"`.
- On new WebLogic domains, `oracle.adf.share.diagnostics.versions` logger will be available in the `logging.xml`.
- On existing domains, `oracle.adf.share.diagnostics.versions` logger will not be present in the `logging.xml`. Use Enterprise Manager Fusion Middleware Control or the WLST command to set the logger level.

6.5.2 How to Find Version Information by Using Shell Scripts in Offline Mode

In offline mode, you can use the `printJarVersions.sh` shell script to print the version information or to redirect the information to a file.

In the `%MW_HOME%/oracle_common/common/bin` directory, run the script:

```
bin$ ./printJarVersions.sh
```

You can redirect the output to a file:

```
bin$ ./printJarVersions.sh > /home/trdsouza/export-JarVersions.csv
```

The version information have a format similar to the following (trimmed):

```
Jar Path,Oracle-Version,Oracle-Label,Oracle-Builder,Oracle-BuildTimestamp,
Specification-Version,Implementation-Version/ade/userYada/oracle/oracle_
common/modules/oracle.adf.model_12.1.3/adfm.jar,12.1.3.0.40.66.68,
JDEVADF_12.1.2.0.0_GENERIC_130608.2330.6668,user=userY
host=abc123456,2013-10-02 04:55:45 -0700,,
```

6.5.3 How to Find Version Information by Using the WLST Scripting Console

In offline mode, you can use the WLST `exportJarVersions` command to find the version information:

```
exportJarVersions('/home/userY/export-JarVersions.csv')
```

In online mode, you can use the WLST `exportApplicationJarVersions` command to find the version information:

```
exportApplicationJarVersions('MyApp', '/home/userY/export-EarCrmCommon-Versions.csv')
```

If you want to find a specific JAR version, you can use the WLST `exportApplicationSelectedJarVersions` command:

```
exportApplicationSelectedJarVersions('MyApp', '/home/userY/export-EarCrmCommon-Versions.csv',
$ORACLE_HOME/modules/oracle.adf.model_12.1.3/adfm.jar;$ORACLE_
HOME/modules/oracle.adf.view_12.1.3/adf-richclient-impl-12.jar')
```

6.5.4 How to Find Version Information Using Fusion Middleware Control

You can use Enterprise Manager Fusion Middleware Control to monitor Oracle products that are built using ADF. You can also use this tool to find the version information of runtime JAR files.

For instructions, see [Section 3.2.2, "Finding Version Information of ADF Runtime JARs."](#)

6.5.5 How to Find Version Information by Inspecting Diagnostic Dump Files

When an `INCIDENT_ERROR` occurs when the application is running in a WebLogic Domain, the Oracle Diagnostic Framework generates dump files in the WebLogic Domain directories. The location of these files are in:

```
<DomainName>/servers/<ServerName>/adr/diag/ofm/<DomainName>/<ServerName>/incident/
```

One of the generated dump files contains the application JAR version information. It is called `adf_DiagnosticsJarsVersionDumpN.txt` where *N* is a numerical value.

For example,

```
<myDomainName>/servers/<myServerName>/adr/diag/ofm/<myDomainName>/<myServerName>/incident/adf_DiagnosticsJarsVersionDump45_i5.txt.
```

You can examine this file to view the version information. You can also upload this file to My Oracle Support for further analysis.

You can trigger a diagnostic dump from code by logging an INCIDENT_ERROR message.

6.6 Troubleshooting Oracle ADF for High Availability

Troubleshooting Fusion web applications that are deployed to high availability environments require additional concerns. For information about deployment issues in high availability environments, see "Troubleshooting High Availability Issues After Deployment" in *Developing Fusion Web Applications with Oracle Application Development Framework*. For information about Oracle ADF replication issues in high availability environments, see "Troubleshooting Oracle ADF Replication and Failover Issues" in *Developing Fusion Web Applications with Oracle Application Development Framework*.

WLST Command Reference for ADF Applications

This chapter describes the WLST commands you can use to deploy, manage, and configure Oracle ADF applications to Oracle WebLogic Server.

This chapter includes the following sections:

- [Section 7.1, "Overview of Custom WLST Commands for Oracle ADF"](#)
- [Section 7.2, "ADF-Specific WLST Commands"](#)

7.1 Overview of Custom WLST Commands for Oracle ADF

Use the ADF-based URL Connections WLST commands to navigate the hierarchy of configuration or runtime beans and control the prompt display. Use the `getADFMArchiveConfig` commands to manage the `ADFMArchiveConfig` object.

To use the custom WLST commands for Oracle ADF, you must invoke the WLST script from the Oracle Common home. For more information about other WLST commands, such as custom Metadata Services (MDS) commands, see the *WLST Command Reference for WebLogic Server*.

7.2 ADF-Specific WLST Commands

Use the commands in [Table 7-1](#) for ADF applications.

Table 7-1 Browse Commands for WLST Configuration

Use this command...	To...	Use with WLST...
<code>adf_createFileUrlConnection</code>	Create a new ADF file connection.	Online or Offline
<code>adf_createHttpUrlConnection</code>	Create a new ADF URL connection.	Online or Offline
<code>adf_setURLConnectionAttributes</code>	Set or edit the attributes of a newly created or existing ADF connection.	Online or Offline
<code>adf_listUrlConnection</code>	List a new URL connection.	Online or Offline
<code>getADFMArchiveConfig</code>	Return a handle to the <code>ADFMArchiveConfig</code> object for the specified archive.	Online or Offline

Table 7-1 (Cont.) Browse Commands for WLST Configuration

Use this command...	To...	Use with WLST...
exportJarVersions	Export CSV format of JARs versions from current ORACLE_HOME at a specified location.	Offline
exportApplicationJarVersions	Export CSV format of runtime JARs versions of a specified application at a specified location.	Online
exportApplicationSelectedJarVersions	Export CSV format of JARs versions of selected JARs at a specified location in coordination with the Versions.xml file.	Online
createWebServiceConnection	Create a Web service connection for an ADF application.	Online
listWebServiceConnection	List Web service connection for an ADF application.	Online
deleteWebServiceConnection	Delete a Web service connection for an ADF application.	Online

7.2.1 `adf_createFileURLConnection`

Use with WLST: Online or Offline.

7.2.1.1 Description

Use this command to creates a new connection based on the `oracle.adf.model.connection.url.FileURLConnection` connection class.

7.2.1.2 Syntax

```
adf_createFileURLConnection(appName, name, URL)
```

Argument	Definition
<code>appName</code>	Application name for which the connection will be created.
<code>name</code>	The name of the new connection.
<code>URL</code>	The URL associated with this connection.

7.2.1.3 Example

```
adf_createFileURLConnection('myapp', 'tempDir', '/scratch/tmp')
```

7.2.2 `adf_createHttpURLConnection`

Use with WLST: Online or Offline.

7.2.2.1 Description

Use this command to create a new connection based on the `oracle.adf.model.connection.url.HttpURLConnection` connection type class.

7.2.2.2 Syntax

```
adf_createURLConnection (appName, name, [URL], [authenticationType], [realm],
[user], [password])
```

Argument	Definition
<i>appName</i>	Application name for which the connection will be created.
<i>name</i>	The name of the new connection.
<i>url</i>	(Optional) The URL associated with this connection.
<i>authenticationType</i>	(Optional) The default is basic.
<i>realm</i>	(Optional) If this connection deals with authentication, then this should be set. The default is basic.
<i>user</i>	(Optional)
<i>password</i>	(Optional)

7.2.2.3 Example

```
adf_createURLConnection('myapp', 'cnn', 'http://www.cnn.com')
```

7.2.3 adf_setURLConnectionAttributes

Use with WLST: Online or Offline.

7.2.3.1 Description

Use this command to set or edit the attributes of a newly created or existing ADF connection.

7.2.3.2 Syntax

```
adf_setURLConnectionAttributes(appname, connectionname, attributes)
```

Argument	Definition
<i>appname</i>	Application name.
<i>connectionname</i>	The name of the connection.
<i>attributes</i>	The array containing attributes to set in key/value pairs.

7.2.3.3 Example

```
adf_setURLConnectionAttributes
('myapp', 'cnn', 'ChallengeAuthenticationType:digest',
'AuthenticationRealm:XMLRealm')
```

7.2.4 adf_listUrlConnection

Use with WLST: Online or Offline.

7.2.4.1 Description

Use this command to list the connections of the application.

7.2.4.2 Syntax

```
adf_listURLConnection (appname)
```

Argument	Definition
appname	Application name.

7.2.4.3 Example

```
adf_listURLConnection ('myapp')
```

7.2.5 getADFMArchiveConfig

Use with WLST: Online or Offline.

7.2.5.1 Description

Returns a handle to the `ADFMArchiveConfig` object for the specified archive. The returned `ADFMArchiveConfig` object's methods can be used to change application configuration in an archive.

The `ADFMArchiveConfig` object provides the following methods:

- `setDatabaseJboSQLBuilder([value])` - Sets the Database `jbo.SQLBuilder` attribute.
- `getDatabaseJboSQLBuilder()` - Returns the current value of the `jbo.SQLBuilder` attribute.
- `setDatabaseJboSQLBuilderClass([value])` - Sets the Database `jbo.SQLBuilderClass` attribute. Value is the full name of the custom builder class.
- `getDatabaseJboSQLBuilderClass()` - Returns the current value of the `jbo.SQLBuilderClass` attribute.
- `setDefaultRowLimit([value])` - Sets the defaults `rowLimit` attribute. Value is a long specifying the row limit (Default -1).
- `getDefaultRowLimit()` - Returns the current value of the `rowLimit` attribute.
- `save([toLocation])` - If you specify the `toLocation`, then the changes will be stored in the target archive file and the original file will remain unchanged. Otherwise, the changes will be saved in the original file itself.

7.2.5.2 Syntax

```
archiveConfigObject = ADFMAdmin.getADFMArchiveConfig(fromLocation)
```

Argument	Definition
<i>fromLocation</i>	The name of the ear file, including its complete path.

The syntax for `setDatabaseJboSQLBuilder([value])` is:

```
archiveConfigObject.setDatabaseJboSQLBuilder([value])
```

Argument	Definition
<i>value</i>	The value of the <code>jbo.SQLBuilder</code> attribute. Valid values are: 'Oracle' (Default), 'OLite', 'DB2', 'SQL92', 'SQLServer', or 'Custom. If 'Custom' is specified, then the <code>jbo.SQLBuilderClass</code> attribute should also be set.

The syntax for `getDatabaseJboSQLBuilder()` is:

```
archiveConfigObject.getDatabaseJboSQLBuilder()
```

The syntax for `setDatabaseJboSQLBuilderClass([value])` is:

```
archiveConfigObject.setDatabaseJboSQLBuilderClass([value])
```

Argument	Definition
<i>value</i>	The value of the <code>jbo.SQLBuilderClass</code> attribute.

The syntax for `getDatabaseJboSQLBuilderClass()` is:

```
archiveConfigObject.getDatabaseJboSQLBuilderClass()
```

The syntax for `setDefaultRowLimit([value])` is:

```
archiveConfigObject.setDefaultRowLimit([value])
```

Argument	Definition
<i>value</i>	The value of the <code>rowLimit</code> attribute.

The syntax for `getDefaultRowLimit()` is:

```
archiveConfigObject.getDefaultRowLimit([value])
```

The syntax for `save([toLocation])` is:

```
archiveConfigObject.save([toLocation])
```

Argument	Definition
<i>toLocation</i>	The file name along with the absolute path to store the changes.

7.2.5.3 Example

In the following example, the `jbo.SQLBuilder` attribute is set to 'DB2'.

```
wls:/offline> archive =
    ADFMAdmin.getADFMArchiveConfig(fromLocation='/tmp/testArchive.ear')
wls:/offline> archive.setDatabaseJboSQLBuilder(value='DB2')
wls:/offline> archive.save()
```

In the following example, the `jbo.SQLBuilder` attribute is removed so that application default is used.

```
wls:/offline> archive =
    ADFMAdmin.getADFMArchiveConfig(fromLocation='/tmp/testArchive.ear')
wls:/offline> archive.setDatabaseJboSQLBuilder()
wls:/offline> archive.save(toLocation='/tmp/targetArchive.ear')
```

In the following example, the `jbo.SQLBuilder` attribute is set to 'Custom', and the `jbo.SQLBuilderClass` attribute is set to the class 'com.example.CustomBuilder'.

```
wls:/offline> archive =
    ADFMAdmin.getADFMArchiveConfig(fromLocation='/tmp/testArchive.ear')
wls:/offline> archive.setDatabaseJboSQLBuilder('Custom')
wls:/offline> archive.setDatabaseJboSQLBuilderClass('com.example.CustomBuilder')
wls:/offline> archive.save(toLocation='/tmp/targetArchive.ear')
```

In the following example, the `rowLimit` attribute is set to 100.

```
wls:/offline> archive = getADFMArchiveConfig(fromLocation='/tmp/testArchive.ear')
wls:/offline> archive.setDefaultRowLimit(100)
wls:/offline> archive.save(toLocation='/tmp/targetArchive.ear')
```

7.2.6 exportJarVersions

Use with WLST: Offline.

7.2.6.1 Description

Use to export CSV format of JARs versions from current `ORACLE_HOME` at a specified location. Exported JARs versions information can be opened in Oracle OpenOffice or MicroSoft Excel.

7.2.6.2 Syntax

```
exportJarVersions(path)
```

Argument	Definition
path	Location to extract JARs versions.

7.2.6.3 Example

This example shows how JARs versions are exported to `/tmp/export-MyApp-Versions.csv`. R/W privileges for the CSV file need to be verified.

```
wls:/offline>exportJarVersions('/tmp/export-MyApp-Versions.csv')
```

7.2.7 exportApplicationJarVersions

Use with WLST: Online.

7.2.7.1 Description

Used to export CSV format of runtime JARs versions of a specified application at a specified location.

7.2.7.2 Syntax

```
exportApplicationJarVersions(applicationName, path)
```

Argument	Definition
applicationName	Application name to export JARs versions
path	Location to export JARs versions.

7.2.7.3 Example

This example shows how MyApp runtime JARs versions are exported to /tmp/export-MyApp-Versions.csv. R/W privileges for the CSV file need to be verified.

```
wls:/DefaultDomain/serverConfig>exportApplicationJarVersions('MyApp',
'/tmp/export-MyApp-Versions.csv')
```

7.2.8 exportApplicationSelectedJarVersions

Use with WLST: Online.

7.2.8.1 Description

Used to export CSV format of JARs versions of selected JARs at a specified location.

7.2.8.2 Syntax

```
exportApplicationSelectedJarVersions(applicationName, path, jarsLocation)
```

Argument	Definition
applicationName	Application name to export JARs versions.
path	Location to extract JARs versions.
jarsLocation	Optional list of selected JARs. If not specified, default JARs runtime version list from %WLSDOMAIN%/config/fmwconfig/Versions.xml will be exported. If the selectedJars property in Versions.xml is empty, version information of adfm.jar, adf-richclient-impl-11.jar, adf-controller.jar, adf-pageflow-impl.jar, adf-share-support.jar and mdsrt.jar will be exported.

7.2.8.3 Example

This example shows how JARs versions are exported to /tmp/export-MyApp-Versions.csv using the selectedJars property of the Versions.xml file. In this case, since the jarsLocation parameter is not specified, the libraries listed in the selectedJars property of the Versions.xml file will be exported. R/W privileges for the CSV file need to be verified.

```
wls:/offline>exportApplicationSelectedJarVersions('MyApp',
'/tmp/export-MyApp-Versions.csv')
```

```
Versions.xml
<Diagnostics xmlns="xmlns.oracle.com/adf/diagnostics">
  <Versions xmlns="xmlns.oracle.com/adf/diagnostics/versions"
    exportVersionsOnApplicationStartup="true"
    selectedJars="$ORACLE_HOME$/modules/oracle.adf.model_11.1.1/adfm.jar;
$ORACLE_HOME$/modules/oracle.adf.view_11.1.1/adf-richclient-impl-11.jar;
$ORACLE_HOME$/modules/oracle.adf.controller_11.1.1/adf-controller.jar;
$ORACLE_HOME$/modules/oracle.adf.pageflow_11.1.1/adf-pageflow-impl.jar;
$ORACLE_HOME$/modules/oracle.adf.share_11.1.1/adf-share-support.jar;
$ORACLE_HOME$/modules/oracle.mds_11.1.1/mdsrt.jar" />
</Diagnostics>
```

This example shows how JARs versions are exported to `/tmp/export-MyApp-Versions.csv` using the `jarsLocation` parameter. In this case, the libraries passed explicitly in the `jarsLocation` parameter will be exported. R/W privileges for the CSV file need to be verified.

```
wls:/offline>exportApplicationSelectedJarVersions('MyApp',
'/tmp/export-MyApp-Versions.csv',
'$ORACLE_HOME$/modules/oracle.adf.model_11.1.1/adfm.jar;$ORACLE_
HOME$/modules/oracle.adf.view_11.1.1/adf-richclient-impl-11.jar')
```

7.2.9 createWebServiceConnection

Use with WLST: Online.

7.2.9.1 Description

Used to create a Web Service connection for an ADF application.

Returns a set of service name and port names in the format {serviceName: List of portName}.

For example: {'PolicyReferenceEchoBeanService':
array(java.lang.String, ['PolicyReferenceEchoBeanPort'])}

7.2.9.2 Syntax

```
createWebServiceConnection(appName, wsConnName, wsdlUrlStr, readerProps)
```

Argument	Definition
appName	Name of the ADF application for which you want to create a Web service connection.
wsConnName	Name of the new Web service connection.
wsdlUrlStr	Name of the service WSDL URL string.
readerProps	The optional WSDL reader properties. For example: '["wsdl.reader.proxy.host=proxy.my.com", "wsdl.reader.proxy.port=80"]'

7.2.9.3 Example

This example shows how to create a Web service connection for an ADF application `myapp`, with a Web service connection `mywsconn`, and a WSDL URL string of `http://myserver/myservice?WSDL`.

```
createWebServiceConnection('myapp', 'mywsconn', 'http://myserver/myservice?WSDL')
```

7.2.10 listWebServiceConnection

Use with WLST: Online.

7.2.10.1 Description

Used to list the Web service connections associated with an ADF application.

7.2.10.2 Syntax

```
listWebServiceConnection(appName)
```

Argument	Definition
appName	Name of the ADF application for which you want to list its Web service connections.

7.2.10.3 Example

This example shows how list the Web service connections for the application myapp.

```
listWebServiceConnection('myapp')
```

7.2.11 deleteWebServiceConnection

Use with WLST: Online.

7.2.11.1 Description

Used to delete a Web service connection associated with an ADF application.

7.2.11.2 Syntax

```
listWebServiceConnection(appName, weConnName)
```

Argument	Definition
appName	Name of the ADF application for which you want to delete a Web service connection.
wsConnName	Name of the Web service connection you want to delete.

7.2.11.3 Example

This example shows how delete the Web service connection mywsconn from the application myapp.

```
deleteWebServiceConnection('myapp', 'mywsconn')
```


Part III

Appendices

Part III contains the following chapters:

- [Appendix A, "Configuring GlassFish Server"](#)
- [Appendix B, "ADF Runtime Libraries"](#)
- [Appendix C, "Audit Reference for Oracle Application Development Framework"](#)

Configuring GlassFish Server

This appendix describes how to configure GlassFish Server for Oracle ADF Essentials. It describes how to obtain the Oracle ADF Runtime and how to install these files into the GlassFish Server

This appendix contains the following sections:

- [Section A.1, "About Configuring GlassFish"](#)
- [Section A.2, "Obtaining GlassFish Server and Oracle ADF Runtime"](#)
- [Section A.3, "Configuring GlassFish with ADF Runtime Libraries"](#)
- [Section A.4, "Additional Configuration Tasks"](#)
- [Section A.5, "Deploying an ADF Application to GlassFish"](#)

A.1 About Configuring GlassFish

GlassFish Server is an application server that can be configured to run Oracle ADF applications. If you do not have a GlassFish Server installation, you can download GlassFish Server from the GlassFish website. Before you can run ADF applications in a GlassFish Server, you need to configure GlassFish with the Oracle ADF Runtime libraries.

For a list of the supported Oracle ADF features for GlassFish, go to the OTN site at <http://www.oracle.com/technetwork/developer-tools/adf/overview/adfessentials-1719844.html>

For instructions on obtaining and installing GlassFish, see <http://glassfish.java.net/downloads/3.1.2-final.html>

For information about developing ADF applications for GlassFish, see the "Deploying ADF Applications to GlassFish" appendix in the *Developing Fusion Web Applications with Oracle Application Development Framework*.

A.2 Obtaining GlassFish Server and Oracle ADF Runtime

Oracle ADF Essentials supports the Open Source and commercial versions of GlassFish Server. With either version, you will need the Full Platform distribution. The Web Profile distribution is not supported. After you have installed the GlassFish Server, you need to obtain the ADF Essentials `adf-essentials.zip` file from OTN and follow the instructions in this appendix to install the ADF Runtime libraries.

For information on how to start and stop the server and other application server tasks, see GlassFish documentation at the GlassFish website.

You will need to perform the following tasks, as described in the following topics:

- [Section A.2.1, "Obtaining GlassFish Server"](#)
- [Section A.2.2, "Obtaining Oracle ADF Runtime"](#)

A.2.1 Obtaining GlassFish Server

You can download the open source version of the GlassFish Server from the GlassFish website:

<http://glassfish.java.net/downloads/3.1.2-final.html>

Follow the instructions and documentation at the GlassFish site to install and configure a GlassFish Server.

A.2.2 Obtaining Oracle ADF Runtime

In order for a GlassFish Server to run Oracle ADF applications, you must install the ADF Runtime library files into the GlassFish installation directory.

You can download the Oracle ADF Essentials `adf-essentials.zip` file from the Oracle Technology Network at

<http://www.oracle.com/go/?&Src=7578263&Act=4&pcode=WWMK12022244MPP044>.

After you have downloaded the `adf-essentials.zip` file, you can extract the files to a flat-structured temporary directory which you can use to copy the required files into the GlassFish installation directories.

For instance, if you are using `unzip`, you can add the `-j` option to create a flat directory structure that has no hierarchical folders.

```
unzip -j <file> -d <destination>
```

A.3 Configuring GlassFish with ADF Runtime Libraries

The ADF Runtime libraries consists of the following:

- ADF Share libraries
- ADF Model libraries
- ADF Controller libraries
- ADF View libraries

The ADF Share libraries must be manually installed into the GlassFish installation. You use your operating system commands or tools to copy the files into GlassFish. For instructions, see [Section A.3.1](#)

The ADF Model, ADF Controller, and ADF View libraries are loaded into GlassFish with the deployed application. When you use JDeveloper to develop the application, you will reference the required libraries before you package the application into an EAR file for deployment.

The steps for installing the ADF Runtime libraries are:

1. Configure ADF Share libraries.
 - Copy the ADF Share libraries from `adf-essentials.zip` into the GlassFish installation.
2. Deploy the application as an EAR file to GlassFish Server.

A.3.1 Installing ADF Share Libraries Manually

It may be helpful to have an understanding of the options that are available to you when you are mapping ADF Share for GlassFish. For more information, see [Section A.3](#)

You will need to complete these tasks:

- Install the GlassFish Server
- Obtain the `adf-essentials.zip` file and unzip it to a temporary directory

To install the ADF Share Runtime libraries:

1. Copy or move the ADF Share library files from the temporary directory to the `<glassfish>/domains/domain1/lib/applibs` folder:

The ADF Share files should be in the temporary directory where you had unzip the `adf-essentials.zip` file as described in [Section A.2.2](#) This directory should be `<temp>/oracle_common/modules`.

You must copy the following JAR files into the `<glassfish>/domains/domain1/lib/applibs` directory:

```

oracle.adf.share.ca_12.1.3/adf-share-base.jar
oracle.adf.share.ca_12.1.3/adf-share-ca.jar
oracle.adf.share_12.1.3/commons-el.jar
oracle.adf.share_12.1.3/adf-share-support.jar
oracle.adf.share_12.1.3/adfsharebean.jar
oracle.adf.share_12.1.3/jsp-el-api.jar
oracle.adf.share_12.1.3/adflogginghandler.jar
oracle.adf.share_12.1.3/oracle-el.jar
oracle.mds_12.1.3/mdsrt.jar
oracle.bali.share_12.1.3/share.jar
oracle.xml_12.1.3/xml.jar
oracle.javatools_12.1.3/resourcebundle.jar
oracle.javatools_12.1.3/javamodel-rt.jar
oracle.javatools_12.1.3/javatools-nodeps.jar
oracle.javatools_12.1.3/oicons.jar
oracle.adf.security_12.1.3/adf-share-security.jar
oracle.adf.security_12.1.3/adf-controller-security.jar
oracle.xdk_12.1.3/xmlparserv2_sans_jaxp_services.jar
oracle.xdb_11.1.0.jar
oracle.jrf_12.1.3/jrf-api.jar
oracle.jdbc_11.2/ojdbc6dms.jar
oracle.dms_12.1.3/dms.jar
oracle.odl_12.1.3/ojdl.jar
oracle.odl_12.1.3/ojdl2.jar
oracle.classloader_11.1.1.jar
oracle.logging-utils_11.1.1.jar
oracle.web-common_12.1.3.jar
org.apache.bcel_5.1.jar
oracle.nlsrtl_11.2.0/orai18n-mapping.jar
org.apache.commons.beanutils_1.8.3
org.apache.commons.logging_1.1.1
oracle.jsp_12.1.3/globaltdcache
oracle.mds_12.1.3/mdsrt
com.oracle.webservices.fmw.oc4j-ws-support-impl_12.1.3
oracle.http_client_12.1.3
oracle.web-common_12.1.3
oracle.web-common-schemas_12.1.3

```

2. Verify the libraries.

A.4 Additional Configuration Tasks

After you have installed the ADF Runtime into the GlassFish Server, you need to perform additional configuration tasks on GlassFish.

The configuration tasks are:

- [Section A.4.1, "Using the GlassFish Administration Console"](#)
- [Section A.4.2, "Creating a Datasource for GlassFish"](#)
- [Section A.4.3, "Configuring the JVM Cache"](#)

A.4.1 Using the GlassFish Administration Console

You can use the GlassFish Administration Console to configure the GlassFish Server including managing applications, JDBC pools, and other resources. If you are using GlassFish 3.1.2, the secure console is disabled by default. You would need to enable secure console to access the admin page remotely or only access the page from the same machine where the GlassFish server is running.

The GlassFish Administration Console is at:

```
http://<machine_name>:4848/
```

A.4.2 Creating a Datasource for GlassFish

It may be helpful to have an understanding of the options that are available to you when you are creating a datasource for GlassFish. For more information, see [Section A.4](#).

You can use the GlassFish console or use `asadmin` commands to create the datasource. For using the GlassFish console to create a datasource, see GlassFish documentation.

To create a datasource for GlassFish using `asadmin` commands:

1. Open a command line window.
2. Invoke the `asadmin` command to create a datasource for GlassFish.

For instance, the following command creates a datasource for an application:

```
asadmin> create-jdbc-connection-pool --datasourceclassname
oracle.jdbc.pool.SampleDataSource
--restype javax.sql.XADataSource
--property user=xyz
:password=xyz:url=jdbc:oracle:thin:@machine.com:1521:machine
--target=server SampleDSPool

asadmin> ping-connection-pool sampleDSPool

asadmin> create-jdbc-resource --connectionpoolid SampleDSPool jdbc/OracleDS
```

A.4.3 Configuring the JVM Cache

You need to configure the JVM settings to `simple` and increase the memory size from 192 Mb to 512 Mb.

It may be helpful to have an understanding of the options that are available to you when you are configuring JVM for GlassFish. For more information, see [Section A.4](#)

To configure JVM Cache for MDS:

1. Start the GlassFish Administration Console.
2. Choose **Configurations > server-config > JVM Settings**.
3. Select **JVM Options** and specify `-Doracle.mds.cache=simple` and `XX:MaxPermSize=512m`.
4. Click **Save**.
5. Or, open the `<glassfish>/domains/domain1/config/domain.xml` file and edit the following entries:

```
<jvm-options>-XX:MaxPermSize=512m</jvm-options>  
<jvm-options>-Doracle.mds.cache=simple</jvm-options>
```

A.5 Deploying an ADF Application to GlassFish

After you have referenced the libraries in the application, you can proceed to deploy the application to GlassFish Server. For more information, see the "Deploying ADF Applications to GlassFish" appendix in the *Developing Fusion Web Applications with Oracle Application Development Framework*.

B

ADF Runtime Libraries

This appendix describes the contents of ADF runtime libraries (`adf.oracle.domain.webapp.war`, `adf.oracle.domain.ear`, `adf.desktopintegration.war`, `adf.desktopintegration.model.ear`, and `system classpath`) deployed into Oracle WebLogic Server to support ADF applications.

The following ADF runtime libraries are described:

- [Section B.1, "Using JDeveloper to Find the ADF Runtime Library"](#)
- [Section B.2, "adf.oracle.domain.webapp.war Library"](#)
- [Section B.3, "adf.oracle.domain.ear Library"](#)
- [Section B.4, "adf.desktopintegration.war Library"](#)
- [Section B.5, "adf.desktopintegration.model.ear Library"](#)
- [Section B.6, "System Classpath"](#)

B.1 Using JDeveloper to Find the ADF Runtime Library

In addition to the listings in this appendix, you can also use JDeveloper to find a JAR's corresponding ADF runtime library.

To find the JDeveloper library for a JAR:

1. In JDeveloper, select **Tools > Manage Libraries**.
2. In the Manage Libraries dialog Libraries tab, click the **Search** icon and select **Jar name** from the dropdown list.
3. In the search field, enter the name of the JAR and click the search icon.

B.2 adf.oracle.domain.webapp.war Library

[Table B-1](#) lists the JAR files that are packaged into the `adf.oracle.domain.webapp.war` file and their corresponding ADF runtime library.

Table B-1 *adf.oracle.domain.webapp.war Library*

JAR	ADF Library
<code>oracle.facesconfigdt_12.1.3/taglib.jar</code>	ADF Faces Change Manager Runtime 11
<code>oracle.facesconfigdt_12.1.3/facesconfigmodel.jar</code>	ADF Faces Change Manager Runtime 11
<code>org.apache.http.components.httpmime-4.1.2.jar</code>	NA

Table B-1 (Cont.) adf.oracle.domain.webapp.war Library

JAR	ADF Library
oracle.adf.view_12.1.3/batik-anim.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/dvt-trinidad.jar	ADF DVT Core Runtime ADF DVT Faces Runtime
oracle.adf.view_12.1.3/batik-transcoder.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/adf-faces-changemanager-rt.jar	ADF Faces Change Manager Runtime 11
oracle.adf.view_12.1.3/dvt-databinding-dt-core.jar	ADF Designtime API
oracle.adf.view_12.1.3/dvt-jclient-adf.jar	Oracle BI Graph ADF Swing Runtime
oracle.adf.view_12.1.3/adf-faces-templating-dt-core.jar	ADF Designtime API
oracle.adf.view_12.1.3/adf-richclient-api-11.jar	Trinidad Databinding Runtime ADF Faces Runtime 11
oracle.adf.view_12.1.3/dvt-shared-js.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/adf-faces-skin-dt-core.jar	NA
oracle.adf.view_12.1.3/inspect4.jar	Oracle JJWT
oracle.adf.view_12.1.3/batik-swing.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/batik-codec.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/adf-faces-databinding-rt.jar	Trinidad Databinding Runtime ADF Faces Databinding Runtime
oracle.adf.view_12.1.3/prefuse.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/adf-dt-at-rt.jar	ADF Model Runtime ADF Designtime API
oracle.adf.view_12.1.3/batik-script.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/jewt4.jar	BC4J Tester Oracle Help for Java Oracle JJWT
oracle.adf.view_12.1.3/adf-view-databinding-dt-core.jar	ADF Designtime API
oracle.adf.view_12.1.3/adf-richclient-impl-11.jar	ADF Faces Runtime 11
oracle.adf.view_12.1.3/dvt-faces.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/dvt-utils.jar	Oracle BI Graph ADF DVT Core Runtime ADF Swing Runtime ADF DVT Faces Runtime
oracle.adf.view_12.1.3/trinidad-impl.jar	ADF Faces Runtime 11 Trinidad Runtime 11
oracle.adf.view_12.1.3/dvt-databindings.jar	ADF DVT Faces Databinding Runtime

Table B-1 (Cont.) adf.oracle.domain.webapp.war Library

JAR	ADF Library
oracle.adf.view_12.1.3/trinidad-api.jar	ADF Faces Runtime 11 Trinidad Runtime 11
oracle.adf.view_12.1.3/batik-xml.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/batik-css.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/batik-dom.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/dvt-as.jar	NA
oracle.adf.view_12.1.3/batik-parser.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/batik-util.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/batik-bridge.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/batik-ext.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/adf-share-web.jar	NA
oracle.adf.view_12.1.3/dvt-basemaps.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/adf-faces-templating-dtrt.jar	ADF Designtime API
oracle.adf.view_12.1.3/batik-gui-util.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/adf-dt-at-rt-wizards.jar	ADF Runtime Wizards
oracle.adf.view_12.1.3/xml-apis-ext.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/adf.constants.jar	NA
oracle.adf.view_12.1.3/batik-svggen.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/batik-svg-dom.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/adf-dynamic-faces.jar	ADF Faces Dynamic Components
oracle.adf.view_12.1.3/batik-extension.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/batik-awt-util.jar	ADF DVT Faces Runtime
oracle.adf.view_12.1.3/dvt-facesbindings.jar	ADF DVT Faces Databinding Runtime
oracle.adf.view_12.1.3/dvt-jclient.jar	Oracle BI Graph ADF DVT Core Runtime ADF Swing Runtime ADF DVT Faces Runtime
oracle.adf.view_12.1.3/bundlresolver.jar	Resource Bundle Variable Resolver
oracle.adf.view_12.1.3/adf-view-unified-databinding-dt-core.jar	ADF Designtime API
oracle.adf.view_12.1.3/adf-richclient-jmx.jar	ADF Faces JMX Runtime 11
oracle.adf.view_12.1.3/adf-faces-databinding-dt-core.jar	ADF Designtime API
oracle.adf.view_12.1.3/adf-richclient-automation-11.jar	Oracle Extended Selenium
oracle.adf.view_12.1.3/batik-gvt.jar	ADF DVT Faces Runtime

Table B–1 (Cont.) adf.oracle.domain.webapp.war Library

JAR	ADF Library
oracle.adf.view_12.1.3/adf-customizationset-ui.jar	NA
oracle.adf.view_12.1.3/adf-customizationset-mig.jar	NA
oracle.xdk_12.1.3/xml.jar	MDS Runtime Dependencies Oracle XML Parser v2 XSQL Runtime
oracle.adf.controller_12.1.3/adf-controller.jar	ADF Controller Runtime
oracle.adf.controller_12.1.3/adf-controller-rt-common.jar	ADF Controller Runtime
oracle.adf.controller_12.1.3/adf-controller-api.jar	ADF Controller Runtime
oracle.adf.share.ca_12.1.3/adf-share-ca.jar	MDS Runtime Dependencies ADF Model Generic Runtime BC4J Runtime BC4J Security ADF Common Runtime
velocity-dep-1.4.jar	ADF Designtime API
org.apache.http.components.httpclient-4.1.2.jar	NA
oracle.adf.pageflow_12.1.3/adf-pageflow-impl.jar	ADF Page Flow Runtime
oracle.adf.pageflow_12.1.3/adf-pageflow-rc.jar	ADF Page Flow Runtime
oracle.adf.pageflow_12.1.3/adf-pageflow-fwk.jar	ADF Page Flow Runtime
oracle.adf.pageflow_12.1.3/adf-pageflow-dtrt.jar	ADF Page Flow Runtime ADF Designtime API
org.apache.http.components.httpclient-cache-4.1.2.jar	NA

B.3 adf.oracle.domain.ear Library

Table B–2 lists the JAR files that are packaged into the `adf.oracle.domain.ear` file and their corresponding ADF runtime library.

Table B–2 adf.oracle.domain.ear Library

JAR	ADF Library
oracle.adf.model_12.1.3/ordim.jar	Oracle Intermedia ADF Swing Oracle Intermedia
oracle.adf.model_12.1.3/adflibfilter.jar	ADF Common Web Runtime
oracle.adf.model_12.1.3/adf-faces-registration.jar	NA
oracle.adf.model_12.1.3/bc4jhtml.jar	BC4J Struts Runtime

Table B–2 (Cont.) adf.oracle.domain.ear Library

JAR	ADF Library
oracle.adf.model_12.1.3/adfbcsvc-share.jar	BC4J Service Runtime BC4J Service Client BC4J System Catalog
oracle.adf.model_12.1.3/adfimportlet.jar	NA
oracle.adf.model_12.1.3/jmxdc.jar	JMX Data Control
oracle.adf.model_12.1.3/jr_dav.jar	Resource Catalog Service
oracle.adf.model_12.1.3/bc4j-mbeans.jar	BC4J Runtime
10 oracle.adf.model_12.1.3/bc4jimdomains.jar	Oracle Intermedia ADF Swing Oracle Intermedia
oracle.adf.model_12.1.3/adfm.jar	BC4J EJB Client ADF Model Runtime BC4J Oracle Domains ADF Model Generic Runtime BC4J Runtime ADF Swing Runtime SR-227 API BC4J EJB Runtime BC4J Client BC4J IAS Client
oracle.adf.model_12.1.3/rcs-adflib-rt.jar	NA
oracle.adf.model_12.1.3/adfbcsvc.jar	BC4J Service Runtime
oracle.adf.model_12.1.3/ordhttp.jar	Oracle Intermedia ADF Swing Oracle Intermedia
15 oracle.adf.model_12.1.3/jdev-cm.jar	BC4J EJB Client ADF Model Runtime BC4J Tester BC4J Runtime BC4J Client BC4J IAS Client Connection Manager BC4J Recorder
oracle.adf.model_12.1.3/dvt-databindings-mds.jar	ADF DVT Faces Databinding MDS Runtime
oracle.adf.model_12.1.3/adflibrary.jar	ADF Model Runtime ADFm Designtime API
oracle.adf.model_12.1.3/adftransactionsdt.jar	ADF Model Runtime ADFm Designtime API ADF Designtime API;
oracle.adf.model_12.1.3/regexp.jar	BC4J Tester BC4J Recordere

Table B–2 (Cont.) adf.oracle.domain.ear Library

JAR	ADF Library
20 oracle.adf.model_12.1.3/adf-sec-idm-dc.jar	User and Role Data Control
oracle.adf.model_12.1.3/adfmweb.jar	ADF Web Runtime
oracle.adf.model_12.1.3/mds-dc.jar	NA
oracle.adf.model_12.1.3/adfbcsvc-client.jar	BC4J Service Client
oracle.adf.model_12.1.3/adf-controller-schema.jar	ADF Controller Schema ADF Controller Schema
25 oracle.adf.model_12.1.3/adfdt_common.jar	ADF Model Runtime ADFm Designtime API
oracle.adf.model_12.1.3/db-ca.jar	BC4J EJB Client ADF Model Runtime BC4J Tester BC4J Runtime BC4J Client BC4J IAS Client BC4J Recorder DB Runtime (db-tests)
oracle.adf.model_12.1.3/rcsrt.jar	Resource Catalog Service
oracle.adf.model_12.1.3/adfm-sqldc.jar	ADF SQL Data Control Runtime
oracle.adf.model_12.1.3/adfm-debugger.jar	BC4J Tester
30 oracle.adf.model_12.1.3/bc4jsyscat.jar	BC4J System Catalog
oracle.adf.model_12.1.3/adfbcsvc-registration.jar	Kava SDO
oracle.adf.model_12.1.3/adftags.jar	Oracle ADF DataTag
oracle.adf.model_12.1.3/datatags.jar	NA
oracle.adf.businesseditor_ 12.1.3/adf-businesseditor-model.jar	ADFm Business Editor Runtime
35 oracle.adf.businesseditor_ 12.1.3/adf-businesseditor-objects.jar	ADFm Business Editor Runtime
oracle.adf.businesseditor_ 12.1.3/adf-businesseditor-settings.jar	ADFm Business Editor Settings ADFm Business Editor Runtime
groovy-all-2.0.5.jar	ADF Model Runtime;ADF Model Generic Runtime;BC4J Runtime
oracle.xdk_12.1.3/oraclexsql.jar	NA
oracle.xdk_12.1.3/xsqlserializers.jar	XSQL Runtime

B.4 adf.desktopintegration.war Library

Table B–3 lists the JAR files that are packaged into the adf.desktopintegration.war file and their corresponding ADF runtime library.

Table B-3 *adf.desktopintegration.war Library*

JAR	ADF Library
oracle.adf.desktopintegration_12.1.3/adf-desktop-integration.jar	ADF Desktop Integration Runtime
oracle.adf.desktopintegration_12.1.3/adf-desktop-integration-admin-tool.jar	NA

B.5 adf.desktopintegration.model.ear Library

Table B-4 lists the JAR files that are packaged into the `adf.desktopintegration.model.ear` file and their corresponding ADF runtime library.

Table B-4 *adf.desktopintegration.model.ear Library*

JAR	ADF Library
oracle.adf.desktopintegration.model_12.1.3/adf-desktop-integration-model-api.jar	ADF Desktop Integration Model API

B.6 System Classpath

Table B-5 lists the JAR files that are loaded into the system classpath and their corresponding ADF runtime library.

Table B-5 *System Classpath*

JAR	ADF Library
oracle.javatools_12.1.3/javatools-jndi-local.jar	NA
oracle.javatools_12.1.3/javamodel-rt.jar	JAX-RPC Client
oracle.javatools_12.1.3/javatools-nodeps.jar	ADF Common Web Runtime MDS Runtime Dependencies ADFm Desgntime API JAX-RPC Client DB Runtime (db-tests)
oracle.javatools_12.1.3/oicons.jar	ADFm Designtime API
oracle.javatools_12.1.3/resourcebundle.jar	ADF Desktop Integration Runtime Resource Bundle Support BC4J Runtime
oracle.adf.security_12.1.3/adf-share-security.jar	ADF Model Runtime BC4J Security ADF Common Runtime
oracle.adf.security_12.1.3/adf-controller-security.jar	ADF Model Runtime ADF Common Runtime

Table B-5 (Cont.) System Classpath

JAR	ADF Library
oracle.adf.share.ca_12.1.3/adf-share-ca.jar	MDS Runtime Dependencies ADF Model Generic Runtime BC4J Runtime BC4J Security ADF Common Runtime
oracle.adf.share.ca_12.1.3/adf-share-base.jar	ADF Common Web Runtime MDS Runtime Dependencies ADF Model Generic Runtime BC4J Runtime ADF Swing Runtime BC4J Security ADF Common Runtime
features/adf.security_12.1.3.jar	NA
features/adf.share_12.1.3.jar	NA
features/adf.share.ca_12.1.3.jar	NA
oracle.mds_12.1.3/oramds.jar	MDS Runtime Dependencies
oracle.mds_12.1.3/mdsrt.jar	MDS Runtime
oracle.mds_12.1.3/mdslcm-client.jar	NA
oracle.mds_12.1.3/mdslcm.jar	NA
oracle.xmldef_12.1.3/xmldef.jar	MDS Runtime Dependencies ADF Faces Change Manager Runtime 11 ADF Model Generic Runtime
oracle.adf.share_12.1.3/jsp-el-api.jar	ADF Model Runtime MDS Runtime Dependencies ADF Model Generic Runtime BC4J Runtime
oracle.adf.share_12.1.3/adf-share-mbeans-wlst.jar	NA
oracle.adf.share_12.1.3/adfflogginghandler.jar	MDS Runtime Dependencies BC4J Tester ADF Model Generic Runtime BC4J Runtime ADF Common Runtime

Table B-5 (Cont.) System Classpath

JAR	ADF Library
oracle.adf.share_12.1.3/adf-share-wls.jar	NA
oracle.adf.share_12.1.3/oracle-el.jar	ADF Model Runtime MDS Runtime Dependencies ADF Model Generic Runtime BC4J Runtime
oracle.adf.share_12.1.3/adf-share-support.jar	MDS Runtime Dependencies ADF Model Generic Runtime BC4J Runtime BC4J Security ADF Common Runtime
oracle.adf.share_12.1.3/adfsharebean.jar	BC4J Runtime ADF Common Runtime
oracle.adf.share_12.1.3/commons-el.jar	ADF Model Runtime MDS Runtime Dependencies ADF Model Generic Runtime BC4J Runtime
oracle.adf.share_12.1.3/adfscripting.jar	NA
oracle.ons_12.1.3/ons.jar	NA
oracle.bali.share_12.1.3/share.jar	MDS Runtime Dependencies BC4J Tester ADF Model Generic Runtime Oracle Help for Java Oracle JWT

Audit Reference for Oracle Application Development Framework

This appendix provides reference information for auditing in Oracle Application Development Framework.

This appendix contains these sections:

- [Section C.1, "About Custom and Standard Audit Reports"](#)
- [Section C.2, "Attributes of ADF Audit Events"](#)
- [Section C.3, "Audit Events in Oracle ADF View"](#)
- [Section C.4, "Audit Events in Oracle ADF Business Components"](#)
- [Section C.5, "Audit Events in Oracle ADF Model"](#)
- [Section C.6, "Audit Events in Oracle ADF Controller"](#)
- [Section C.7, "Audit Events in Jedi"](#)

C.1 About Custom and Standard Audit Reports

The Common Audit Framework in Oracle Fusion Middleware provides a set of standard reports based on your audit records. It also enables you to modify the standard reports and create your own custom audit reports.

This appendix provides details about events that can be audited in Oracle Application Development Framework. Use this information to understand the structure of each event record to develop custom reports.

The following documents provide more information to help you write custom reports:

- "The Audit Schema" in the *Securing Applications with Oracle Platform Security Services*

The following documents provide additional information about how to configure auditing and view standard reports:

- "Configuring and Managing Auditing" in the *Securing Applications with Oracle Platform Security Services*

C.2 Attributes of ADF Audit Events

ADF audit events use the attributes specified in [Table C-1](#)

Table C-1 *Attributes for ADF Audit Events*

Attribute Name	Description
mds-mo-name	Metadata object name.
mds-mo-type	Metadata object type.
mds-cust-layer-name	Customization layer name.
mds-cust-layer-value	Customization layer value.
mds-sandbox-name	Sandbox name.
custom-operation	The operation that triggered this event.
custom-attribute1-name	The name of custom attribute 1.
custom-attribute1-value	The value of custom attribute 1.
custom-attribute1-old-value	The old value of custom attribute 1.
custom-attribute2-name	The name of custom attribute 2.
custom-attribute2-value	The value of custom attribute 2.
custom-attribute2-old-value	The old value of custom attribute 2.
custom-attribute3-name	The name of custom attribute 3.
custom-attribute3-value	The value of custom attribute 3.
custom-attribute3-old-value	The old value of custom attribute 3.
custom-attribute4-name	The name of custom attribute 4.
custom-attribute4-value	The value of custom attribute 4.
custom-attribute4-old-value	The old value of custom attribute 4.
custom-attribute5-name	The name of custom attribute 5.
custom-attribute5-value	The value of custom attribute 5.
custom-attribute5-old-value	The old value of custom attribute 5.
custom-sub-object-name	User-recognizable name for the sub-object (XML element/attribute) that changed.
custom-sub-object-type	User-friendly type for the sub-object (XML element/attribute) that changed.
mds-repository-name	MDS repository name.
mds-partition-name	MDS repository partition name.
custom-object-name	User-friendly object name.
custom-object-type	User-recognizable object type.

C.3 Audit Events in Oracle ADF View

Table C-2 lists the audit events for ADF View:

Table C–2 Oracle ADF View Audit Events

Event	Description
CreatePage	Create a new page.
CreateRegion	Add a region to a page.
CreateDataboundComponent	Add a databound component to a page.

C.4 Audit Events in Oracle ADF Business Components

Table C–3 lists the audit events for ADF Business Components:

Table C–3 Oracle ADF Business Components Audit Events

Event Type	Event	Description
View Object	AddViewLinkDef	Add a view link definition.
	RemoveViewLinkDef	Remove a view link definition.
	AddAttributeDef	Add an attribute.
Entity Object	AddAttributeDef	Add an attribute definition.
	AddTrigger	Add a trigger.
	AddValidator	Add a validator.
	RemoveTrigger	Remove a trigger.
Application Module	RemoveValidator	Remove a validator.
	CreateViewObject	Create a view object.
	RemoveViewObject	Remove a view object.

C.5 Audit Events in Oracle ADF Model

Table C–4 lists the audit events for ADF Model:

Table C–4 Oracle ADF Model Audit Events

Event Type	Event	Description
Page Definition	AddControBinding	Add a control binding.
	AddIterator	Add an iterator binding.
	AddVariable	Add a variable binding.
	AddExecutable	Add an executable binding.
	AddParameter	Add a parameter binding.
	AddNestedContainer	Add a nested container binding.
	RemoveExecutable	Remove an executable binding.
	RemoveIterator	Remove an iterator binding.
Application	RemoveParameter	Remove a parameter binding.
	AddDataControlReference	Add data control reference.
	AddPageDefinitionUsage	Add a page definition usage.
	AddPageMapEntry	Add a page map entry.
	RemoveDataControl	Remove data control usage.

Table C-4 (Cont.) Oracle ADF Model Audit Events

Event Type	Event	Description
Configuration	RemovePageDefinitionUsage	Remove page definition usage.
	RemovePageMapEntry	Remove a page map entry.
	AddDataControlDefinition	Add a data control definition.
	RemoveDataControlDefinition	Remove a data control definition.

C.6 Audit Events in Oracle ADF Controller

Table C-5 lists the audit events for ADF Controller:

Table C-5 Oracle ADF Controller Audit Events

Event	Description
setPageName	Set a page name.
addControlFlowCase	Add a control flow case.
addActivity	Insert a new activity into a flow.
getMutableTaskFlowDefinition	Edit a task flow definition.
addEntry	Add a resource entry.
updateEntry	Update a resource entry.
deleteEntry	Delete a resource entry.

C.7 Audit Events in Jedi

Table C-6 lists the audit events for Jedi:

Table C-6 Jedi Audit Events

Event	Description
AddAttribute	Add an attribute.
AddOperation	Add an operation.
AddValidationRule	Add a validation rule.
DeleteOperation	Delete an operation.
EditAttribute	Edit an attribute.
RemoveAttribute	Remove an attribute.
NewCustomObject	Add new custom object.
ChangedProperty	Change a property.
ChangedLabel	Display label change.
ChangedRequired	Required change.
ChangedUpdateable	Updatable change.