Professional Expertise Distilled

Oracle Fusion Applications Administration Essentials

Administer, configure, and maintain your Oracle Fusion Applications

Faisal Ghadially
Kalpit Parikh

[PACKT] PUBLISHING enterprise
professional expertise distilled

Oracle Fusion Applications Administration Essentials

Administer, configure, and maintain your Oracle Fusion Applications

Faisal Ghadially

Kalpit Parikh

[PACKT] PUBLISHING

enterprise

professional expertise distilled

BIRMINGHAM - MUMBAI

Oracle Fusion Applications Administration Essentials

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: December 2013

Production Reference: 1181213

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-84968-686-0

www.packtpub.com

Cover Image by Ravaji Babu (ravaji_babu@outlook.com)

Credits

Authors

Faisal Ghadially

Kalpit Parikh

Reviewers

Saumitra Chattopadhyay

Tim Warner

Acquisition Editors

Saleem Ahmed

Nikhil Karkal

Commissioning Editor

Neil Alexander

Technical Editors

Iram Malik

Aman Preet Singh

Copy Editors

Alisha Aranha

Mradula Hegde

Dipti Kapadia

Aditya Nair

Alfida Paiva

Project Coordinator

Aboli Ambardekar

Proofreader

Simran Bhogal

Indexer

Marriammal Chettiyar

Graphics

Yuvraj Mannari

Production Coordinator

Komal Ramchandani

Cover Work

Komal Ramchandani

About the Authors

Faisal Ghadially has 18 years experience in the technology-consulting domain. He has held leadership positions in several organizations, and is a recognized thought leader in the technology domain.

He has worked extensively with Oracle E-Business Suite, Oracle Fusion Middleware, and Oracle Fusion Applications, and has been responsible for charting the roadmap to emerging technologies such as SOA and Cloud for several organizations.

He is a regular speaker at industry events and his speeches have been published in several industry magazines.

An undertaking of this nature requires support from both friends and family. I am grateful to my family for their understanding in terms of missed baseball games and other family events.

Kalpit Parikh is an Oracle WebLogic 12c and Oracle E-Business Suite Certified Implementation specialist, and has more than 11 years' experience working on Oracle Application, Database, and Oracle Fusion Middleware, including WebLogic Server and Oracle Identity Management Suite.

He is currently working as a technical architect at Capgemini, and has worked closely with a number of customers across North America on architecting and capacity planning, installation, configuration, performance tuning, and production support.

Many people have contributed to the successful completion of this book. I would like to thank Faisal Ghadially for providing me with a great opportunity to write a book. I thank the Packt Publishing team for their support and guidance throughout this book. I appreciate Aboli and Neil's persistence and kindness during the entire project. We had a great fortune of having two great experts (Tim Warner and Saumitra Chattopadhyay) reviewing our book and providing numerous suggestions to improve the book.

I would like to thank my manager Suzanne Larabie and my colleagues at Capgemini for being very supportive and appreciative about me completing this book.

I would like to acknowledge my heartfelt appreciation to my wife Megha and son Shlok for all their sacrifice and support during the course of writing this book. I would like to thank my father, my mother, my brother, my sister as well as the rest of my family for their love and affection over the years.

About the Reviewers

Saumitra Chattopadhyay, with his IT experience spanning over 20 years, has been working with Oracle Application development technologies, and Service Oriented Architecture (SOA Suite). He has led the technical team on many large-scale Oracle Application implementations including Fusion Apps. He has played the Sr. Architect role in several award-winning implementations, showcasing the use of Oracle Fusion Middleware. His achievements are as follows:

- Expert in Oracle Application Technologies
- Oracle Certified Expert—Oracle Service Oriented Architecture
- Received the Oracle Titan award for two projects in E-Business Suite and SOA Category
- Presented SOA and MDM solution at Collaborate-11 (OAUG) and other conferences

Tim Warner is an IT professional with over 25 years' experience. He is the co-owner of Certus Solutions (UK) LLP, a leading consultancy in the field of Oracle HR/Payroll and Fusion HCM implementations.

He is the Chairman of UKOUG's Oracle HCM Special Interest Group, and has been an Oracle ACE since 2008.

He is a frequent speaker at Oracle conferences around the world, having spoken at Oracle OpenWorld, OHUG, and Collaborate in the USA as well as UKOUG and DOAG conferences in Europe.

He has a blog and a LinkedIn group (Oracle Fusion HCM Interest Group) to encourage debate and knowledge sharing on Fusion HCM.

Table of Contents

Preface

Oracle Fusion Applications are an exciting new entrant in the ERP domain. Oracle has invested heavily in building this best-of-breed application, which not only uses an industry-leading technology platform (Oracle Fusion Middleware), but also embeds leading business processes.

The component-based architecture introduces new complexity in terms of installing and administering Oracle Fusion Applications. This book addresses all aspects of Oracle Fusion Applications administration. Beginning with the installation of Oracle Fusion Applications, each chapter successively covers new areas of fusion maintenance and administration.

The book provides detailed how-to steps. However, it does this by ensuring that the high-level context and purpose of these activities are clearly understood. Our purpose in writing this book is to give you a jump-start to managing Oracle Fusion Applications from a technical perspective. Readers who have worked with other Oracle ERP products will find this is an insightful introduction to Oracle Fusion Applications.

What this book covers

Chapter 1, *Getting Started with Fusion Applications*, provides an architectural overview and an introduction to the technical components of Oracle Fusion Applications.

Chapter 2, *Fusion Application Installation*, discusses the detailed steps in installing Oracle Fusion Applications.

Chapter 3, *Functional Setup Manager Overview*, covers the functionality and usage of the Functional Setup Manager. It also explains how to use FSM to migrate setup data.

Chapter 4, *Managing Fusion Applications' Security*, provides an in-depth review of security aspects of Oracle Fusion Applications.

Chapter 5, *Enterprise Scheduler Service and Jobs*, covers the scheduling functionality of Oracle Fusion Applications.

Chapter 6, *Monitoring and Troubleshooting Oracle Fusion Applications*, delves into the audit and logging mechanisms of Oracle Fusion Applications. It also reviews diagnostic and troubleshooting details.

Chapter 7, *Managing Fusion Applications*, provides detailed information on administering Oracle Fusion Applications. This includes areas, such as patching, backup, and recovery, and extending Oracle Fusion Applications.

What you need for this book

Although not mandatory, access to an Oracle Fusion Applications environment would be a useful reference while reading this book.

Who this book is for

If you are looking to get a quick jump start in administering Oracle Fusion Applications, this book is ideal for you. It is useful for readers who would like to get a technical perspective on the architecture of Oracle Fusion Applications.

It is also useful for Database Administrators, System Administrators, and Network Administrators who are attempting to install and maintain Oracle Fusion Applications.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: " The system diagnostic module file (`Module-FMWDFW.xml`) is stored under the `$DOMAIN_HOME/config/diagnostics` directory."

Any command-line input or output is written as follows:

```
execute esspurge.purge_requests(systimestamp, 100);
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Navigate to **Help** | **Troubleshooting** | **Troubleshooting Options** | **Enable Application Logging** and choose a desired severity level."

[Warnings or important notes appear in a box like this.]

[Tips and tricks appear like this.]

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1
Getting Started with Fusion Applications

Oracle Fusion Applications (**OFA**) is an exciting entrant in the **Enterprise Resource Planning** (**ERP**) application arena. It is based on leading practices leveraged from Oracle's existing ERP suites, namely Oracle E-Business Suite, Oracle PeopleSoft, and Siebel. However, it fundamentally changes the architecture and approach for end-to-end ERP functionality.

To state that Oracle Fusion Applications is unique is an understatement. The application has been architected differently from any prior ERP application. It has been developed using industry-leading platforms that seamlessly work together. It can be deployed in multiple models, from on-premise to cloud, and various hybrid options in between.

This book will progressively explain each new feature of Oracle Fusion Applications. It will provide a detailed explanation of what each feature does, and how to configure it. Best practices are specified, where there are multiple implementation options to help the reader make an informed decision.

In this introductory section, we will put the uniqueness of Oracle Fusion Applications in context by looking at the evolution of ERP applications. We will look at the architecture and deployment models for Oracle Fusion Applications. We will then review the technology platforms on which Oracle Fusion Applications is built.

Changing the application's ecosystem

Oracle Fusion Applications and the business functionalities that it provides have evolved. New business processes have usually been the trigger for driving change in systems and applications. However, in recent years, technology has opened up novel approaches to perform business processes, and in some cases, opened up entirely new business domains (for example, Cloud or Elastic computings).

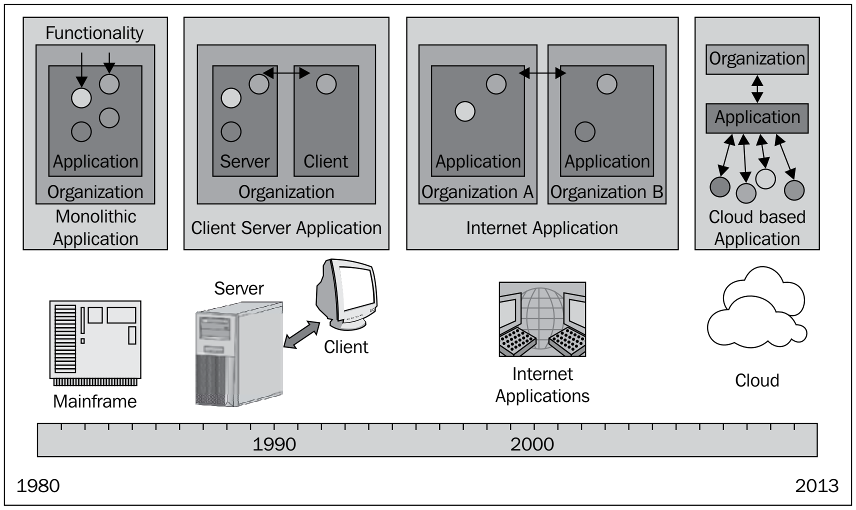
There is a healthy history of business needs and technology advances complementing each other. However, within the age of computers, technology has led to business innovation. Business users have benefited from the computing capability and flexibility that technical developments have brought to businesses. The PC era put meaningful computing capability on the desktop. This drove a change from mainframe-based monolithic data centers to business processes that empowered the end user.

An inventory report was no longer something that arrived once a month. Business users now had the capability to get business information they needed, when they needed it. This capability profoundly changed the speed of business, and more importantly, the speed in which business decisions were made. Departments such as sales, engineering, and finance now had visibility to each other's vital data. This led to efficiencies in business and ultimately higher profits and customer satisfaction.

This flexibility in the business domain during the client-server period soon extended to the consumer with the advent of the Internet. Consumers could now interact directly with businesses, which enabled a completely new dimension of customer centricity. Business users saw a delayed benefit of the Internet. The need to be interconnected outside the organization evolved with the advent of concepts, such as **Service Oriented Architecture** (**SOA**). This introduced business-process flexibility as well as business-process delegation. A need for business-process orchestration was realized.

The advent of elastic computing based on cloud-based computing models has created entirely new possibilities for businesses. Now, entire applications and their intrinsic functionalities are made available on demand, and can scale to business needs. This has opened up new avenues of doing business as well as business models that have never been seen before. Applications have started to evolve to embrace this new approach, and Oracle Fusion Applications is one such example.

The following diagram visualizes the evolution of computing paradigms and their impact on businesses. It shows how applications evolved along with changing business models. It provides an overlay with the functionality that was inherent within those applications.



The diagram shows a gradual migration of functionality away from the core of the organization to a highly distributed model. This evolution can be explained by the technology drivers that were available during each computing era.

In the mainframe era where computing resources were at a premium, applications were focused on computations. Interactions with the mainframes required advanced skills and training. There was marginal attention paid to the user interface and distributed computing.

With the advent of the client-server model, computing resources became distributed, and this led to the first era of true business applications. The client layer focused on presentation and simplistic data validation. The server layer provided data persistence and complex processing. This approach allowed the first wave of true ERP applications to be developed.

A whole host of applications from multiple vendors surfaced during this period. This included applications such as Oracle E-Business Suite, PeopleSoft, Siebel, SAP, and Great Plains, apart from several industry-specific applications.

These server-oriented ERP applications had complex data structures, which resulted in large monolithic databases. Also, the business-processing logic was tightly coupled with the database structures resulting in applications that provided good business functionality, but they had two major drawbacks, which are as follows:

- They were hard to modify or customize
- Integrations outside of the applications were complex

Middleware and SOA

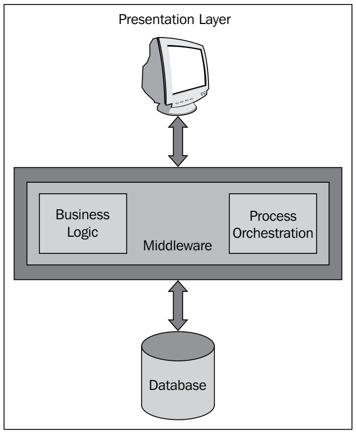
With the advent of the Internet, there was a move towards moving business applications to a web-based frontend. This resulted in the evolution of middleware that could insulate the data layer from the presentation layer. However, functionality provided by ERP applications did not fundamentally change. The Internet simply provided a powerful new mechanism for users to interact with the business applications.

Internet-based connectivity did provide new approaches for business-to-business (B2B) communication. This also led to progress in the areas of business-to-customer (B2C) capabilities. However, core business processes in the areas of Finance, Order Management, Supply Chain, and Human Resource Management remained mostly unchanged.

No more monolithic applications

The introduction of middleware provided a unique ability for business applications to communicate with each other. This communication could now be real time and could be orchestrated for multiple back-and-forth messaging patterns. This led to the development of standards that would allow the exchange of information between applications. It also resulted in a specialized approach in business processing, wherein niche applications for specialized functionality could be integrated with larger ERP applications. Typical examples of this were specialized customer service systems integrated with ERP systems, or asset management systems integrated with core financial systems.

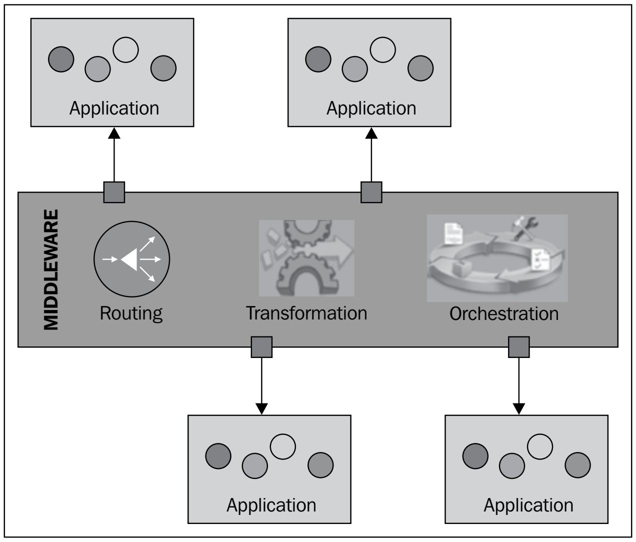
The following diagram shows the separation of the presentation and data layers. Middleware orchestrates the business processes, and in many cases, also contains the business logic that is the core process for the organization:



While the middleware products were evolving, the SOA-based approach for service integration had also been maturing. The confluence of these two accelerated the ability to weave together functionality across applications.

The following diagram illustrates the realization of the SOA-based approach in the middleware layer. Middleware products such as Oracle Fusion Middleware provide comprehensive message transformation and routing capabilities. However, the ability to provide process orchestration is vital to aligning business processes with siloed applications.

As seen in the following diagram, when applications are connected using middleware, they appear as providers of functionality. This functionality is then orchestrated into an end-to-end business process by the middleware connecting these applications. The technology stack is now much more closely aligned with the business process than it has ever been in the past.



Middleware and SOA approaches allowed the orchestration of processes that spanned multiple applications. The large ERP applications that had appeared "monolithic" up to this point now resembled the silos of functionality. ERP systems started to appear restrictive as they were limited in their flexibility, but continued to be the primary storage of institutional data.

The immense flexibility provided by SOA (enabled by middleware) stood in stark contrast to the restrictions imposed by "big-box" ERP software.

The impact of Cloud

Cloud-based deployments provided the ability to provide elastic resource capabilities. This has several technical and cost advantages. However, from a business perspective, it opened up the possibility of a "plug-in" functionality that did not have to be developed in-house.

This was a tremendously liberating aspect for the business, and its impact is similar to the dawn of the Internet age. You could now browse for functionality that was required, and pay-per-use for the same. Using middleware, this functionality could be integrated with back-end systems. It is this empowering of the business that is now fundamentally changing the ERP space.

The uniqueness of Oracle Fusion Applications

The Oracle Fusion Middleware product suite has integration and process orchestration features required for this new generation of ERP applications. It provides the platform for integration and process orchestration that is vital to a flexible ERP.

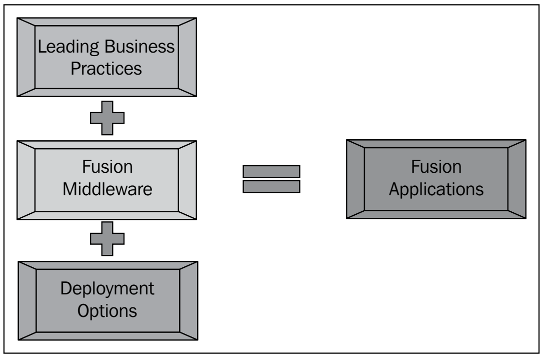
Oracle has leveraged its Fusion Middleware capability to build Oracle Fusion Applications. However, Oracle Fusion Applications are different in the following unique ways:

- **Leading practices**: Oracle Fusion Applications is built on leading practices from Oracle's stable of existing ERP products. It has those features that were considered the most meaningful and commonly used from these existing ERP product lines.
- **Oracle Fusion Middleware**: Oracle Fusion Middleware does provide comprehensive SOA-based integration. It is an industry leading middleware platform. Oracle Fusion Applications is built on this platform, and hence, inherits the value that technology stacks brings. Oracle Fusion Middleware brings some additional benefits to Oracle Fusion Applications, which are as follows:
    - **Standards based on the J2EE platform**: This is important as using Java (as opposed to a proprietary language) opens up a vast community of people who are familiar with the technology platform.
    - **Web-based frontend**: The UI is completely web-based. There is no thick client that needs to be deployed on client desktops.

    - **BPEL-based workflow**: Prebuilt business process flows are built using BPEL workflows using the BPMN notification. This replaces the implicit workflow technology that used to be highly proprietary to the application vendor.

- **Deployment**: Oracle Fusion Applications has multiple deployment options. These include the following:
    - **On-premise**: This provides a bare metal install on in-house hardware. This involves a full install of the technology's platform and the entire application.
    - **Cloud model**: This is similar to the on-premise model except that the application is installed on the infrastructure that is on the cloud.
    - **SaaS model**: **SaaS** stands for **Software as a Service** model. The application is hosted by Oracle and the subscription to the application is on a pay-per-use model. The users are not responsible for maintaining the application.

The cloud models have the additional option of being single tenant (single business users for the entire application), or multi-tenant (multiple businesses running within the same application). Thus, Oracle Fusion Applications has a wide range of deployment options that can scale based on the needs of an organization. The uniqueness of Oracle Fusion Applications is summarized in the following diagram:

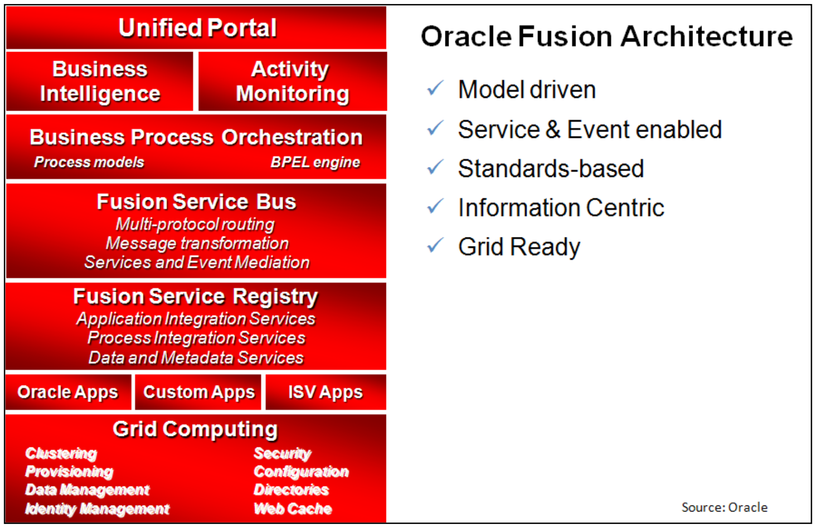


The combination of a technology platform along with leading business practices, delivered across multiple channels, makes Oracle Fusion Applications a unique ERP. It is best suited to provide the value that business users expect from their business applications.

The architecture of Oracle Fusion Applications

The architecture of Oracle Fusion Applications is shown in the following diagram at a high-level to set a baseline understanding. Components of this architecture will be discussed in detail throughout this book. Administration of key components and areas that need special attention will also be addressed.

The following diagram shows the foundational building blocks on which Oracle Fusion Applications is built. It also lists the guiding design principles that have driven the design and architecture of the overall application.



Unified Portal

Business Intelligence | Activity Monitoring

Business Process Orchestration
Process models | BPEL engine

Fusion Service Bus
Multi-protocol routing
Message transformation
Services and Event Mediation

Fusion Service Registry
Application Integration Services
Process Integration Services
Data and Metadata Services

Oracle Apps | Custom Apps | ISV Apps

Grid Computing
Clustering | Security
Provisioning | Configuration
Data Management | Directories
Identity Management | Web Cache

Oracle Fusion Architecture

- ✓ Model driven
- ✓ Service & Event enabled
- ✓ Standards-based
- ✓ Information Centric
- ✓ Grid Ready

Source: Oracle

Summary

We have discussed the evolution of Enterprise Resource Applications and the impact that technology has had on this evolution. We have also seen how functionality can be externalized to create a unique approach to business applications.

The architecture of Oracle Fusion Applications was reviewed, and the uniqueness of this suite of applications was also discussed.

2

Fusion Application Installation

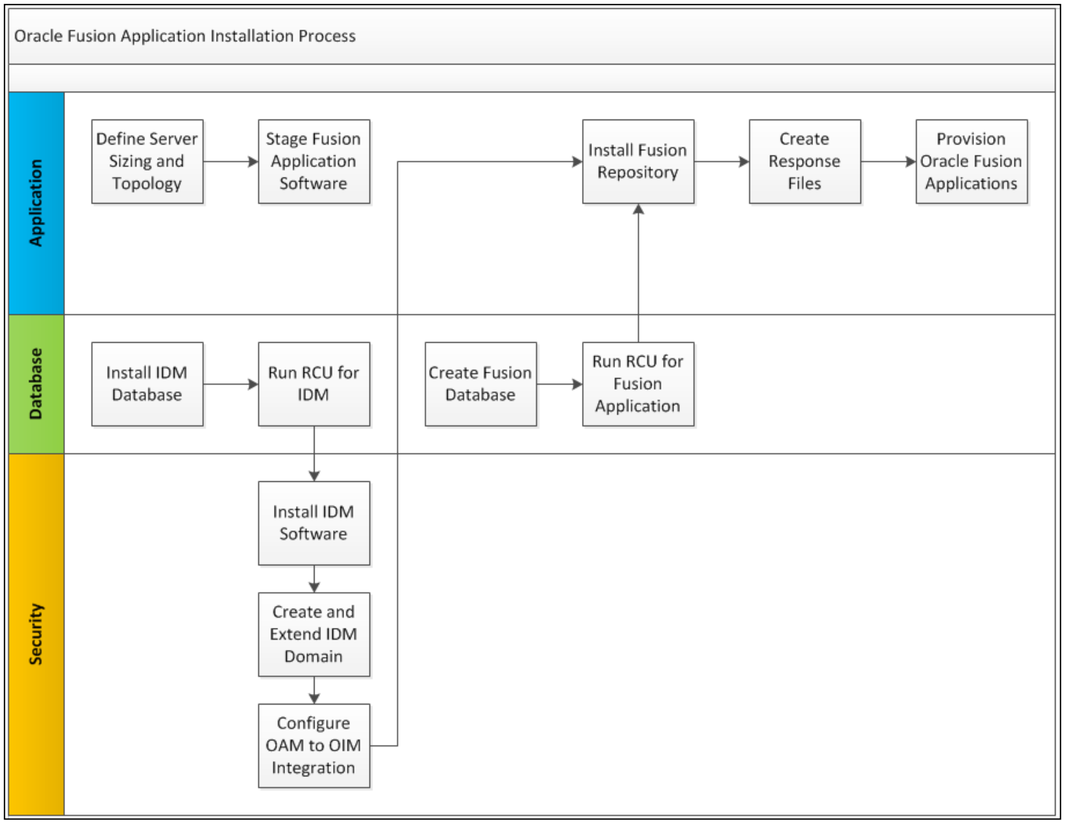
In this chapter, we will discuss the installation process for Oracle Fusion Applications. Oracle Fusion Applications is built on multiple Oracle Fusion Middleware components, and hence there is a well-defined sequence of steps required for the installation. In the following sections, we will review the installation chronology.

The following areas related to the installation of Oracle Fusion Applications will be covered:

- Installation process flow
- Server sizing
- Oracle Fusion Applications topology
- Oracle Fusion Applications security
- Oracle Database and Applications installation
- Provisioning of the application
- Troubleshooting the install

Installation process flow

The application installation has tasks to be performed at the database, security, and application levels. Tasks that need to be performed at each of these layers are sequenced, requiring careful planning and coordination. The following diagram of the process flow outlines the tasks to be performed by each of these areas. It provides a logical flow for the sequence of activities that will be performed during the installation process.


Oracle Fusion Application Installation Process
Application
Database
Security
Define Server Sizing and Topology
Stage Fusion Application Software
Install Fusion Repository
Create Response Files
Provision Oracle Fusion Applications
Install IDM Database
Run RCU for IDM
Create Fusion Database
Run RCU for Fusion Application
Install IDM Software
Create and Extend IDM Domain
Configure OAM to OIM Integration

The roles corresponding to the tasks that need to be performed are:

- **WebLogic administrators**: They perform application installation functions. They are familiar with WebLogic administration, including application deployment and troubleshooting.
- **Database administrators**: They perform database installation and maintenance functions. They are familiar with the Oracle Database.

- **OIM specialists**: They perform security-related tasks. They are familiar with the **Oracle Identity Management** (**OIM**), **Oracle Access Manager** (**OAM**), and **Oracle Internet Directory** (**OID**) suite of products.

Server sizing

Hardware requirement is driven by Product offering selected during provisioning process and concurrent user access requirements. Product offering is the highest level of collection of functionality customer can implement. The default installation for Oracle Fusion Applications installs between six to eight domains depending upon the product selection.

Memory requirement for each domain depends on type of each domain. There are two types of domains, Main and Non-main. Main Domain is the domain which is created for offering what you have selected as part of provisioning process and Common Domain. The memory requirement for this domain is larger than the rest of the domains.

Oracle Fusion Applications has a large footprint in terms of resource needs. The memory requirement alone can be as much as 140 GB. Specific resource requirements for memory and CPU vary based on the selection of products being installed, as well as the number of nodes that are part of the installation.

Typical memory requirements for the SCM Fusion Applications are provided as shown in the following list:

Products	Maximum memory
Supply Chain Management (SCM)	22 GB
Customer Relationship Management (CRM)	22 GB
Financial	9 GB
Procurement	3 GB
Human Capital Management (HCM)	6 GB
Common	15 GB
Business Intelligence (BI)	3 GB

It is important to be aware of the advances in the hardware stack that provides additional choices for the installation of Oracle Fusion Applications. Oracle Exalogic is a preconfigured appliance for Oracle Fusion Middleware. Since Oracle Fusion Applications extensively uses the Oracle FMW tech stack, an Exalogic-based install would provide comprehensive capacity and performance advantages.

Similarly, Oracle Exadata provides immense capacity and performance at the database level. Using an Exadata machine as part of the Oracle Fusion Applications installation would yield long-term benefits.

Oracle Fusion Applications topology

Oracle Fusion Applications can be installed in a multinode topology. This can include a hybrid mix of operating systems and virtualized environments. The domains can be configured to be spread across servers. It is important to identify the different hosts that are part of each domain, as these are the administrative points of entry.

The different hosts in an Oracle Fusion Applications environment are as follows:

- **Primordial host**: This is the node on which the Admin Server for the common domain resides
- **Primary host**: This is the node where the Admin Server of other domains resides
- **Secondary host**: This is the node where Managed Servers reside (multihost domain)

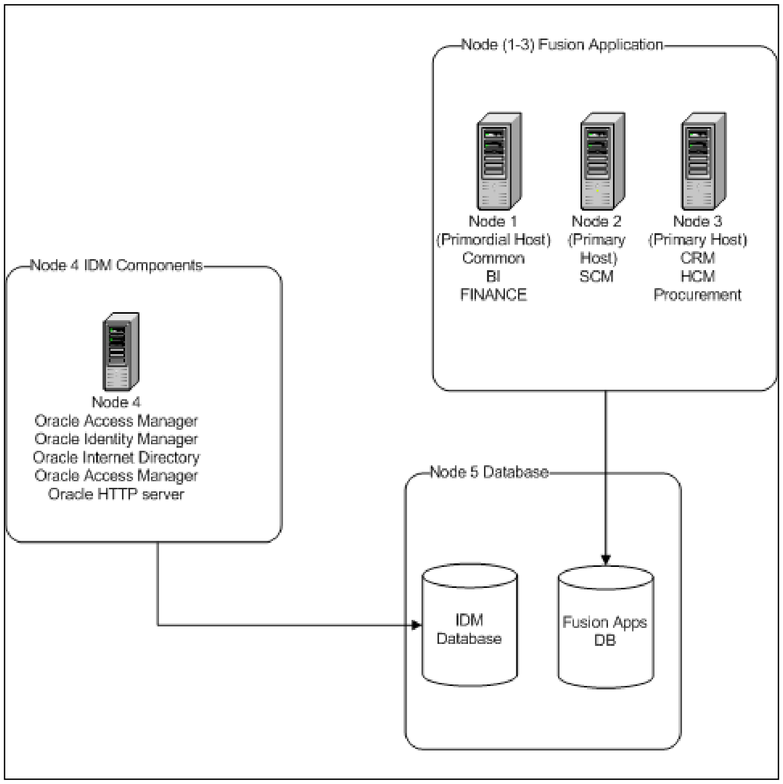
In a multihost environment, there will only be a single primordial host. There will be multiple primary and secondary hosts. The provisioning wizard is started on the primordial host.

Based on server sizing information, we have defined our topology for fusion application installation as follows:

Node	**Domains**
Node 1	Common, Financial, and Business Intelligence (BI)
Node 2	Supply Chain Management (SCM)
Node 3	Human Capital Management (HCM), Customer Relationship Management (CRM), and Procurement
Node 4	Oracle Identity Management
Node 5	Database

Each server has 32 GB RAM and two CPUs.

The high-level topology for this installation is shown in the following diagram:



Oracle Fusion Applications security

Oracle Fusion Applications is a secure out of box, and one of the pre-requisites before provisioning Fusion Applications environment is to have successful Oracle Identity Management (IDM) installation. This is a vital section of the install, as errors in this area of the install could impact Fusion Applications provisioning process.

The IDM components that are installed as part of this portion of the install are:

- Oracle Internet Directory
- Oracle Virtual Directory (optional)
- Oracle Identity Manager
- Oracle Access Manager

These components are installed using the standard installer process. We will focus on the validation aspects of OIM to ensure that the installed components are error free. The steps to be followed are as follows:

1. Log in to the IDM Domain Admin Server console and make sure the Managed Server for ODSM, SOA, OIM, and OAM are up and running.
2. Go to **Environment** | **Server**. All five servers' (including the Admin Server) health should be OK.
3. Validate Oracle Internet Directory, ODSM, and Oracle Virtual Directory.
4. Log in to ODSM by going to `http://idmhost.mycompany.com:7005/odsm`.
5. Verify the connection to OID and OVD in both SSL (LDAPS) and non-SSL (LDAP) modes.
6. Validate `jpsroot` if available.
7. Validate Oracle Access Manager.
8. Log in to the OAM Admin console and validate the login with `Oamadmin`.
9. Access `http://idm.mycompany.com:7001/oamconsole`.
10. Access Oracle Fusion Middleware control (EM) through the virtual host and port and make sure the OAM login page is accessible.
11. Go to `http://sso.mycompany.com:7777/em` and log in with the `weblogic_idm/` password.
12. Access OIM and ensure access to the homepage.
13. Go to `http://sso.mycompany.com:7777/oim`. The homepage should be accessible without a re-login.
14. Validate Oracle Identity Manager access.
15. Go to `http://sso.mycompany.com:7777/oim/faces/pages/Admin.jspx` and log in with `xelsysadm/` password.

Installing Oracle Database and applications

The database and application install is the core of the install process. This installs the database structures and application code for Oracle Fusion Applications. The installation files can be downloaded from the Oracle Software Delivery Cloud.

These files are to be staged in a predefined directory structure. This file location is referred to as the provision repository. This location should be made accessible to all of the hosts involved in the installation of Oracle Fusion Applications. Installers for various components of the install are located in this repository. The installer location is `$provision_repository_locaiton/installers`.

Provisioning framework

It is important to understand the structure of the provision repository.
All installation files are placed in the provision repository to create a provision framework. Provisioning framework provides essential supporting components to provision Fusion Applications environment. It helps to streamline the installation process.

The directory structure for the provisioning framework is as follows:

Directory	Description
ant	This is `$ANT_HOME`. It is used for ant-based deployment of applications.
bin	It includes the Provisioning Wizard. It also has the command-line utility for provisioning, and the configuration of logging and environment files.
lib	All jar files used during the provisioning process.
provision-build	Build files used by the installer during the deployment process.
provision-plan	Configuration files used by the installer during the installation of Oracle Fusion Middleware components.
template	Template database files used during the database creation process.
util	Utility tools used by the installer for infrastructure checks.

The Provisioning Wizard (mentioned in the second column of the preceding table) is a graphical UI tool used to initiate and monitor the installation process. It is the primary installer that invokes other installers as needed during the installation.

The provisioning framework supports the installation process. The installation process has seven phases as described:

- **Preverify**: This checks for all prerequisites for the installation. This is an important precheck for the installation to ensure all the infrastructure is in place. It spans all of the hosts that are part of the installation.

- **Install**: This is the installation phase. The database and middleware components that form the technology foundation for the application are installed during this phase.
- **Preconfigure**: This sets up the application for the configuration based on the response and environment files that have been updated. It is important to note that during the preconfigure phase, the application install is linked to the Fusion Database. During this phase, the `connections.xml` file in all the application EAR's is updated with the endpoint information.
- **Configure**: This creates the Managed Server in which the applications will be deployed. This phase configures the primordial host and the primary hosts.
- **Configure-secondary**: This configures the secondary hosts (if any).
- **Postconfigure**: This configures Oracle SOA Suite and the web tier. It includes the application of security policies.
- **Startup**: This starts all the Admin and Managed servers. During this phase, the **Globally Unique Identifier** (**GUID**) is set up.
- **Validate**: This starts the Managed Server and checks the deployment.

To install the provisioning framework, go to `$PROVISIONING_REPOSITORY/installers/faprov/disk1` and run the binary `./runInstallers` file.

Before starting the installation

It is important to ensure that prechecks and dependencies have been addressed before starting the install. Restarting an install is troublesome and time consuming. It is better to have all the requirements in place and checked out before initiating the installation process. The following are good areas to check prior to initiating the installation:

- Check all OS prerequisites. Ensure that the entropy value is set correctly.
- Review the latest release notes and ensure that patches for OIM and the database have been evaluated.
- Refrain from using generic hostnames, such as dev and prod. These will cause issues when cloning the environment in the future.
- During the creation of response files, do not select **Local Storage** options for the domain.

Installing the database

The Oracle Fusion Application Database can be installed in two modes:

- **Manual**: This allows the database to be created manually. Explicit configuration variables will need to be provided as part of the installation process. This approach is useful for production installations.

  This approach is also required when a database in a **Real Application Cluster** (**RAC**) mode is to be installed. During Manual process, you have to apply patches manually specified in Release Notes. This step is already taken care of when you create database through Provisioning Wizard.

  > Ensure that the **Oracle Label Security** component is enabled during the database creation process.

- **Provisioning Wizard**: This allows the creation of a preconfigured database with default configuration parameters. This is a faster mechanism to set up an Oracle Fusion Applications instance and is usually used for demo and sandbox environments.

  This approach leveraged the template databases that are made available through the provisioning framework.

  > Select the **Application Transaction Database** option when using this approach for installing the database.

Once the database is created, the fusion application tablespaces and schemas are created. This is done using the **RCU** utility. This is a self-contained utility that installs all the database components required by Oracle Fusion Applications.

To run the **RCU** utility, go to `$PROVISIONING_REPOSITORY/APPS_RCU_HOME/bin` and run the script `./rcu.sh`.

Installing and provisioning Oracle Fusion Applications

The Oracle Fusion Applications Provisioning Process uses inputs from prebuilt response files. The attributes provided in these response files feed the installers with responses during the installation process. The variables that the response files provide include:

- OIM specifics
- Database details
- Host details
- IP information
- Ports
- Products being installed
- User details and passwords

To create the response file, go to `$PROVISIONING_FRAMEWORK/provisioning/bin` and run the script `./provisioiningWizard.sh`.

The new Oracle Fusion Applications environment can be provisioned once the response file has been successfully created. As discussed earlier, the provisioning process will go through the following seven phases:

- Preverify
- Install
- Preconfigure
- Configure
- Postconfigure
- Startup
- Validate

During the install phase, the application is installed. This includes Oracle Fusion Middleware components and other aspects of Oracle Fusion Applications. The components that are installed are as follows:

- WebLogic Server

- Oracle Fusion Applications directories
- Database

The Oracle Fusion Middleware components are as follows:

- Oracle Content Management
- Oracle WebCenter
- Oracle SOA Suite
- Oracle Data Integrator
- Secure Enterprise Search
- Oracle WebGate

To provision the new fusion environment on the primordial host, go to `$PROVISIONING_FRAMEWORK/provisioning/bin` and run the script `./provisioiningWizard.sh`.

To provision the new fusion environment on all other hosts, go to `$PROVISIONING_FRAMEWORK/provisioning/bin` and run the script `./runProvisioining.sh`.

During installation options, select Provisioning new fusion application environment using existing response file.

Troubleshooting the install

During the installation of a large application, such as Oracle Fusion Applications, things can go wrong. Oracle Fusion Applications provides the ability to troubleshoot issues during the installation process. These issues can be resolved, and the installation process can be restarted from the last point of failure.

To achieve these features, Oracle Fusion Applications uses a concept called **PhaseGuards**. The installer creates a set of files in the `$APPLICATION_BASE/provisioning/phaseguards` directory. These files can be monitored during the installation process to track progress. Errors are recorded in these files. Also, a restart of the installation can be initiated by the phase guard files.

All phase guard files follow a naming convention that helps to identify the host, process, and status of the activity that is being performed. The naming convention is as follows:

Naming convention	Description
`STARTED.grd`	The phase is running and installation is in progress.
`COMPLETED.grd`	All actions as part of the phase have successfully completed.
`ENDED.grd`	The back up of the instance directory has finished. This follows the completed stage.
`FAILED.grd`	The provisioning process has failed. This file enables the **Retry** option in the installer.

The default logging level can be changed in a situation where a more detailed level of logging is required. To change the default log level, change the trace level in the `framework_location/provisioning/bin/prov-logging-config.xml` file.

Although it is not recommended, the log level for the ant scripts can also be changed to get more detailed information during ant-based deployments of the installation. To increase the log level, set `ANT_ARGS=-debug -verbose` in the `$PROVISIONING_FRAMEWORK/provisioning/bin/envConfig.sh` file.

Cleanup and restore

If an error occurs during the installation process, the installer will stop. Once the issue has been resolved, the installer can be restarted. Restarting the installer will initiate a cleanup process for that phase of the installation. This cleanup process is initiated with the primordial host, but the action is initiated across all hosts involved in the installation.

The steps performed during cleanup depend on the phase and the products being installed. It usually includes:

- Shutting down Admin Server and Managed Server (for failed products)
- Shutting down BI processes
- Shutting down web-tier processes
- Shutting down **Informatica Identity Resolution** (**IIR**)

This cleanup is in anticipation of the restore process. During the restore process, the response files and other variables are read, and the installation process is restarted. It is assumed that the issue that caused the original error has been resolved at this point. The steps performed during the restore step are as follows:

1. Deleting the restart phase, that is, the `name.grd` file located at `$APPLICATION_BASE/provisioning/restart`.
2. Restoring `instance.tar` from the backup taken after the last successful phase.
3. Starting the Common Domain Admin Server.
4. Starting the web-tier processes.
5. Performing **Lighweight Directory Access Protocol** (**LDAP**) cleanup.

Once the restore is complete, the application installation can proceed.

There are unique situations where an error can be ignored and the installer needs to be forced to continue. This can happen in situations where there are known issues that have been confirmed by product support. In such cases, the following steps need to be performed:

- Remove the `$APPLICATION_BASE/provisioning/phasegurads/phase-name-FAILED.grd` file
- Create the `$APPLICATION_BASE/provisioning/phasegurads/phase-name-COMPLETED.grd` file

This allows the installer to continue by overlooking the failed activity.

Deinstalling Oracle Fusion Applications

If there is a need to deinstall Oracle Fusion Applications, the Provisioning Wizard provides a deinstall utility. To deinstall all the components of Oracle Fusion Applications, the following steps need to be executed:

1. Use the Provisioning Wizard and the command-line utility to deinstall the application.
2. The Provisioning Wizard should be used on the primordial host. The command line should be used on all other hosts.

3. Use the DBCA utility to delete the database.
4. Uninstall the OIM components using the installer with the **Deinstall** option.
5. It is important to stop all the running components of the Oracle Fusion Applications environment before running the deinstall. Once the deinstall process is complete, all directories and files will need to be manually deleted.

Summary

In this chapter we discussed the installation of Oracle Fusion Applications. We reviewed the prerequisites and server sizing needed for the install. The various servers and the topology of the application were discussed. Application security and its importance to the installation process were discussed in detail.

The provisioning framework was reviewed, along with the phases of the application install. The database installation and application provisioning was reviewed. Finally, troubleshooting the install and methods to clean up and restart the application install were discussed.

3
Functional Setup Manager Overview

Configuring a large ERP system is time consuming due to the sheer number of parameters that need to be configured. Also, there are dependencies between parameters and the different roles within an organization that need to work in tandem for the right configuration.

Once an environment is configured, strong processes are required to ensure that only approved changes are made. Propagating a configuration change requires strict change control. Finally, figuring out what went wrong in an environment requires tedious comparisons across instances. Based on all of the above, there is a clear need for stream-lined configuration management for an ERP environment.

We will review the following aspects of the **Functional Setup Manager** (**FSM**):

- Architecture and features
- Using FSM
- Migrating configurations using FSM

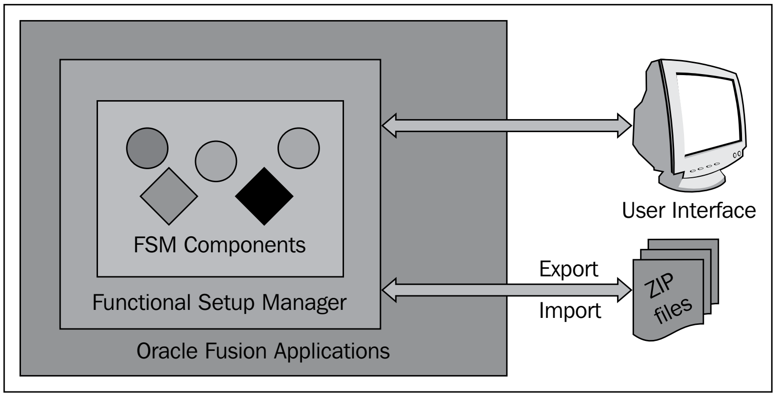
The architecture of Functional Setup Manager

Large applications face the following issues when it comes to configuration:

- Ensuring all configurations are done correctly
- Configurations are sequenced correctly
- Collaboration across various roles is achieved
- Ability to move configurations across environments

To alleviate these issues, Oracle Fusion Applications have taken a fundamentally different approach to configuration by introducing the FSM. The FSM externalizes configurations and provides the ability to modularize the application configuration process.

The high-level architecture for the FSM is shown as follows:



An overview of FSM feature

The FSM provides a host of features that streamline configuration management in Oracle Fusion Applications. The main features are as follows:

Feature	Description
Consolidated UI	A single-user interface for managing all the aspects of configurations
Dependency Management	Ability to configure based on a decision tres
Configuration Management	Ability to export and import configuration and configuration subsets
Predefined Configurations	Pre-built configuration sets for standard setups that accelerate the implementation process
Custom Objects	Ability to extend configurations with custom setup objects

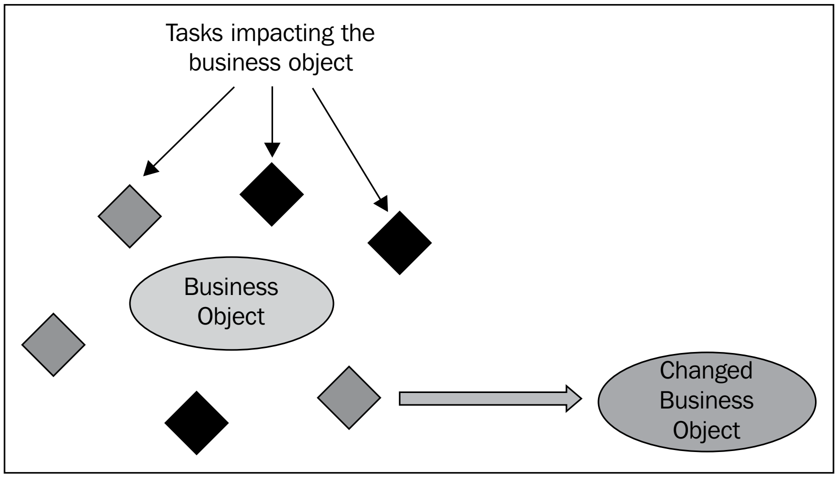
With this rich feature set, FSM fundamentally changes the approach for the configuration of an environment. It makes configurations portable and allows rapid deployments.

FSM components

FSM has a set of foundational components as part of its architecture. These components are grouped together to provide the various features of FSM. The primary component of FSM is the **Business Object**.

A business object is any foundational entity that is part of an organization's process. These can be both physical and conceptual entities. Examples of business objects are employee, invoice, order, item, and ledger.

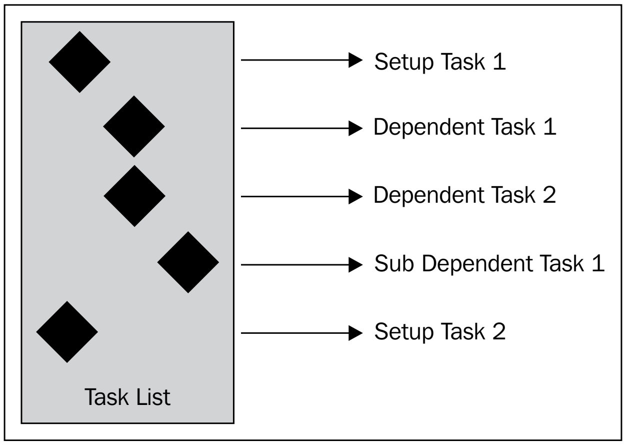
These business objects have activities that can be performed on them. The state of a business object would change based on the activity that is executed. The FSM refers to these activities as **Tasks**. The relationship between a business object and a task is shown in the following diagram:



Tasks are classified into the following two types:

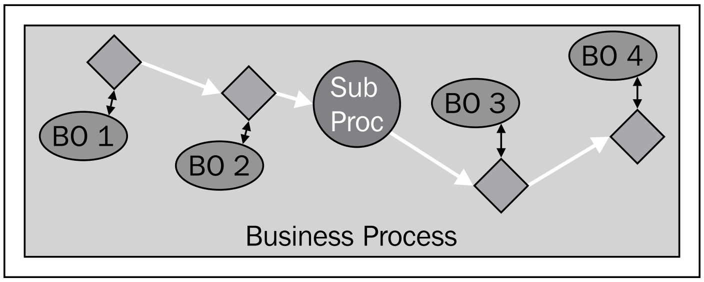
- **Setup Tasks**: These tasks are performed during the configuration of the application and are usually one-time events that need to be performed to configure the application. Examples of setup tasks are assigning reporting currency and creating item structure, just to name a few important tasks.
- **Tasks**: These are transactional activities that are performed on a regular basis and are the standard functions that would be performed on a business object.

Setup tasks are not performed in isolation. They are usually part of a sequence of steps that need to be performed in a predefined order. The FSM provides **Task Lists** that group setup tasks and ensure that they are interlocked correctly.

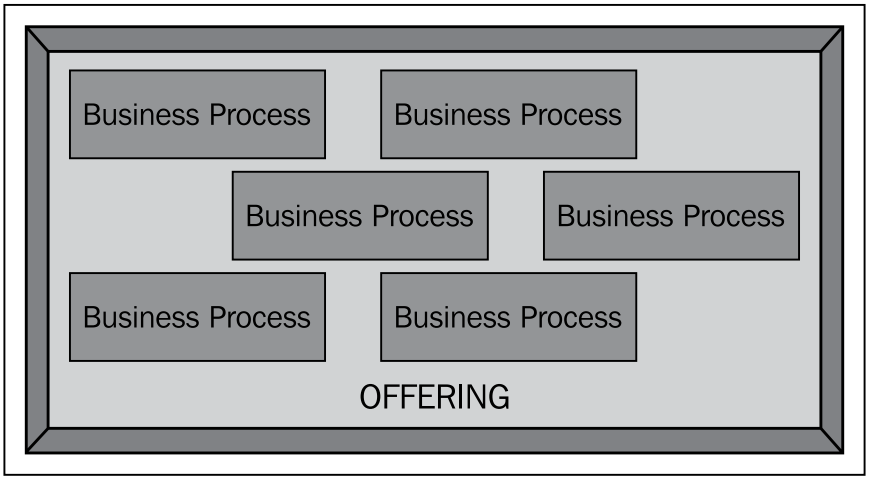


Tasks (distinctly different from setup tasks) are performed as part of a transaction and are usually grouped together to complete a business objective. This grouping of tasks from a logical perspective is achieved in the FSM using the concept of a **Business Process**.

A business process is a sequence of tasks that need to be performed to complete a business transaction. These tasks can span multiple business objects. The granularity of business processes is designed in a manner that promotes reuse, hence, there are **sub processes** that have limited scope but are reused across different business processes as shown in the following figure:

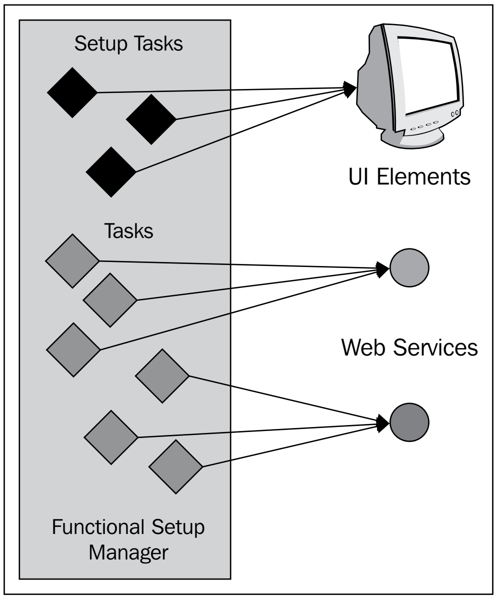


At a macro-level business, processes are grouped together by logical domains to create functionality grouped within a functional domain. This macro-level grouping of business processes creates an **Offering**. Examples of offerings are financials and procurement. Offerings are further controlled by the licensing models that have been acquired as shown in the following figure:



These logical components are surfaced to the user using UI elements and web services. Setup task lists are made available in the form of UI forms. Users can then interact and execute these setup tasks. Oracle Fusion Applications also provide a logical grouping in the UI in the form of work areas. Based on the users' role and area of setup, this work area would show related tasks.

Similarly, transactional tasks are exposed as web services. These can be invoked for execution. The groups of tasks that are orchestrated as a unit are also exposed as a combined business service as shown in the following figure:

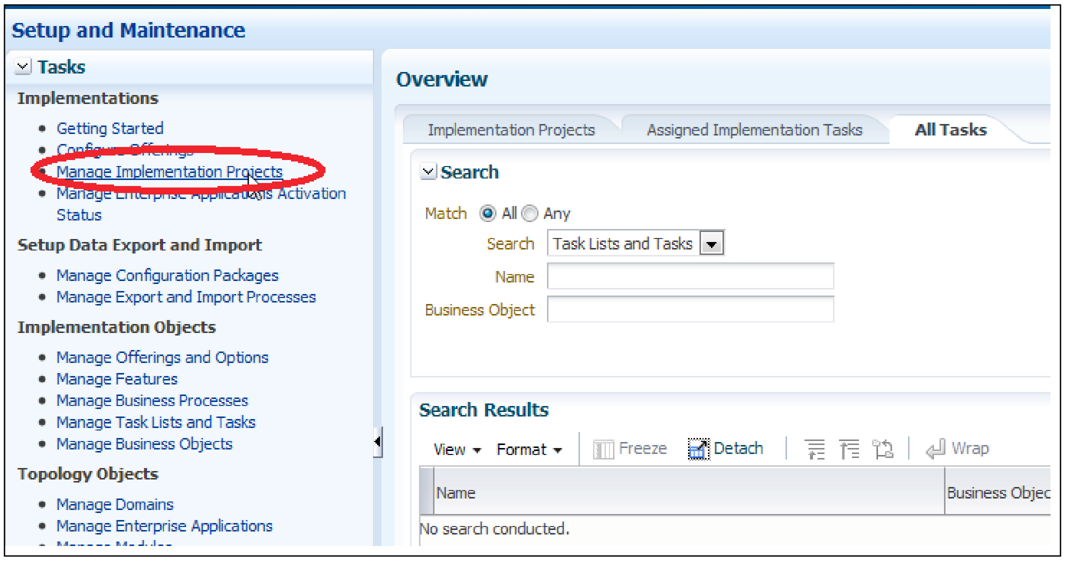


An implementation project

An implementation project is a bundle of setup tasks required to configure product offerings or particular business processes. The implementation project allows a project manager or implementation manager to assign a list of setup tasks to users or functional consultants and provides reporting functionality to the track progress of each task. The implementation project could include all the available product offerings or we could segment multiple implementation projects to have a manageable list of setup tasks.

To create an implementation project, the user should have the Application Implementation Manager Role. The following steps will create an implementation project:

1. Click on the navigation menu and then on **Setup and Maintenance**.
2. Click on the **Manage Implementation Projects** link. Through **Manage implementation Projects**, you can search for existing projects or create a new one:



Setup and Maintenance

Tasks

Implementations

- Getting Started
- Configure Offerings
- Manage Implementation Projects
- Manage Enterprise Applications Activation Status

Setup Data Export and Import

- Manage Configuration Packages
- Manage Export and Import Processes

Implementation Objects

- Manage Offerings and Options
- Manage Features
- Manage Business Processes
- Manage Task Lists and Tasks
- Manage Business Objects

Topology Objects

- Manage Domains
- Manage Enterprise Applications
- Manage Modules

Overview

Implementation Projects | Assigned Implementation Tasks | **All Tasks**

Search

Match ● All ○ Any

Search: Task Lists and Tasks

Name:

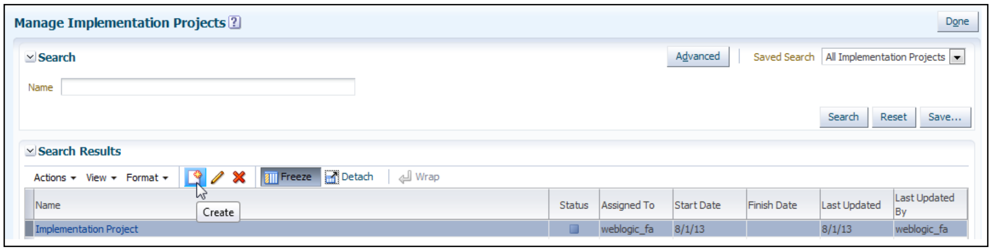
Business Object:

Search Results

View ▾ Format ▾ Freeze Detach Wrap

Name	Business Objec
No search conducted.	

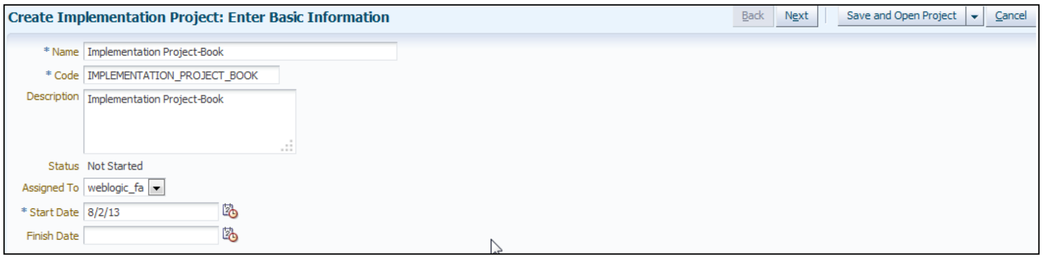
3. Click on the Create **Button**:



Name	Status	Assigned To	Start Date	Finish Date	Last Updated	Last Updated By
Implementation Project	■	weblogic_fa	8/1/13		8/1/13	weblogic_fa

4. This will open the **Create Implementation Project** page. On this page, start by providing a meaningful implementation project name. Populate the project code and description fields using the appropriate text. Note that the code field only accepts uppercase letters (A-Z) and numbers (0-9). The user that created the project is the default owner of the project. This can, however, be changed using the drop-down list of values.

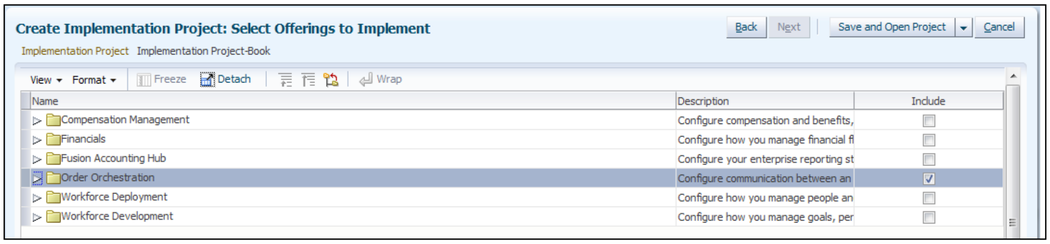
   The start and end dates for the project are useful from an analytics perspective. Data within those data ranges would be used for analytic reporting:



* Name	Implementation Project-Book
* Code	IMPLEMENTATION_PROJECT_BOOK
Description	Implementation Project-Book
Status	Not Started
Assigned To	weblogic_fa
* Start Date	8/2/13
Finish Date	

5. Click on the **Next** button. On the next page, you can select the product offering you would like to include as part of the implementation project. Click on **Save and Open Project**. This will generate the setup task lists based on the product offering and options selected.

   The implementation project does not necessarily have to be part of an offering. It can be a stand alone implementation project. Preconfigured tasks and/or task lists can be included with an implementation project:

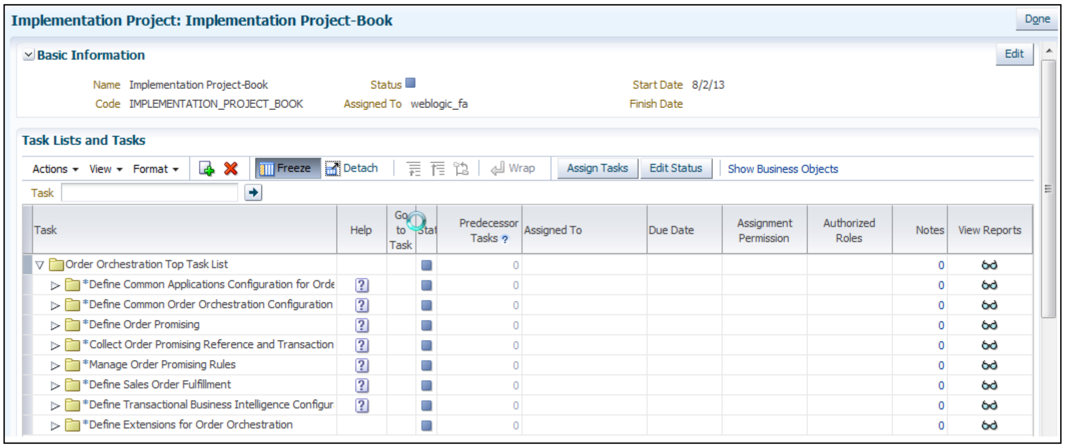


Implementation Project Implementation Project-Book

Name	Description	Include
Compensation Management	Configure compensation and benefits,	□
Financials	Configure how you manage financial fi	□
Fusion Accounting Hub	Configure your enterprise reporting st	□
Order Orchestration	Configure communication between an	☑
Workforce Deployment	Configure how you manage people an	□
Workforce Development	Configure how you manage goals, per	□

6. Note that task lists generated for common tasks for all the offerings are positioned first, followed by product-specific tasks. The sequence of tasks is important as there exists a dependency between successive tasks. The generated task lists includes end-to-end tasks required to configure product offerings.

   The tasks start with synchronization tasks and continue all the way through to product-specific tasks. The task lists generated for the implementation project will drive the implementation. Business objects associated with each task can be viewed through the **Show Business objects** link. To assign individual setup tasks or task lists to users or function consultants, click on **Assign Tasks**.



Implementation Project: Implementation Project-Book

Done

Basic Information

Edit

Name Implementation Project-Book
Code IMPLEMENTATION_PROJECT_BOOK
Status ■
Assigned To weblogic_fa
Start Date 8/2/13
Finish Date

Task Lists and Tasks

Actions View Format Freeze Detach Wrap Assign Tasks Edit Status Show Business Objects

Task

Task	Help	Go to Task	Stat	Predecessor Tasks ?	Assigned To	Due Date	Assignment Permission	Authorized Roles	Notes	View Reports
Order Orchestration Top Task List			■	0					0	👓
*Define Common Applications Configuration for Orde	?		■	0					0	👓
*Define Common Order Orchestration Configuration	?		■	0					0	👓
*Define Order Promising	?		■	0					0	👓
*Collect Order Promising Reference and Transaction	?		■	0					0	👓
*Manage Order Promising Rules	?		■	0					0	👓
*Define Sales Order Fulfillment	?		■	0					0	👓
*Define Transactional Business Intelligence Configur	?		■	0					0	👓
*Define Extensions for Order Orchestration			■	0					0	👓

7. Click on **Done**.

Setup migration

The Oracle FSM not only guides you through functional setup but also provides you with the functionality to migrate setup from Dev to Test or production environment. Through Oracle FSM, we can export and import setups from one environment to another, which reduces the overall implementation time and avoids human errors.

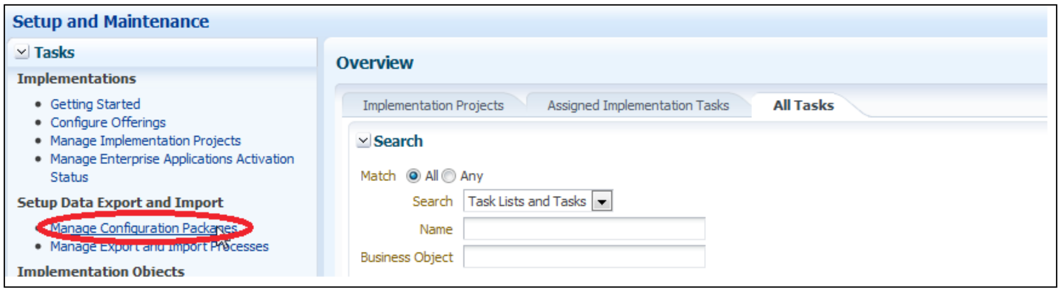
The following steps explain the high-level process to migrate setup tasks from the Test environment to production:

1. Create a configuration package definition
2. Export setup tasks (with or without setup data)
3. Import setup tasks

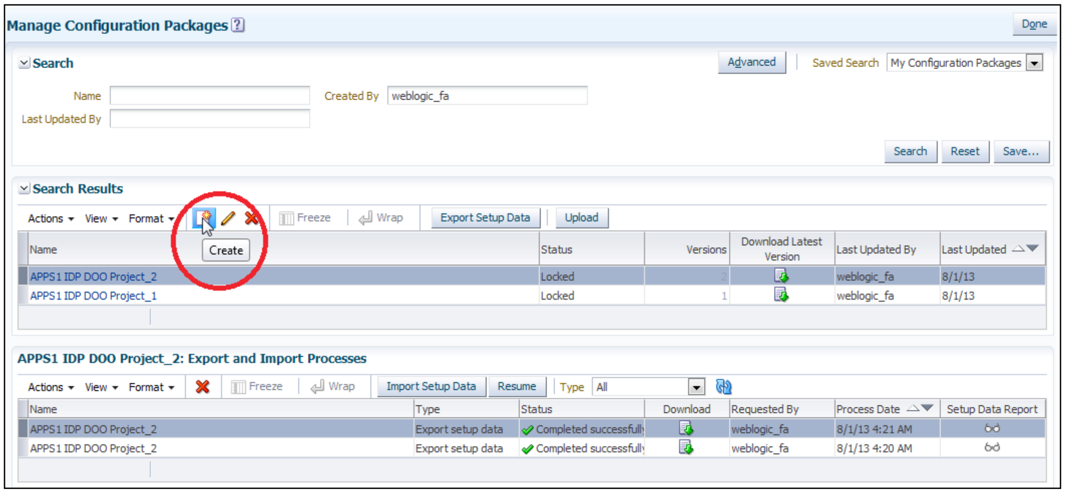
A configuration package directs export and import process in which setup tasks and their setup data are part of the migration. It contains export and import process definitions.

Steps for creating configuration package and export setup data:

1. Click on the navigation menu and then click on **Setup and Maintenance**.
2. Click on **Manage Configuration Package**:


Setup and Maintenance
Tasks
Implementations
Getting Started
Configure Offerings
Manage Implementation Projects
Manage Enterprise Applications Activation Status
Setup Data Export and Import
Manage Configuration Packages
Manage Export and Import Processes
Implementation Objects
Overview
Implementation Projects
Assigned Implementation Tasks
All Tasks
Search
Match ● All ○ Any
Search Task Lists and Tasks
Name
Business Object

3. Click on **Create**:



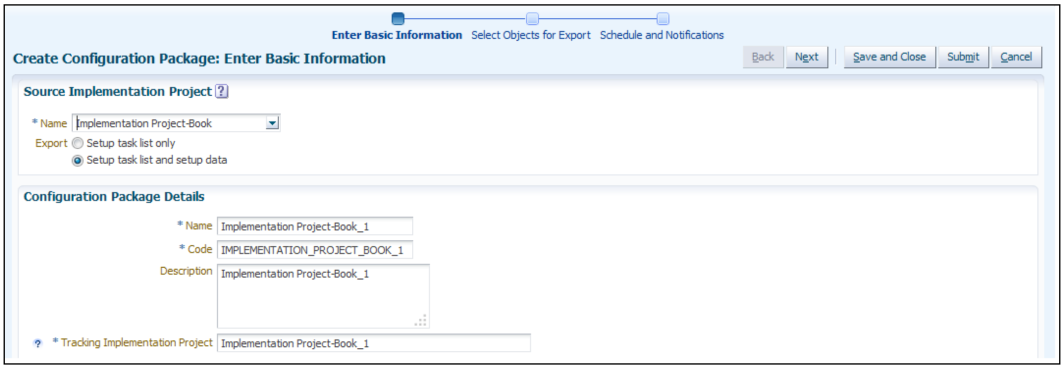
Name
Created By weblogic_fa
Last Updated By

Name	Status	Versions	Download Latest Version	Last Updated By	Last Updated
APPS1 IDP DOO Project_2	Locked	2		weblogic_fa	8/1/13
APPS1 IDP DOO Project_1	Locked	1		weblogic_fa	8/1/13

APPS1 IDP DOO Project_2: Export and Import Processes

Name	Type	Status	Download	Requested By	Process Date	Setup Data Report
APPS1 IDP DOO Project_2	Export setup data	✓ Completed successfull		weblogic_fa	8/1/13 4:21 AM	
APPS1 IDP DOO Project_2	Export setup data	✓ Completed successfull		weblogic_fa	8/1/13 4:20 AM	

4. Provide the **Configuration Package** details. Select **Source Implementation Project** from drop-down list. Setup tasks and their associated business objects that are part of **Source Implementation Project**, define export and import and the sequence for the configuration package. We can create a configuration package that only includes setup tasks or setup data. The **Tracking Implementation Project** is a copy of the source implementation project, which tracks setup tasks and setup data after import.



Create Configuration Package: Enter Basic Information

Source Implementation Project

* Name: Implementation Project-Book

Export: ○ Setup task list only
● Setup task list and setup data

Configuration Package Details

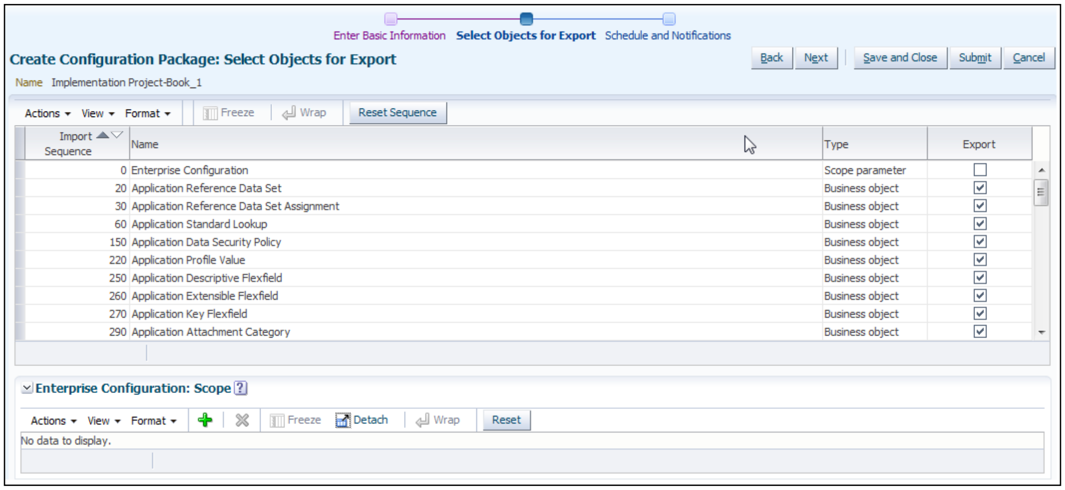
* Name: Implementation Project-Book_1

* Code: IMPLEMENTATION_PROJECT_BOOK_1

Description: Implementation Project-Book_1

* Tracking Implementation Project: Implementation Project-Book_1

5. By default, the configuration package includes all of the setup tasks and associated business objects that are part of the source implementation project. We can customize a configuration package and choose only business objects that we would like to migrate. An import sequence is derived from the source implementation project and we can customize it in order of the business object imported. We need to consider data dependency before modifying the sequence to avoid errors during migration. We can filter data that we would like as part of migration using the screen text options. For example, if migration should only include data related to a particular legal entity, by using the **Scope** option, we can choose an appropriate legal entity and export only the process data part of that legal entity:



Create Configuration Package: Select Objects for Export

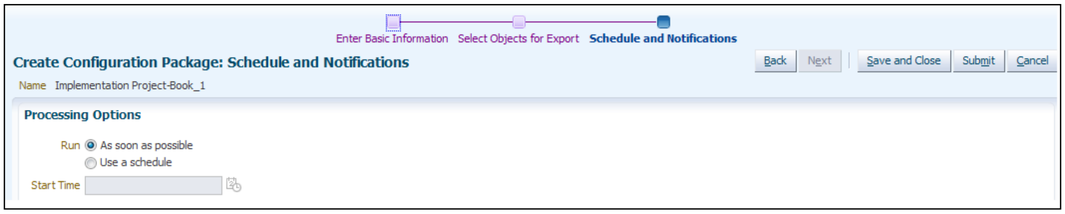
Name Implementation Project-Book_1

Import Sequence	Name	Type	Export
0	Enterprise Configuration	Scope parameter	□
20	Application Reference Data Set	Business object	☑
30	Application Reference Data Set Assignment	Business object	☑
60	Application Standard Lookup	Business object	☑
150	Application Data Security Policy	Business object	☑
220	Application Profile Value	Business object	☑
250	Application Descriptive Flexfield	Business object	☑
260	Application Extensible Flexfield	Business object	☑
270	Application Key Flexfield	Business object	☑
290	Application Attachment Category	Business object	☑

Enterprise Configuration: Scope

No data to display.

6. After creating the configuration package, we can run an export immediately or schedule a process:



Create Configuration Package: Schedule and Notifications

Name Implementation Project-Book_1

Processing Options

Run ● As soon as possible
○ Use a schedule

Start Time

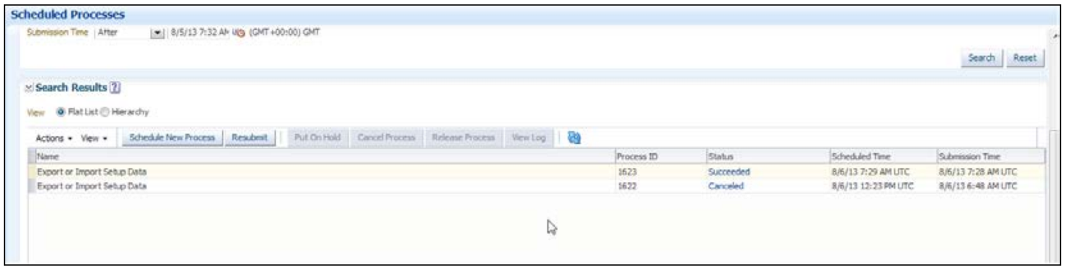
7. Clicking on **Save and Close** will create a configuration package with the **definition in progress** status.

At this time, the configuration package only includes export and import definitions and allows modifications to the existing configuration.

Clicking on **Submit** will start the export process as per requested schedule, lock the configuration package definition, and not allow any more modification.

Another approach to start the export process is given as follows:

1. Click on **Manage Configuration Package**.
2. Search for the configuration package created by following steps 1-7 in the preceding set of instructions.
3. Click on **Export Setup Data**. In the background, the FSM will submit the **Export or Import Setup Data** job. You can also monitor progress and review logs through the **Schedule Processes** workarea.



Scheduled Processes

Submission Time After 8/5/13 7:32 AM (GMT +00:00) GMT

Search Results

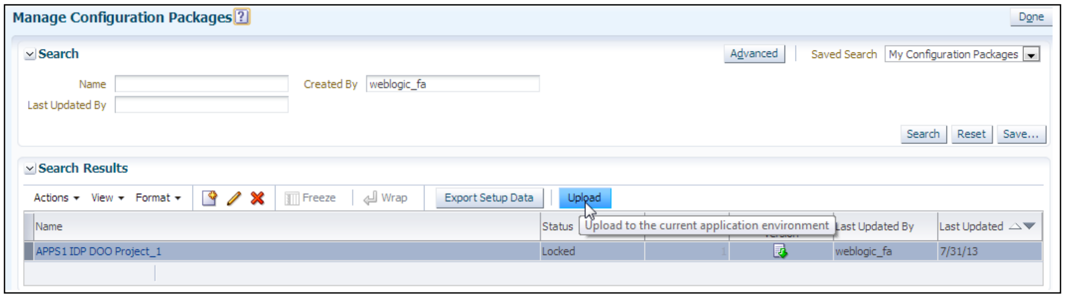
View ● Flat List ○ Hierarchy

Name	Process ID	Status	Scheduled Time	Submission Time
Export or Import Setup Data	1623	Succeeded	8/6/13 7:29 AM UTC	8/6/13 7:28 AM UTC
Export or Import Setup Data	1622	Canceled	8/6/13 12:23 PM UTC	8/6/13 6:48 AM UTC

Before proceeding with the import, download `Configurationpackage.zip` from the successful export. The `ConfigurationPackage.zip` file includes setup tasks (XML files) and setup data (XLS files).

The steps for importing the setup data are as follows:

1. Click on the navigation menu and then on **Setup and Maintenance**.
2. Click on **Manage Configuration Package.**
3. Click on **Upload**. On the next screen, browse to the `ConfigurationPackage.zip` file downloaded from the source environment and upload it to the target environment. Verify the configuration details before proceeding with the import by clicking on **Get Details**. If all the details look good, click on **Submit**:



Manage Configuration Packages

Search

Saved Search: My Configuration Packages

Name:

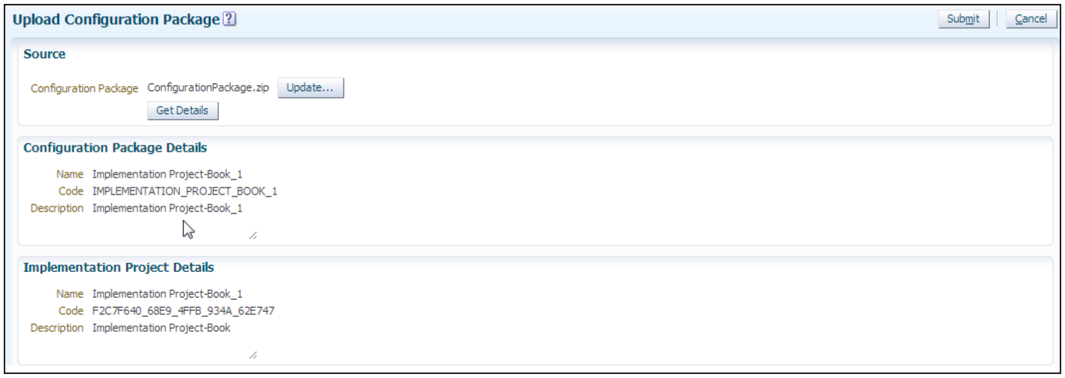
Created By: weblogic_fa

Last Updated By:

Search Results

Upload to the current application environment

Name	Status			Last Updated By	Last Updated
APPS1 IDP DOO Project_1	Locked	1		weblogic_fa	7/31/13



Source

Configuration Package ConfigurationPackage.zip

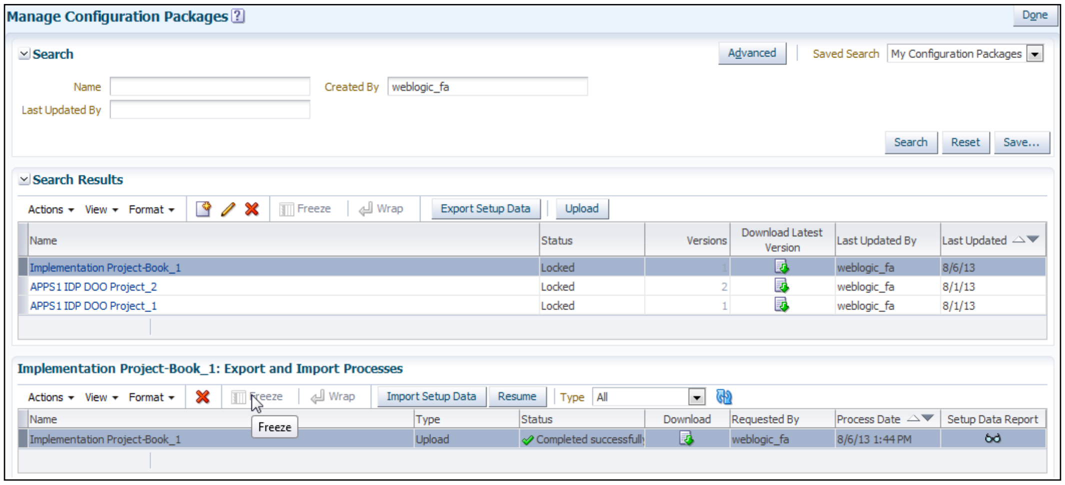
Configuration Package Details

Name Implementation Project-Book_1
Code IMPLEMENTATION_PROJECT_BOOK_1
Description Implementation Project-Book_1

Implementation Project Details

Name Implementation Project-Book_1
Code F2C7F640_68E9_4FFB_934A_62E747
Description Implementation Project-Book

4. We can now see a configuration package on the target environment with type **Upload**:



Manage Configuration Packages
Done
Search
Advanced
Saved Search My Configuration Packages

Name
Created By weblogic_fa
Last Updated By

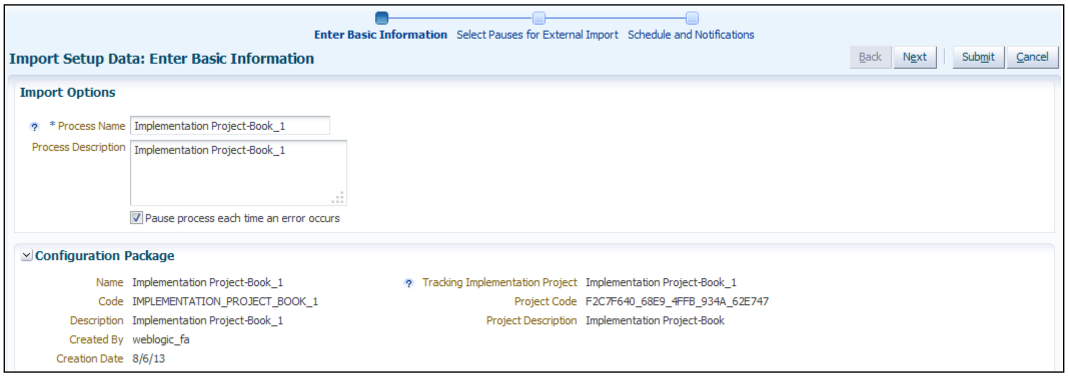
Name	Status	Versions	Download Latest Version	Last Updated By	Last Updated
Implementation Project-Book_1	Locked	1	⭳	weblogic_fa	8/6/13
APPS1 IDP DOO Project_2	Locked	2	⭳	weblogic_fa	8/1/13
APPS1 IDP DOO Project_1	Locked	1	⭳	weblogic_fa	8/1/13

Implementation Project-Book_1: Export and Import Processes

Type All

Name	Type	Status	Download	Requested By	Process Date	Setup Data Report
Implementation Project-Book_1	Upload	✓ Completed successfull	⭳	weblogic_fa	8/6/13 1:44 PM	👓

5. Click on **Import Setup Data**. Review the information on the next two screens and click on **Submit** to submit the import process:



Import Setup Data: Enter Basic Information

Import Options

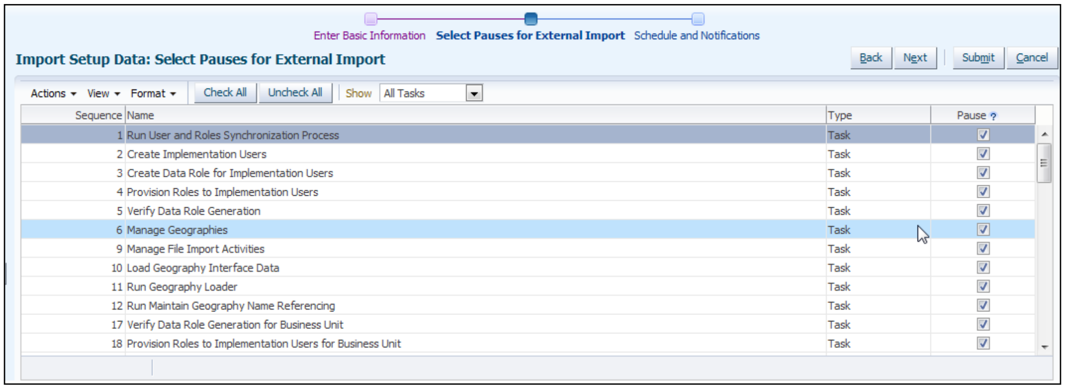
* Process Name: Implementation Project-Book_1

Process Description: Implementation Project-Book_1

☑ Pause process each time an error occurs

Configuration Package

Name	Implementation Project-Book_1	Tracking Implementation Project	Implementation Project-Book_1
Code	IMPLEMENTATION_PROJECT_BOOK_1	Project Code	F2C7F640_68E9_4FFB_934A_62E747
Description	Implementation Project-Book_1	Project Description	Implementation Project-Book
Created By	weblogic_fa		
Creation Date	8/6/13		

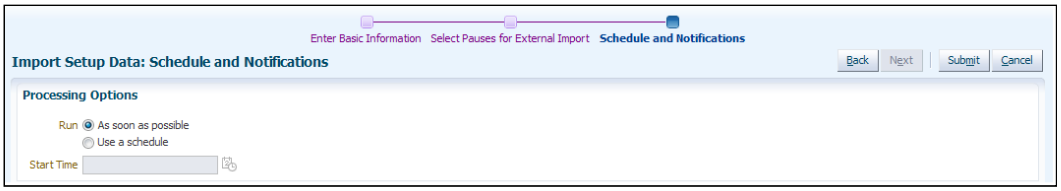


Import Setup Data: Select Pauses for External Import

Show: All Tasks

Sequence	Name	Type	Pause
1	Run User and Roles Synchronization Process	Task	☑
2	Create Implementation Users	Task	☑
3	Create Data Role for Implementation Users	Task	☑
4	Provision Roles to Implementation Users	Task	☑
5	Verify Data Role Generation	Task	☑
6	Manage Geographies	Task	☑
9	Manage File Import Activities	Task	☑
10	Load Geography Interface Data	Task	☑
11	Run Geography Loader	Task	☑
12	Run Maintain Geography Name Referencing	Task	☑
17	Verify Data Role Generation for Business Unit	Task	☑
18	Provision Roles to Implementation Users for Business Unit	Task	☑

Monitor the import process and if the process stopped for any reason, review the status details for more information and take the appropriate actions:



Import Setup Data: Schedule and Notifications

Processing Options

Run ● As soon as possible
○ Use a schedule

Start Time

Moving setup files

The wizard-based approach uses web services to extract configuration information. This data is persisted in XML files. The entire set of exported configurations is bundled into a ZIP file. This file forms the basis on which a target instance is configured.

On the target side, the ZIP file is imported. This is decomposed into the corresponding XML files. This configuration data is then used to setup the Oracle Fusion Application instance. A web service-based approach is used to read the setup data and perform the corresponding setup tasks.

Summary

In this chapter, we reviewed the Functional Setup Manager and how it helps to streamline application configuration. We discussed in detail the steps needed to create implementation projects. The migration of setups from one environment to another using an export- or import-based approach was also discussed in detail.

In the following chapter, we will review the security aspects of Oracle Fusion Applications.

4
Managing Fusion Applications' Security

Large ERP applications have multiple users. Each user interacts with the application for different purposes. It is important that each user has access to the appropriate amount of data and functionality. There would be process and security issues if all users have access to all data contained within an application.

Application security has multiple aspects. Oracle Fusion Applications provide a comprehensive approach as well as a toolset for managing application security. The following areas of security will be reviewed:

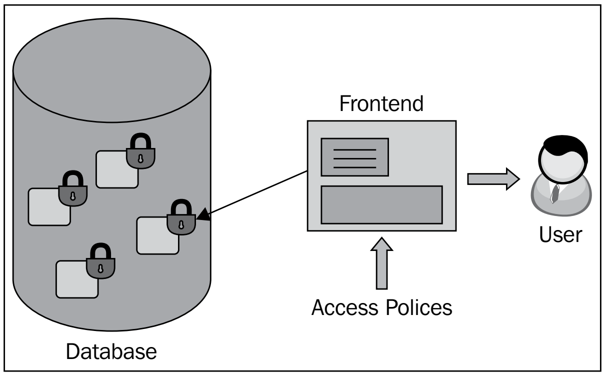
- An overview of application security
- User management in Oracle Fusion Applications
- A comparison of security architecture

An overview of application security

Applications provide users with the ability to perform business transactions. These transactions both create data and operate on existing data. Additionally, several kinds of reporting would be performed on the enterprise data that is created.

Applications have complex interdependencies as this is usually the nature of the business being performed. A data change in one aspect of an application can have far-reaching consequences in the other domains of an applications. Strong rules need to be in place in order to ensure that data being altered is governed by business logic. These rules also need to be enforced.

Applications channel access to data using a frontend user interface. This interface forms a controlled medium through which the user performs business transactions. These transactions, in turn, affect underlying data objects. An effective frontend would provide a logical structure for the tasks that an end user needs to perform.



These data controls and the user interface, collectively, provide portals for the application's security. This kind of security is required for the following reasons:

- Business integrity needs to be maintained across transactions
- Segregation of duties needs to be enforced
- Malicious use of data needs to be prevented

The definition of application security extends beyond controls. It also involves the ability to prevent unapproved access from external entities. This falls in the realm of network security. Applications have to be built such that they can be properly secured from unauthorized external access.

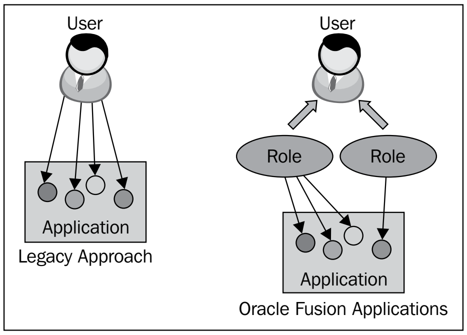
While preventing unauthorized access from external entities, organizations need to exchange data with external sources. The proper handling of data that enters and exits the application is also part of application security. Applications within an organization also need to interact with each other. These interactions require user context to be carried from one application to another.

Clearly, there are several layers to security. Each aspect is important and has to be addressed appropriately. At the same time, application security cannot be too restrictive. It needs to be both efficient and pragmatic.

Oracle Fusion Applications' security concepts

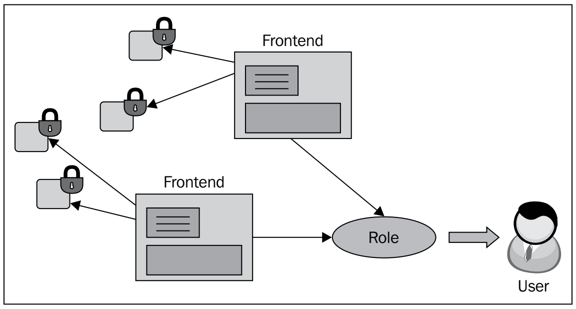
Oracle Fusion Applications handles security by addressing functional security as opposed to user security. Legacy applications focused on business users created an access profile for each user. This worked well when each user had a clearly defined set of tasks and there was a minimal overlap in the functionality between users.

Business and technology have evolved, such that application users have multiple roles. These roles and functions are overlapping and change over time. Hence, Oracle Fusion Applications have taken an approach that is role-based as opposed to user-based.



This approach to security uses the function and the data that it works on as the fundamental units of security. These functions can then be grouped together in various combinations to create higher-level groupings. However, this grouping does not change the atomic level at which Oracle Fusion Applications are secured.

This approach that secures both the business activity and its corresponding data subset is unique to Oracle Fusion Applications. Although the approach is not unique (it is referred to as Role-Based Access Control), the granularity at which it is enforced is fundamentally different.



Besides internal security aspects, the application perimeter needs to be secured as well. Oracle Fusion Applications' security is built on the **Oracle Identity Management** (**OIM**) suite. This provides comprehensive role-based security as well as network and message security.

OIM is intrinsic to the Oracle Fusion Middleware suite of products. Hence, it provides a seamlessly integrated platform for security. OIM includes comprehensive security for web services. This is important from an Oracle Fusion Applications' perspective as web services is the primary mechanism of delivered functionality.

Web services also help in integrating across multiple deployment options as well as integrating with cloud-based functionality. Having OIM as the center of security provides a comprehensive approach to overall application and data security.

Oracle Fusion Applications' security architecture

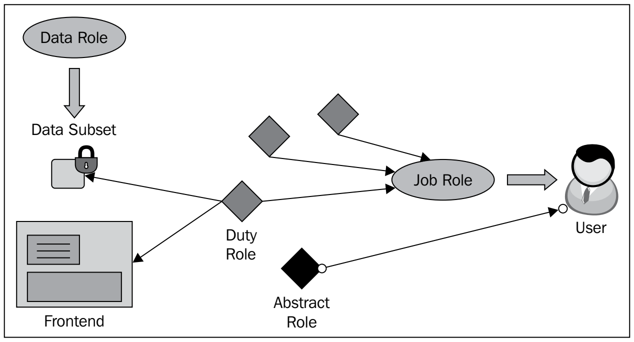
Oracle Fusion Applications' security is based on roles. These roles are the aggregation of atomic security privileges that are grouped together in various combinations. Since roles are a fundamental aspect of Oracle Fusion Applications, we will review different kinds of roles and the hierarchy in which they work.

Roles in Oracle Fusion Applications

Oracle Fusion Applications provides the following roles:

Roles	Description of roles
Job roles	This is an aggregation of roles that defines the profile of a business user. These roles are logical groupings of business functions at the highest level.
Abstract roles	These roles cannot be directly assigned to an end user. They are used to create logical security blocks, which are then further grouped together.
Duty roles	These roles are the most granular. They carry entitlements to business functions and related data.
Data roles	These roles provide explicit entitlements to data.

Abstract roles tend to be generic roles that would be implicit to multiple users. A good example for an Abstract role is an employee. Most business users of an application would be an employee of the organization. This role would be appended to other more specific profiles that would create a specialized security entitlement for a user.



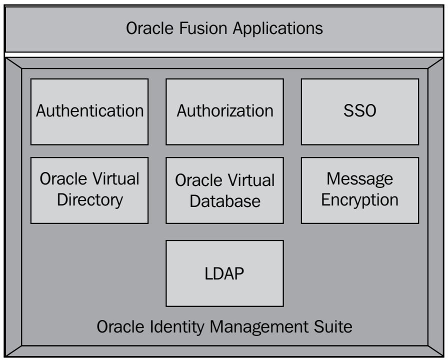
Segregation of duties

A user cannot have roles that transcend normal checks and balances in a business process. For example, a user should not have the right to issue a purchase order, and then make payment for the same purchase order. Oracle Fusion Applications have a built-in set of rules that enforce the **segregation of duties** (**SOD**). The configuration and enforcement of SOD in Oracle Fusion Applications is done by the **Application Access Controls Governor** (**AACG**). This is a product that is part of **Oracle's Governance Risk and Compliance** (**GRC**) family.

AACG is used to define policies at the role or entitlement level. These policies are then enforced whenever an entitlement to a role is created or a role is assigned to a user. There is a predefined list of security policies that implicitly enforce SOD controls. These can be further refined as needed to meet an organization's needs.

Security architecture

The OIM suite provides the technology foundation for security in Oracle Fusion Applications. Several components of OIM are used to implement internal application security as well as perimeter security.



The **Single Sign-On** (**SSO**) capability provides the ability to seamlessly authenticate across multiple applications within an organization. This is especially useful when a user's security profile needs to be carried across applications.

The **Oracle Virtual Directory** (**OVD**) provides the capability to virtualize **Lightweight Directory Access Protocol (LDAP**) and provide multiple sources for authentication. This plays a pivotal role in federated security situations where authentication across organizational boundaries needs to be carried out.

OVD provides private database functionality. This provides granular security of data, and it is especially useful in situations where **Personal Identifiable Information** (**PII**) needs to be secured.

Message line and on-the-wire encryption capabilities are provided by OIM. There are multiple options for encryption, and these policies can be applied in any combination as needed by the business process.

Oracle Fusion Applications primarily use roles to provide application authorization. However, roles are not limited to access and authorization alone. They also provide data segregation and access controls around data privacy. Oracle Fusion Applications are designed to be multitenant applications. Hence, data partitioning is a vital feature of the application. Oracle Fusion Applications use a combination of roles and features, such as **Multi Org Access Control** (**MOAC**), to ensure that data access is segregated based on the user type. This is vital for the application's end users as well as for security compliance.

User management

User roles are assigned to named user accounts for authentication and authorization purposes. Oracle Fusion Applications primarily have two types of named users:

- Users who configure and implement Oracle Fusion Applications: This type of user is referred to as the *Implementation User*.
- Users who administer Oracle Fusion Applications: This type of user is referred to as the *Application User*.

Irrespective of the user type, each user is provisioned in OIM as well as created in the Oracle Fusion Applications' table structures.

Implementation user

The primary implementation user is **FAADMIN**. This is a super user account created during the installation process. This user has the following roles:

- Application implementation role
- IT security manager role

The IT security manager role includes the OIM roles that are required for user creation. These include the administrator and identity user administration roles. This super user has all the access rights required to configure offerings and perform setup tasks. The security guidelines dictate that at least one implementation user should be created who will define the enterprise structure. In non-production environments, this user can be limited to a single account.

Oracle Fusion Applications provide a task list for the creation of implementation users. This task list is the *Create Implementation User* task list. The actions in this task list direct the super user to specific OIM pages where the user is created.

Since implementation users are responsible for creating the enterprise structure, no such structure exists when these users are created. Hence, implementation users, when created, are not assigned to the HR structure, and no entries are created in the `PER_USER` table. Implementation users will not have access to any functionality in Oracle Fusion Applications until one of the following predefined roles is assigned to them:

- **Application implementation manager**: This role has the following rights:
    - To manage an implementation project
    - To assign tasks
    - To setup export and import processes
- **Application implementation consultant**: This role has broad access rights. It includes the ability to set up all enterprise structures.

Implementation users can be restricted to specific data subsets. This is done by creating a new data role using the *Create Data Role for Implementation Users* task list.

Application user

Once the enterprise structure is defined, end users or application users can be created. These users are the true end users of the application. An application user is created by invoking the *Manage User* task.

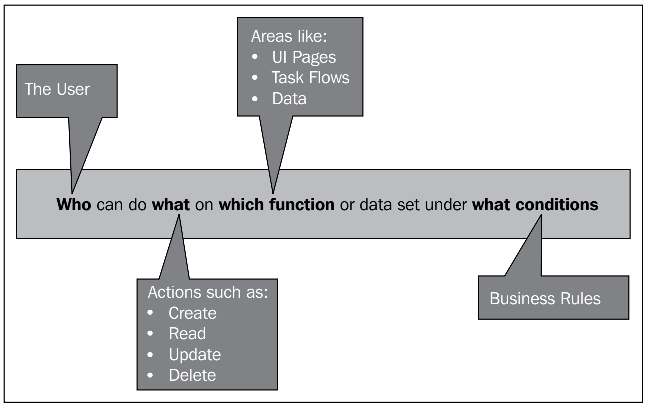
The Manage User task does not use OIM pages. All user creation information is gathered from the Oracle Fusion Applications' HCM pages. Once all the information has been collected, an internal web service call is made to provision the user in OIM. To maintain security, this service call is made using the **Service Provisioning Markup Language** (**SPML**).

Application users have rights to transact with Oracle Fusion Applications. They have the ability to create and modify application data via application-defined functionality. Application users have an entry in the `PER_USER` table.

Before creating a user it is recommended to run the **Retrieve Latest LDAP Change** job. This retrieves the latest roles and any role changes from OIM so that they are available in Oracle Fusion Applications.

Managing authorization

Oracle Fusion Applications define authorization as "Who can do what on which function or data set under what conditions." This maps to the components of Oracle Fusion Applications as follows:

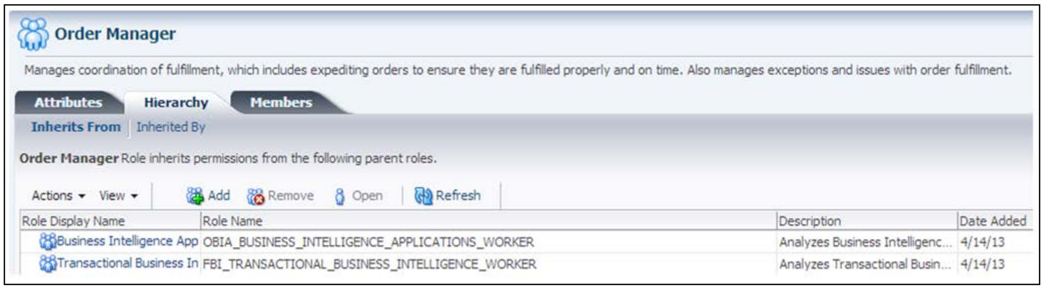


The job, abstract, and data roles that are used to implement the authorization approach are stored in the LDAP. The LDAP is a part of the OIM suite and is persisted within the OIM suite. From an application maintenance perspective, Oracle Fusion Applications provide a user interface for managing all enterprise roles. This is the **Oracle Provisioning Manager** (**OPM**).

Duty roles are also known as applications' roles since they are specific to actions that can be performed within Oracle Fusion Applications. Duty roles are stored in the policy store, and their scope is restricted to the application.

Oracle Fusion Applications have a comprehensive list of predefined enterprise roles. These roles can be managed via APM. APM also provides the ability to extend the existing role definitions. These custom roles extend the authorizations provided by the existing roles.

New job roles can be created using the **Manage Job Roles** task. This task links to the APM home page and provides the ability to add job roles to the existing role hierarchy. The following screenshot shows the role hierarchy for the **Order Manager** role in OIM:

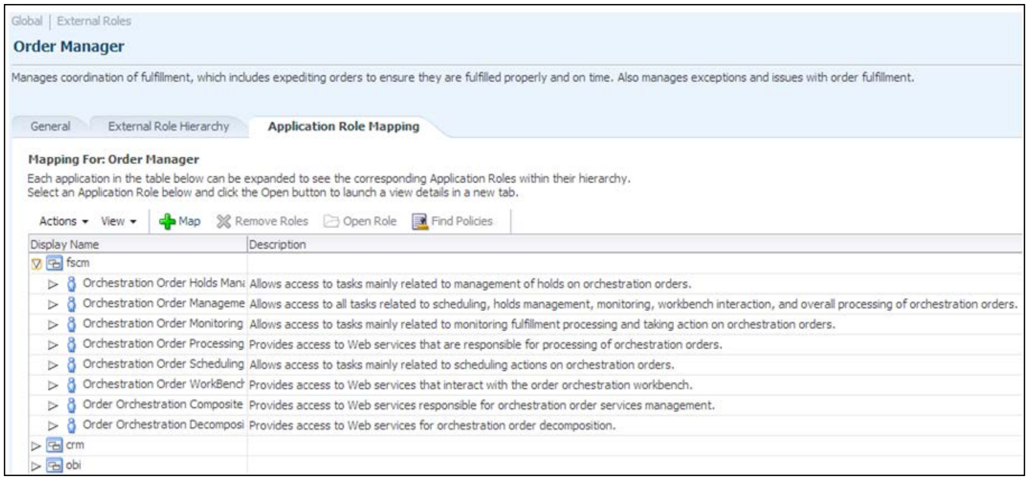


Manages coordination of fulfillment, which includes expediting orders to ensure they are fulfilled properly and on time. Also manages exceptions and issues with order fulfillment.

Order Manager Role inherits permissions from the following parent roles.

Role Display Name	Role Name	Description	Date Added
Business Intelligence App	OBIA_BUSINESS_INTELLIGENCE_APPLICATIONS_WORKER	Analyzes Business Intelligenc...	4/14/13
Transactional Business In	FBI_TRANSACTIONAL_BUSINESS_INTELLIGENCE_WORKER	Analyzes Transactional Busin...	4/14/13

The duty role hierarchy is managed through the Oracle Authorization Policy Manager. The **Manage Duties** task provides the ability to view duty roles inherited by enterprise roles. It also shows entitlements and policies granted by the enterprise role hierarchy. The hierarchy for the predefined **Order Manager** job role is provided for reference.



Manages coordination of fulfillment, which includes expediting orders to ensure they are fulfilled properly and on time. Also manages exceptions and issues with order fulfillment.

Mapping For: Order Manager

Each application in the table below can be expanded to see the corresponding Application Roles within their hierarchy.
Select an Application Role below and click the Open button to launch a view details in a new tab.

Display Name	Description
▽ fscm	
▷ Orchestration Order Holds Mana	Allows access to tasks mainly related to management of holds on orchestration orders.
▷ Orchestration Order Manageme	Allows access to all tasks related to scheduling, holds management, monitoring, workbench interaction, and overall processing of orchestration orders.
▷ Orchestration Order Monitoring	Allows access to tasks mainly related to monitoring fulfillment processing and taking action on orchestration orders.
▷ Orchestration Order Processing	Provides access to Web services that are responsible for processing of orchestration orders.
▷ Orchestration Order Scheduling	Allows access to tasks mainly related to scheduling actions on orchestration orders.
▷ Orchestration Order WorkBench	Provides access to Web services that interact with the order orchestration workbench.
▷ Order Orchestration Composite	Provides access to Web services responsible for orchestration order services management.
▷ Order Orchestration Decomposi	Provides access to Web services for orchestration order decomposition.
▷ crm	
▷ obi	

A comparison of security architecture

Oracle Fusion Applications build on leading practices from other major ERP Suites. A comparison with Oracle EBS is listed as follows to provide a comparative point of view for the security architecture within Oracle Fusion Applications.

Oracle Fusion Applications	Oracle E-Business Suite
Authentication:	**Authentication:**
Does not have its own user store User management is based on task flows Integrated with Oracle Identity Management Oracle Access Manager provides SSO capabilities LDAP-based user repository	Has its own authentication built-in using AOL table structures User maintenance is done via System Administrator responsibility SSO needs to be configured and is not implicit to the application
Authorization:	**Authorization:**
Data role or Abstract role	Responsibility
Job role	Top-level menu
Duty role or Application role	Submenu
Privilege	Form function

Summary

In this chapter we reviewed the security architecture of Oracle Fusion Applications. Oracle Fusion Applications provide a role-based approach for managing security within the application. This transcends frontend elements as well as data subsets.

The different types of roles and their relationships were discussed. User creation and related screenshots used for this task were also reviewed. A comparison with Oracle EBS was provided to enhance the understanding of Oracle Fusion security architecture.

5
Enterprise Scheduler Service and Jobs

ERP applications are perceived primarily as transactional systems. Business users access ERP environments to perform day-to-day business operations. The frontend UI and backend data structures are geared for this type of daily interaction. This daily processing during business hours, however, is not the bulk of system processing.

Large amounts of data are processed by applications that have no human interaction. These tasks include the following activities:

- Data aggregation and maintenance
- Predefined reporting
- Major business-processing periods (such as month-end close)

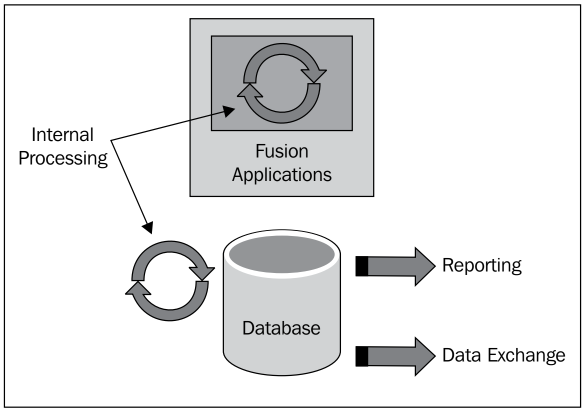
Many of these tasks can be scheduled to execute at predefined intervals. This scheduling of jobs is now a common feature in all major ERP applications. We will review these scheduling engines and the manner in which Oracle Fusion Applications provides for Enterprise scheduling.

Enterprise schedulers' overview

ERP applications consume transactional data during normal business hours. These human-based interactions feed data into the underlying systems. Once this data has been captured, downstream processing needs to be carried out. This processing can be categorized as follows:

- **Processing**: This implies the processing of data based on predefined business rules. This includes data manipulation and summarization.

- **Reporting**: This process helps in generating resource-intensive reports that require aggregation across large data subsets. It also helps to generate periodic and scheduled reports required for statutory purposes.
- **Data Exchange**: This process is used to transfer data within and outside the organization. This includes data synchronization processes between systems. It also includes EDI and other data exchanges outside the organization.



To perform these kinds of scheduled jobs, a scheduling engine is required. Historically, the **Operating System** (**OS**) would provide this kind of utility service. In UNIX-based systems, this was called **cron**. This approach worked for simplistic non-rule based scheduling. The UI for interacting with the cron was usually text-based, and required a highly structured programming construct.

The OS-based scheduler would not work for complex ERP processing. The ERP applications required a scheduler that would have the following features:

- **Complex Schedules**: This provides the ability to process complex scheduling needs, for example, every third week or every alternate day, except on the last week of the quarter.
- **Dependencies**: This provides the ability to understand dependencies between scheduled jobs. This ensures that the processing follows business rules. It ensures that the logic dependencies are maintained.
- **Grouping**: This provides the ability to group tasks and execute them together. This provides for parallelism as well as the ability to enforce data integrity.
- **Reprocessing**: This provides the mechanism to reprocess jobs when tasks fail to complete successfully. A clean restart is required.
- **Monitoring**: This allows us to monitor jobs that are running. It also helps us to monitor the upcoming schedules and status of jobs that have been completed.

- **Logging and Alerts**: This provides detailed logging of the jobs. It also includes alerts for job completion and failures.

Oracle Fusion Applications' Enterprise scheduler

Fusion Applications provides a scheduling engine called the Oracle Enterprise Scheduler. This is embedded within Oracle Fusion Applications and does not require a separate installation. The Oracle Enterprise Scheduler provides a comprehensive package of scheduling features for Oracle Fusion Applications. The jobs that can be scheduled using this scheduler are as follows:

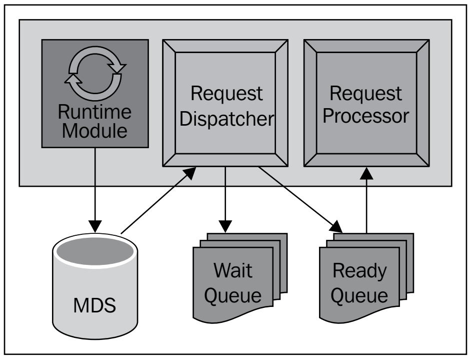
- Java packages
- OS based scripts
- PL/SQL packages

This combination of job types accommodates almost any kind of scheduling job needs. This includes simple OS-level commands and scales up to complex functionality that can be coded within Java. The PL/SQL option allows database-specific functionality that needs to be executed.

Details about all scheduled jobs are stored in the metadata repository. This repository can be queried for job status as well as reporting purposes. This metadata repository is embedded within the Oracle Fusion Applications database.

The architecture of Oracle Enterprise Scheduler

The Oracle Enterprise Scheduler uses a combination of processes and queues to perform its functions. The high-level architecture is as follows:



The primary component is **Runtime Module**. This is the core engine of the scheduler. It processes scheduling requests by updating the metadata repository. It puts jobs to be processed in the **Wait Queue**. The **Request Dispatcher** polls the Wait Queue and the metadata repository and moves jobs to the **Ready Queue**. The **Request Processor** processes the jobs in the Ready Queue.

The Oracle Enterprise Scheduler allows complex scheduling needs. This is achieved by a set of advanced functionalities such as:

- Parameterization
- Job sets
- Job incompatibility
- Subrequests

Jobs can be set up to be based on parameters that are passed at runtime. This parameterization allows dynamic runtime changes. Parameterization also helps in the reuse of IC components that are invoked.

Jobs can be grouped together to be executed in sets. This enables logical grouping of longer job schedules. These job sets either complete successfully or fail as a group. This provides enhanced reprocessing capabilities.

Certain jobs should not be run concurrently. In order to provide for this dependency, incompatibility between jobs can be defined. Once these incompatibilities are defined, the scheduler will be able to enforce these dependencies. Subrequests allow the running of jobs in parallel threads. This improves the overall performance while improving resource utilization.

Maintaining Oracle Enterprise Scheduler (ESS)

In this section, we will look at general maintenance for the Oracle Enterprise Scheduler. This includes the following areas:

- Starting or stopping the scheduler
- Purging scheduler logs
- Troubleshooting scheduler issues

Start and Stop ESS

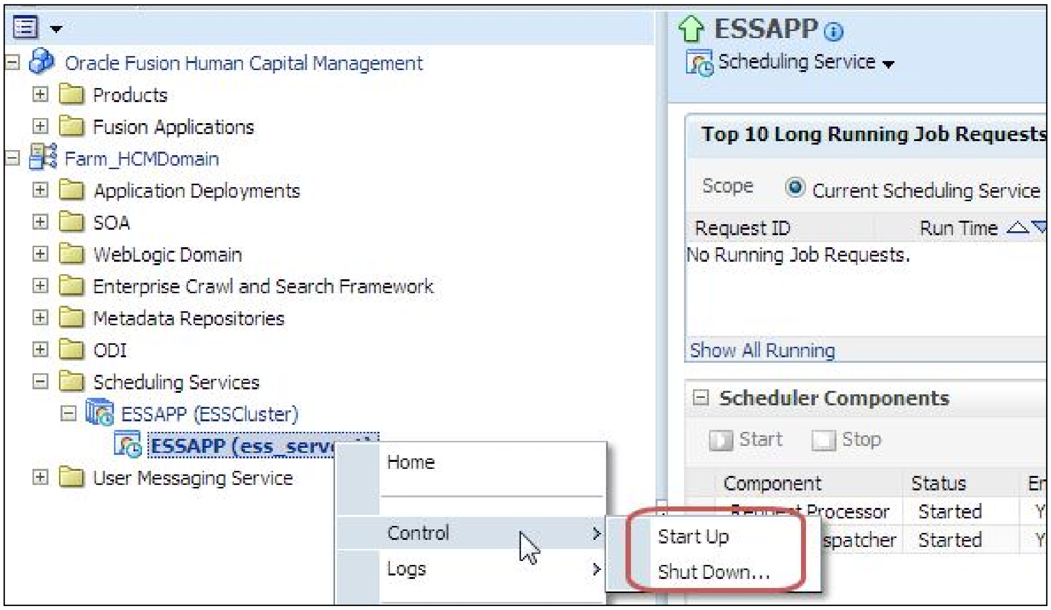
The ESS components that can be managed are:

- The ESSAPP application (Oracle Enterprise Scheduler instance)
- Components (Request Processor and Request Dispatcher)

The ESSAPP application is deployed on the ESS Managed server, which allows us to control the start/stop for a Request process, a Dispatcher, and an Oracle ESS instance. It also provides an interface to view and edit the job definition, search for job requests, purge polices, and so on. We can use the Fusion Middleware control to start/stop an instance and ESS components.

To start/stop the Oracle Enterprise Scheduler instance, perform the following steps:

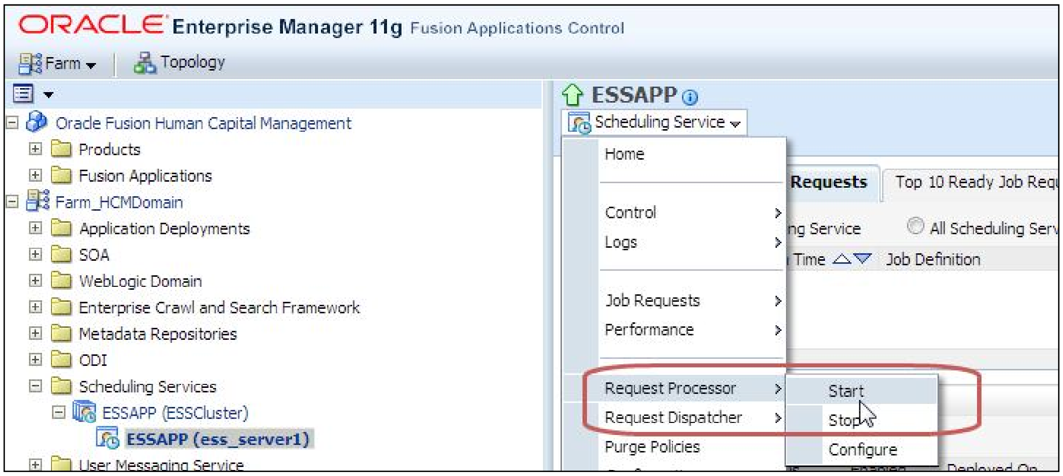
1. Log in to **Fusion Middleware Control** (`http://hostname:port/em`).
2. In the directory tree, navigate to the folder **Farm_HCMDomain** (or any domain name) then expand **Scheduling Services**.
3. Click on **ESSAPP**.
4. Navigate to **ESSAPP | Control** and click on **Shut Down...** or **Start Up**, shown as follows:



Request ID	Run Time
No Running Job Requests.	

Component	Status	En
Request Processor	Started	Y
spatcher	Started	Y

Now, **Request Processor** and the **Request Dispatcher** can be started or stopped as shown in the following screenshot:



Purging ESS jobs

Oracle Fusion Applications approaches purging in two phases. The phases are in the following order:

- Logical purging
- Physical purging

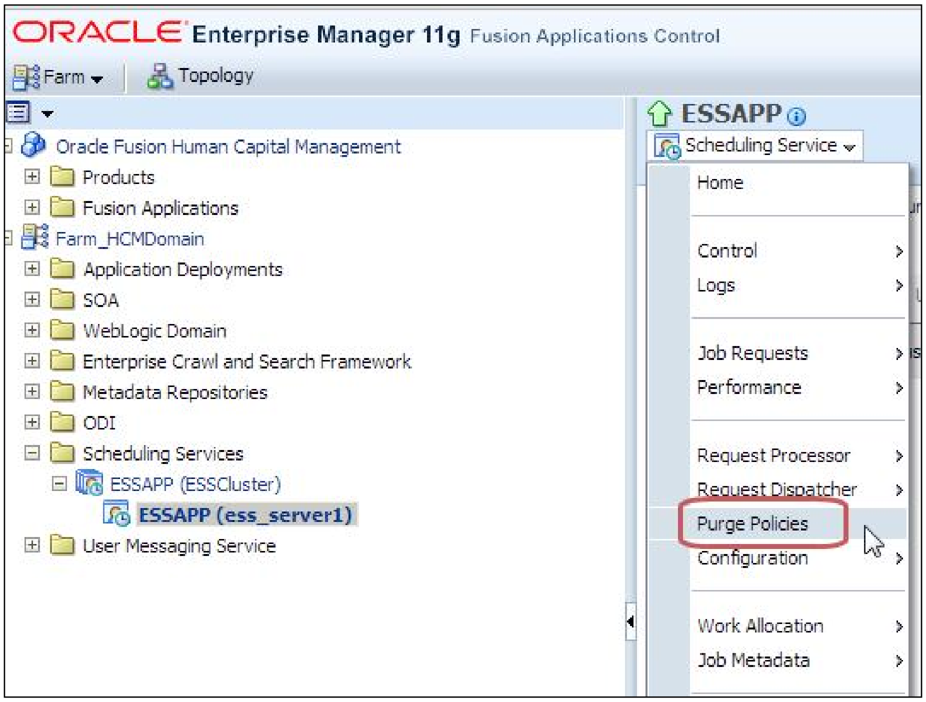
In logical purging, the jobs are marked as purged but are not physically removed. This happens during the physical purge process. We will review the setup for both the logical and physical purge processes.

Logical purging

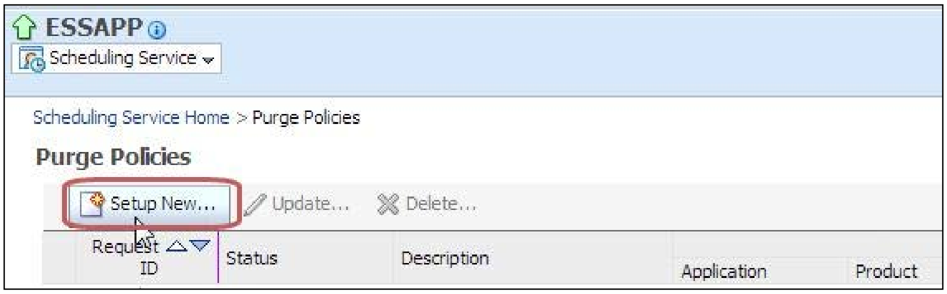
Oracle Fusion Applications provides a flexible approach to purge completed jobs. You can purge requests by job types, job definitions, products, and applications, or for jobs run by a particular user. When you configure the purge policy, it internally calls the **BatchDeleteJob**. The steps to schedule a purge policy are as follows:

1. Log in to Oracle **Fusion Application Control** (`http://hostname:port/em`).

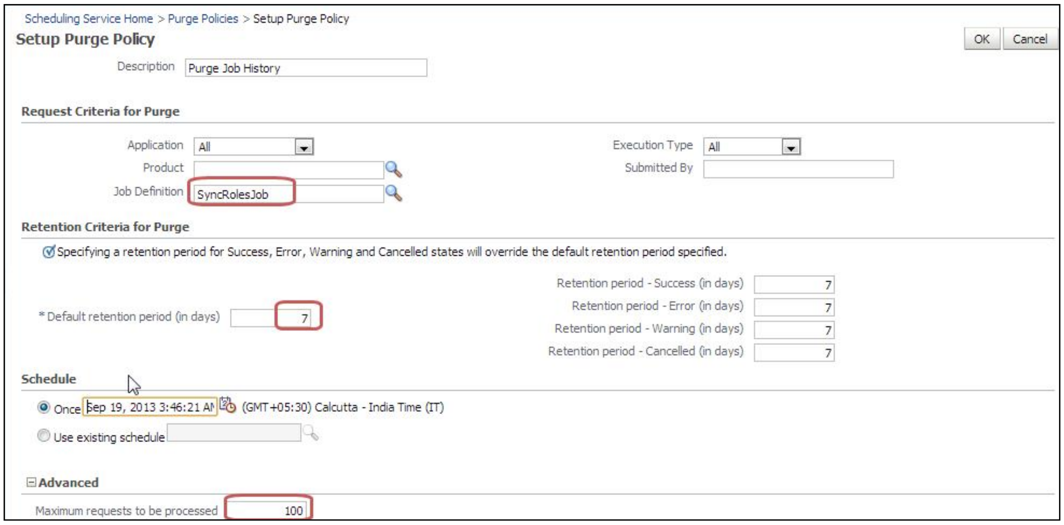
2. Expand the directory tree **Farm_Domainname** | **Expand Scheduling Services**, and click on **ESSAPP**.
3. Right-click on **ESSAPP** and click on **Purge Policies**, shown as follows:



4. Click on **Setup New..., as shown in the following screenshot:**.



5. In this purge policy, the **SyncRolesJob** is being deleted. This job retrieves the latest LDAP changes. In this example, data that is 7 days old is deleted, but it is restricted to a maximum of 100 jobs. Jobs exceeding the first 100 jobs will not be deleted even though they are older than the retention period, as shown in the following screenshot:


Scheduling Service Home > Purge Policies > Setup Purge Policy
OK
Cancel

Setup Purge Policy

Description: Purge Job History

Request Criteria for Purge

Application: All
Product:
Job Definition: SyncRolesJob
Execution Type: All
Submitted By:

Retention Criteria for Purge

Specifying a retention period for Success, Error, Warning and Cancelled states will override the default retention period specified.

* Default retention period (in days): 7
Retention period - Success (in days): 7
Retention period - Error (in days): 7
Retention period - Warning (in days): 7
Retention period - Cancelled (in days): 7

Schedule

● Once Sep 19, 2013 3:46:21 AM (GMT+05:30) Calcutta - India Time (IT)
○ Use existing schedule

⊟ Advanced

Maximum requests to be processed: 100

Physical purging

Although logical purging appears to have removed the old job data, the data is persisted in the backend tables. The logically purged jobs will not show up if you search the job history using the `Search Job Requests` report.

The logically purged jobs are marked as purged, but the rows in the `fusion_ora_ess` schema continue to exist. The BatchDeleteJob will update the *deleted column* in the `request_history` table to `Y`. However, the rows are not physically deleted. This table is based in the `FUSION_TS_TOOLS` tablespace, and over time, this tablespace will continue to grow.

To physically delete the job's history, the procedure `esspurge.purge_requests` needs to be executed. This procedure will physically remove all the data for jobs that have completed and have been marked as logically deleted. The runtime command for this process is:

```
execute esspurge.purge_requests(systimestamp, 100);
```

Troubleshooting ESS

The Fusion Application Administrator console provides many views of the Oracle Enterprise Scheduler. The scheduler's history is always available in the `request_history` table under the `fusion_ora_ess` schema. This table can also be queried to get the status of the scheduler's jobs.

If a job remains in the `WAIT` state, the following checks can be made:

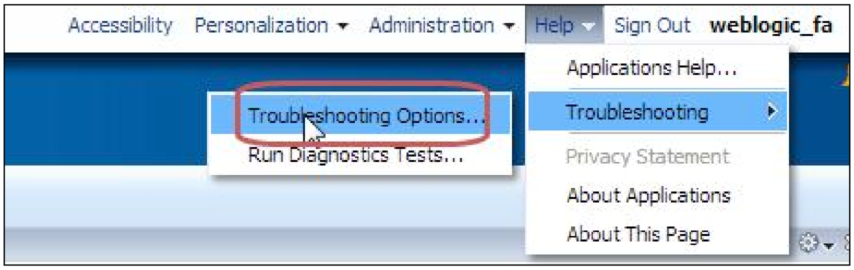
1. Make sure the ESS Managed server is up and running.
2. Make sure the ESS Application instance is up and running.
3. Make sure the Request Dispatcher is up and running.

If a job remains in the `RUNNING` state, the following checks can be made:

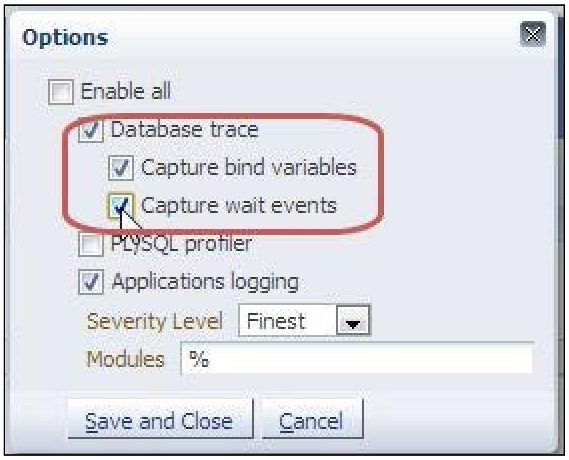
1. Make sure that **Request Processor** is up and running.
2. In case of asynchronous jobs, check on the status of the remote schedule job, and if required, troubleshoot the long-running job.
3. If the remote schedule job is completed, check on its ability to notify the completion status.
4. If the job is running on the database, the performance issues can be traced at the database level.

In some cases, a job may run for a long time and would be seen consuming a large number of CPU cycles. It may also be consuming large amount of memory on the database server. In such cases, it would be possible to troubleshoot the issue by tracing the database session. To trace a database session, perform the following steps:

1. Log in to **Application** (`http://hostname.domain.name:port/homePage`).
2. Navigate to **Help** | **Troubleshooting** | **Troubleshooting Options**, shown as follows:



3. Select **Database trace**, shown as follows:



Options

☐ Enable all

☑ Database trace

☑ Capture bind variables

☑ Capture wait events

☐ PL/SQL profiler

☑ Applications logging

Severity Level: Finest

Modules: %

Once the database trace is enabled, rerun the long-running job. Retrieve the request ID of the job. The database trace will be identified by its `fnd_session_id`. The `fnd_session_id` can be retrieved from the `request_cp` table.

For example, let us trace the job with the request ID `2227`. Log in to the Oracle Fusion Database with the `fusion_ora_ess` schema using the following query:

```
Sqlplus fusion_ora_ess/password
SQL> select fnd_session_id from REQUEST_CP where requestid=2227;
```

The output of the previous command is as follows:

```
FND_SESSION_ID
--------------------------------------------------------------
E6D22A8416306FEDE043EFFC4B0A5001
```

Now we go to the `user dump dest` and look for the trace identifier:

```
E6D22A8416306FEDE043EFFC4B0A5001


[oracle@fadbhost trace]$ pwd
/u01/app/oracle/fadb/diag/rdbms/fadb/fadb/trace
[oracle@fadbhost trace]$ ls -ltr *E6D22A8416306FEDE043EFFC4B0A5001*.trc
-rw-r----- 1 oracle dba  214970 Sep 22 00:51 fadb_ora_28686_
E6D22A8416306FEDE043EFFC4B0A5001.trc
-rw-r----- 1 oracle dba 5422699 Sep 22 00:51 fadb_ora_28653_
E6D22A8416306FEDE043EFFC4B0A5001.trc
```

Multiple trace files can be consolidated with the `trcess` command. The following command accepts the `client_id`, `session_id`, service, action, or module:

```
trcsess output=test.txt clientid=weblogic_fa
*E6D22A8416306FEDE043EFFC4B0A5001*
```

Once the trace files are consolidated, the `tkprof` command can be used to find bottlenecks in the database.

Troubleshoot warning or failed jobs

If a job is in the *warning* or *failed* state, the following checks can be made:

1. Make sure the Content Management server deployed on the `CommonDomain` is running. Since ESS uses the Content Management server for all the jobs' logs and output files, it is important that this server is running.
2. Run the following query to get an error message from the `request_history` table:

   ```
   Sql> select requestid, name, username, submitter, listener,
   state, error_warning_message, error_warning_detail, definition,
   logworkdirectory, outputworkdirectory
   from FUSION_ORA_ESS.request_history
   where requestid=<requestid>;
   ```

   This query will list the current state of the request. The `error_warning_detail` column will provide details of error messages for the failed job.

3. Review the following logfiles to get more information about errors:
   - `$DOMAIN_HOME/servers/ess_server1/logs/apps/ess_server1-diagnostic.log`
   - ESS-managed server-diagnostic logfile

Summary

In this section, we looked at the typical functionality of an enterprise scheduler. We reviewed the architecture of the Oracle Enterprise Scheduler provided by Oracle Fusion Applications. Maintenance aspects for the scheduler as well as troubleshooting of the scheduler were also reviewed.

6

Monitoring and Troubleshooting Oracle Fusion Applications

Once the application is installed and configured, the runtime aspects of the application take over. This includes monitoring the application as well as troubleshooting issues that occur within the application.

In this section, we will look at setting up logging for specific areas of the application. Diagnostic areas for troubleshooting application issues will also be considered. We will also review the Oracle Fusion Applications Cloud Control-based functionality for out of the box monitoring of Oracle Fusion Applications.

Using the Oracle Fusion Applications logging framework

Oracle Fusion Applications allows you to configure logging at the following three levels:

- **Site**: This level is used to enable system-wide logging. This is the highest level of logging and can have an impact on the performance.
- **Product**: This level enables logging at a product-specific level.
- **User**: This level is used to troubleshoot issues at a user level. The impact on the performance at this level of logging is minimal.

Enabling logging

Logging can be enabled in Oracle Fusion Applications using the following two options:

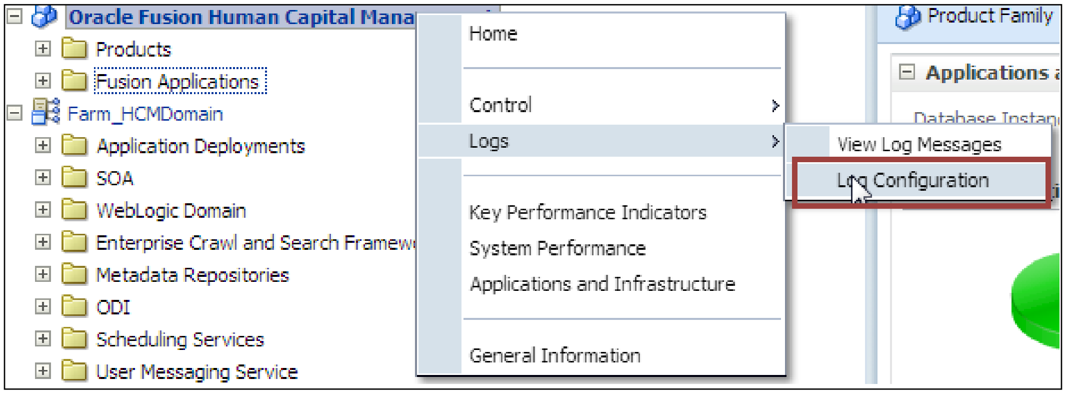
- Fusion Applications Control
- Functional Setup Manager

Using Fusion Applications Control

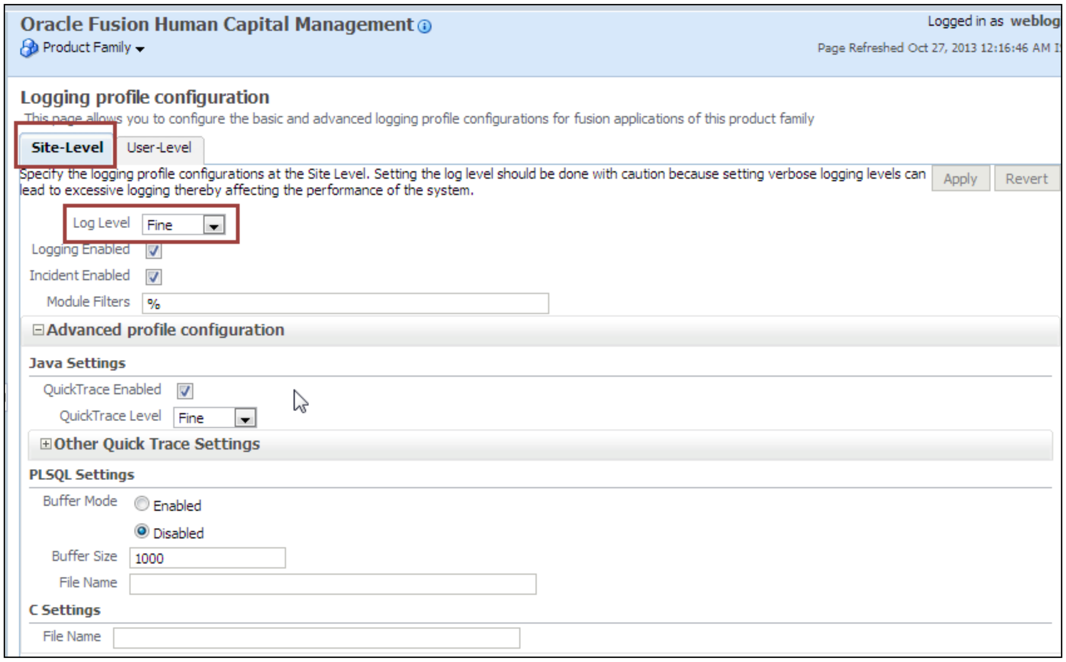
Fusion Applications Control consists of a UI to enable logging. To enable logging, perform the following steps:

1. Log in to Oracle Fusion Applications Control at `http://hostname.domainname.com:port/em`.
2. Navigate to **Product Family Target** | **Logs** | **Log Configuration**.

The following screenshot shows the navigation to the **Log Configuration** section of the console:



The following screenshot depicts the options that are available for logging. It also shows the various options available for each logging level. For example, to set the logging at the site level, click on **SiteLevel**, and set the log level of the subcomponents to the desired value.



Oracle Fusion Human Capital Management

Product Family

Logged in as weblog

Page Refreshed Oct 27, 2013 12:16:46 AM I

Logging profile configuration

This page allows you to configure the basic and advanced logging profile configurations for fusion applications of this product family

Specify the logging profile configurations at the Site Level. Setting the log level should be done with caution because setting verbose logging levels can lead to excessive logging thereby affecting the performance of the system.

Log Level: Fine

Logging Enabled ☑

Incident Enabled ☑

Module Filters: %

⊟ Advanced profile configuration

Java Settings

QuickTrace Enabled ☑

QuickTrace Level: Fine

⊞ Other Quick Trace Settings

PLSQL Settings

Buffer Mode ○ Enabled ● Disabled

Buffer Size: 1000

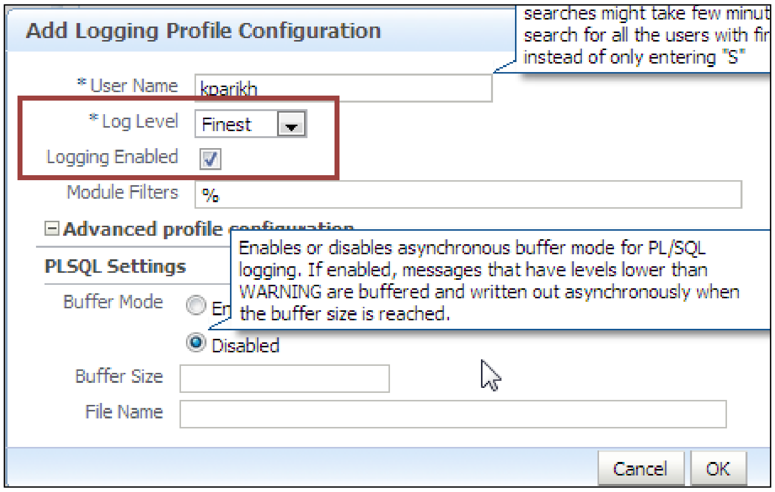
File Name:

C Settings

File Name:

To increase the logging level for the user level, navigate to **User-Level** | **Add**. Search the username and choose the desired **Log Level** and then click on **Logging Enabled**.

In the following screenshot, the logging level has been set to **Finest**. This will provide a detailed log output and will have some impact on the performance.



Add Logging Profile Configuration

searches might take few minut
search for all the users with fi
instead of only entering "S"

* User Name: kparikh

* Log Level: Finest

Logging Enabled ☑

Module Filters: %

⊟ Advanced profile configuration

Enables or disables asynchronous buffer mode for PL/SQL logging. If enabled, messages that have levels lower than WARNING are buffered and written out asynchronously when the buffer size is reached.

PLSQL Settings

Buffer Mode ○ En ● Disabled

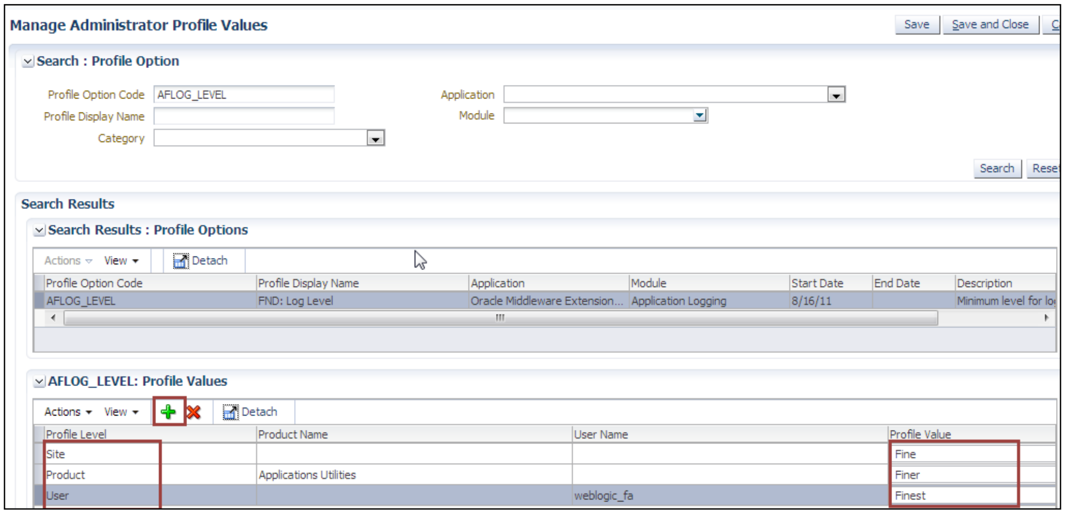
Buffer Size:

File Name:

Using the Functional Setup Manager

The Functional Setup Manager provides capabilities to configure Oracle Fusion Applications from a functional perspective. It also has the capability to set up logging. To use Functional Setup Manager to enable logging, follow these steps:

1. Choose **Setup** and **Maintenance**.
2. Go to **Task Define Profiles** | **Manage Administrator Profile Values** and click on **Go to task**.
3. Search **AFLOG_LEVEL** in the **Profile option code** field.
4. Click on the **+** sign and choose **Profile Level** (**Site**, **Product**, or **User**) and then the desired value for that profile. In the following screenshot, logging has been enabled at the site level to **Fine**, at the product level to **Finer**, and at the user level to **Finest**:



Manage Administrator Profile Values

Search : Profile Option

Profile Option Code: AFLOG_LEVEL
Profile Display Name:
Category:
Application:
Module:

Search Results

Search Results : Profile Options

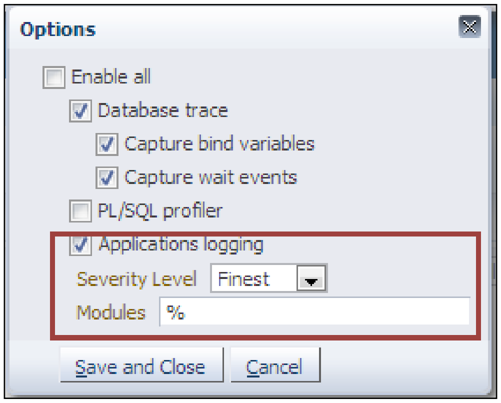
Profile Option Code	Profile Display Name	Application	Module	Start Date	End Date	Description
AFLOG_LEVEL	FND: Log Level	Oracle Middleware Extension...	Application Logging	8/16/11		Minimum level for lo

AFLOG_LEVEL: Profile Values

Profile Level	Product Name	User Name	Profile Value
Site			Fine
Product	Applications Utilities		Finer
User		weblogic_fa	Finest

Users can enable logging for themselves. To do so, they must have the **Supportability Level Manager Duty** option enabled. To set up logging at the user level (after the appropriate access has been granted), follow the given steps:

1. Log in to Fusion Applications.
2. Navigate to **Help** | **Troubleshooting** | **Troubleshooting Options** | **Enable Application Logging** and choose a desired severity level. Through this option, you can also enable **Database trace** and **PL/SQL profiler**, which is helpful in diagnosing performance issues.



Options

☐ Enable all

☑ Database trace

☑ Capture bind variables

☑ Capture wait events

☐ PL/SQL profiler

☑ Applications logging

Severity Level Finest

Modules %

QuickTrace logging

The logging options discussed so far will have some form of impact on performance. Oracle Fusion Applications has a flight recorder kind of functionality, which provides an in-memory option for logging. This form of logging is called **QuickTrace**. With this approach, all the log information is written to a memory buffer, which is recycled in a circular manner.

QuickTrace is usually deployed as a rapid debugging mechanism. If a user is having a performance issue and has enabled user-level logging, QuickTrace can be enabled to identify what else is running at the system level. By taking this approach, the system-wide logging does not need to be enabled.

It is important to note that QuickTrace logs information about all processes and requests running on one WebLogic Server. This logging takes place for a specific time period. Hence, it can get difficult to isolate log entries related to a particular problem.

Configuring QuickTrace

By default, QuickTrace is enabled at the **Fine** level. You can enable/disable QuickTrace by using the **AFLOG_QUICKTRACE_ENABLED** profile, and to increase/decrease the logging level, you can use the **AFLOG_QUICKTRACE_LEVEL** profile options.

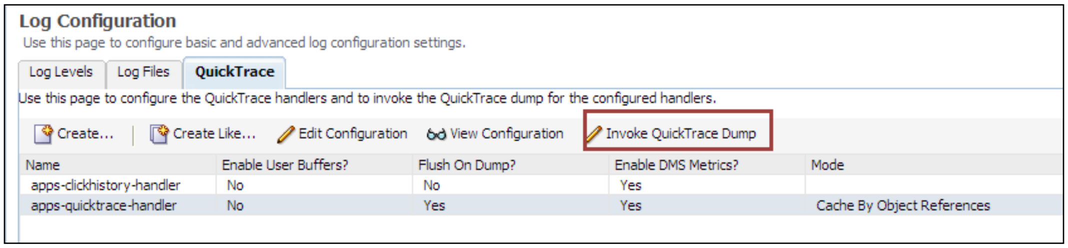
Generating the QuickTrace log

Oracle dumps the QuickTrace log whenever an incident occurs, and the logfile will be a part of the incident directory. An alternate option is to simulate the incident, which is useful in a scenario where you are diagnosing the performance issues.

How to manually dump the QuickTrace log

To manually create a QuickTrace logfile, perform the following steps:

1. Log in to Fusion Applications Control at `http://hostname.domainname.com:port/em`.
2. Expand **Farm_HCMDomain** and choose **WebLogic Server**. Right-click and navigate to **Log Configuration | Choose QuickTrace Handler Name** and click on **Invoke QuickTrace Dump**.



Log Configuration

Use this page to configure basic and advanced log configuration settings.

Use this page to configure the QuickTrace handlers and to invoke the QuickTrace dump for the configured handlers.

Name	Enable User Buffers?	Flush On Dump?	Enable DMS Metrics?	Mode
apps-clickhistory-handler	No	No	Yes	
apps-quicktrace-handler	No	Yes	Yes	Cache By Object References

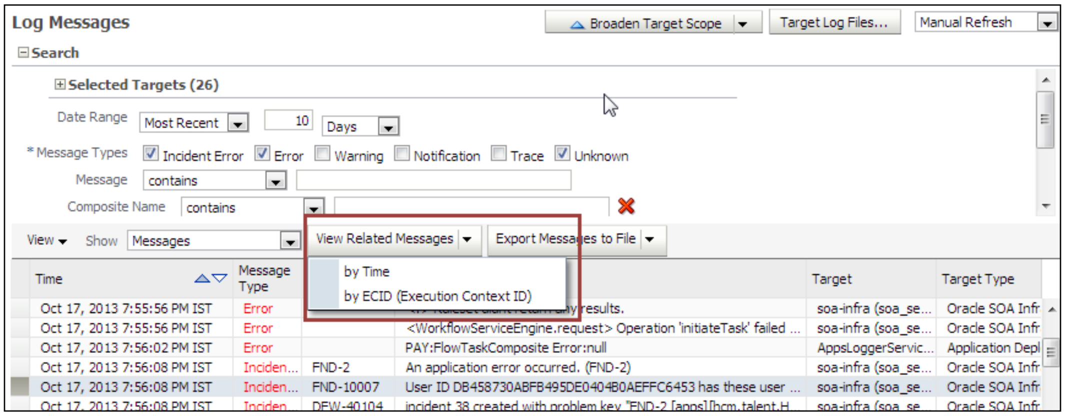
3. The previous step will prompt you to choose **Common Buffer** or **User Buffer (if defined)** and once you click on **Ok**, it will take you automatically to the log message screen to view all the QuickTrace dump messages.

Viewing logfiles

Once logging is enabled, the next step is to find the related information to troubleshoot an issue. Depending on the type of code, Oracle Fusion Applications will write information to their respective logfiles. The following table lists the location of commonly used logfiles:

Component/code type	Logfile location
Java	`${domain.home}/servers/${servername}/logs/apps/${servername}-diagnostic.log`
SOA	`${domain.home}/servers/${servername}/logs/apps/${servername}-diagnostic.log`
ESS	`${domain.home}/servers/${servername}/logs/ess/${servername}-ess.trc`
PL/SQL	Review the value of the `AFLOG_PLSQL_FILENAME` profile
C	Review the value of the `AFLOG_FILENAME` profile
WebLogic Server	`${domain.home}/servers/${servername}/logs/apps/${servername}.log`

Oracle Fusion Applications has a large number of components. Each of these components creates its own log messages. With the plethora of messages and logfiles, it gets difficult to correlate messages. To resolve this, Oracle Fusion Applications Control provides the **View Related Messages** functionality to view the log messages by time or by ECID (ECID is generated for each client request and helps to track all the processing for a particular request). Using this approach, it is possible to retrieve a cogent set of messages related to a particular incident. The following screenshot shows how to view the related messages:



Date Range: Most Recent 10 Days

* Message Types: ☑ Incident Error ☑ Error □ Warning □ Notification □ Trace ☑ Unknown

Message: contains

Composite Name: contains

Time	Message Type			Target	Target Type
Oct 17, 2013 7:55:56 PM IST	Error		results.	soa-infra (soa_se...	Oracle SOA Infr
Oct 17, 2013 7:55:56 PM IST	Error		<WorkflowServiceEngine.request> Operation 'initiateTask' failed ...	soa-infra (soa_se...	Oracle SOA Infr
Oct 17, 2013 7:56:02 PM IST	Error		PAY:FlowTaskComposite Error:null	AppsLoggerServic...	Application Depl
Oct 17, 2013 7:56:08 PM IST	Inciden...	FND-2	An application error occurred. (FND-2)	soa-infra (soa_se...	Oracle SOA Infr
Oct 17, 2013 7:56:08 PM IST	Inciden...	FND-10007	User ID DB458730ABFB495DE0404B0AEFFC6453 has these user ...	soa-infra (soa_se...	Oracle SOA Infr
Oct 17, 2013 7:56:08 PM IST	Inciden...	DFW-40104	incident 38 created with problem key "FND-2 [apps] [hcm.talent.H...	soa-infra (soa_se...	Oracle SOA Infr

Oracle Fusion Applications Control does not display logfiles related to C and PL/SQL components. Additional configuration is required to view files for these components.

The Oracle Fusion Applications Diagnostic Framework

With prior versions of Oracle ERP applications, issue resolution required several back-and-forth interactions with Oracle Support. Once an issue was logged, the support team would request additional logfiles. Based on the analysis of those logfiles, there would be diagnostic steps that would be performed. This would result in additional files being sent back and forth. The whole process was effort intensive and time consuming. Oracle introduced the Oracle Fusion Diagnostic Framework to make issue resolution much more efficient. Using this framework, all information related to an incident is collected and can be sent to Oracle Support for review. Any and all information required to debug this issue is collected by the diagnostic framework.

Oracle Fusion Applications utilizes **Watches and Notification** components that are part of the WebLogic Diagnostic Framework. This framework comes with predefined system diagnostic modules. It also provides the ability to create custom diagnostic components. The foundational components of the diagnostic framework are as follows:

- **Harvester**: This component captures metrics from runtime MBeans and provides a historical record of the metrics
- **Instrumentation**: This component provides a way to embed diagnostic code at defined points inside the application code or server instances
- **Watches and Notification**: This component enables alerts for a particular state of the server based on defined rules (these are called watches) and automates the dispatch of notifications

Predefined watch rule

As a part of the installation process, the standard diagnostic module called Module-FMWDFW is created. It has the following watch rules:

Watch rule	Rule expression
Deadlock	`((SEVERITY = 'Critical') OR (SEVERITY = 'Info')) AND ((MSGID = 'WL-000394') OR (MSGID = 'BEA-000394'))`
StuckThread	`(SEVERITY = 'Error') AND ((MSGID = 'WL-000337') OR (MSGID = 'BEA-000337'))`
UncheckedException	`(SEVERITY = 'Error') AND ((MSGID = 'WL-101020') OR (MSGID = 'WL-101017') OR (MSGID = 'WL-000802') OR (MSGID = 'BEA-101020') OR (MSGID = 'BEA-101017') OR (MSGID = 'BEA-000802'))`

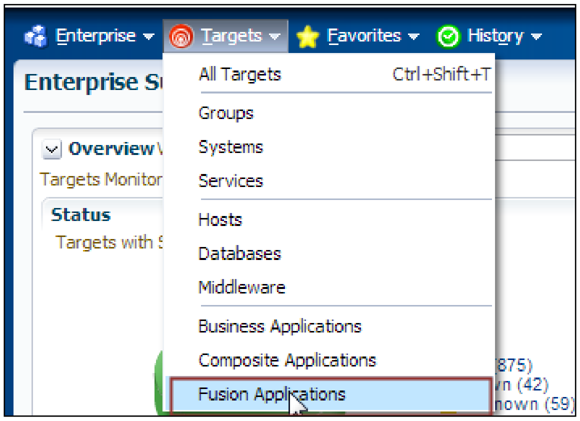
The system diagnostic module file (`Module-FMWDFW.xml`) is stored under the `$DOMAIN_HOME/config/diagnostics` directory. New diagnostic modules can be created, and new watches can be set up using the WebLogic console or the WSLT command line. Multiple diagnostic modules can be created for a domain. However, each target server or cluster can have only one diagnostic module enabled.

Oracle Fusion Applications Cloud Control

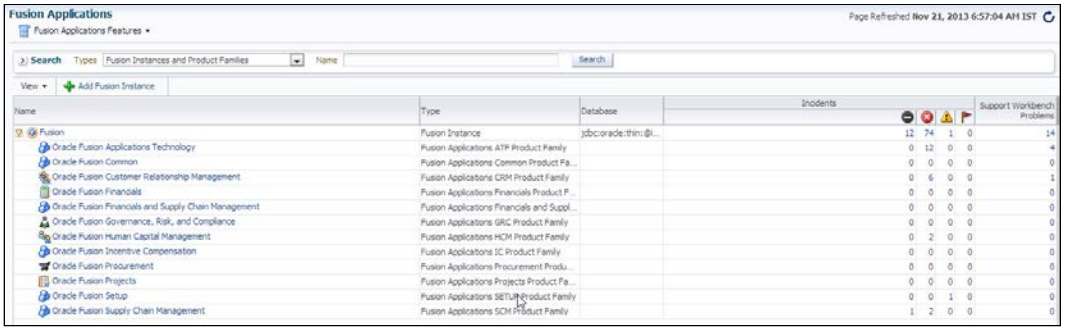
The typical Oracle Fusion Applications environment consists of around five to seven WebLogic domains, and each will have numerous Java EE applications and SOA composites. Monitoring enterprise applications and SOA composites deployed across multiple tiers becomes more complex and time consuming. Another challenge is to track individual transactions and figure out which application consumes more time and resources.

Oracle Enterprise Manager Cloud Control 12c provides a comprehensive management solution to manage and monitor Oracle Fusion Applications. It provides a one stop solution to manage underlying fusion middleware products and WebLogic Server. Cloud control not only allows the monitoring and managing of Oracle Fusion Applications components, but it also helps to configure and manage the life cycle of all its components. It also has comprehensive functionality for node management, which is done using the Enterprise Manager Grid Control components.

You can access the Fusion Applications home page by selecting **Fusion Applications** from the **Targets** drop-down menu in the Enterprise Manager Cloud Control 12c home page, as shown in the following screenshot:



The Fusion Applications home page will provide **Fusion Instance** and product family details. The home page also shows you the **Incidents** details and **Support WorkBench Problems** along with the database connection details. The following screenshot shows the product family listing, and the ability to pick a particular product family for drill down.



Fusion Applications

Fusion Applications Features ▾

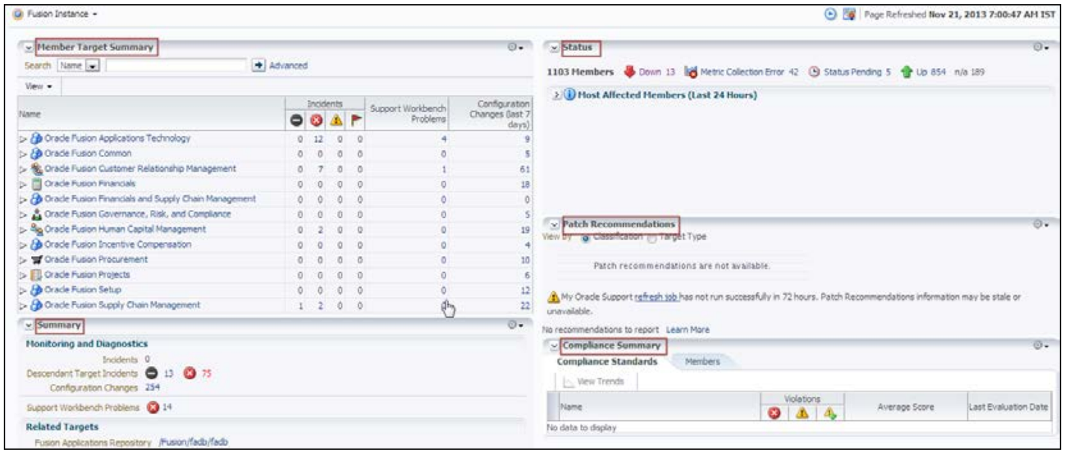
Page Refreshed **Nov 21, 2013 6:57:04 AM IST**

Search Types: Fusion Instances and Product Families Name: Search

View ▾ Add Fusion Instance

Name	Type	Database	Incidents ⊖	⊗	⚠	⚑	Support Workbench Problems
Fusion	Fusion Instance	jdbc:oracle:thin:@i...	12	74	1	0	14
Oracle Fusion Applications Technology	Fusion Applications ATF Product Family		0	12	0	0	4
Oracle Fusion Common	Fusion Applications Common Product Fa...		0	0	0	0	0
Oracle Fusion Customer Relationship Management	Fusion Applications CRM Product Family		0	6	0	0	1
Oracle Fusion Financials	Fusion Applications Financials Product F...		0	0	0	0	0
Oracle Fusion Financials and Supply Chain Management	Fusion Applications Financials and Suppl...		0	0	0	0	0
Oracle Fusion Governance, Risk, and Compliance	Fusion Applications GRC Product Family		0	0	0	0	0
Oracle Fusion Human Capital Management	Fusion Applications HCM Product Family		0	2	0	0	0
Oracle Fusion Incentive Compensation	Fusion Applications IC Product Family		0	0	0	0	0
Oracle Fusion Procurement	Fusion Applications Procurement Produ...		0	0	0	0	0
Oracle Fusion Projects	Fusion Applications Projects Product Fa...		0	0	0	0	0
Oracle Fusion Setup	Fusion Applications SETUP Product Family		0	0	1	0	0
Oracle Fusion Supply Chain Management	Fusion Applications SCM Product Family		1	2	0	0	0

A detailed drill-down view is shown in the following screenshot:



Fusion Instance ▾

Page Refreshed **Nov 21, 2013 7:00:47 AM IST**

Member Target Summary

Search: Name Advanced

View ▾

Name	Incidents ⊖	⊗	⚠	⚑	Support Workbench Problems	Configuration Changes (last 7 days)
Oracle Fusion Applications Technology	0	12	0	0	4	9
Oracle Fusion Common	0	0	0	0	0	5
Oracle Fusion Customer Relationship Management	0	7	0	0	1	61
Oracle Fusion Financials	0	0	0	0	0	18
Oracle Fusion Financials and Supply Chain Management	0	0	0	0	0	0
Oracle Fusion Governance, Risk, and Compliance	0	0	0	0	0	5
Oracle Fusion Human Capital Management	0	2	0	0	0	19
Oracle Fusion Incentive Compensation	0	0	0	0	0	4
Oracle Fusion Procurement	0	0	0	0	0	10
Oracle Fusion Projects	0	0	0	0	0	6
Oracle Fusion Setup	0	0	0	0	0	12
Oracle Fusion Supply Chain Management	1	2	0	0	0	22

Summary

Monitoring and Diagnostics

Incidents 0

Descendant Target Incidents ⊖ 13 ⊗ 75

Configuration Changes 254

Support Workbench Problems ⊗ 14

Related Targets

Fusion Applications Repository /Fusion/fadb/fadb

Status

1103 Members Down 13 Metric Collection Error 42 Status Pending 5 Up 854 n/a 189

Most Affected Members (Last 24 Hours)

Patch Recommendations

View by ● Classification ○ Target Type

Patch recommendations are not available.

⚠ My Oracle Support refresh job has not run successfully in 72 hours. Patch Recommendations information may be stale or unavailable.

No recommendations to report Learn More

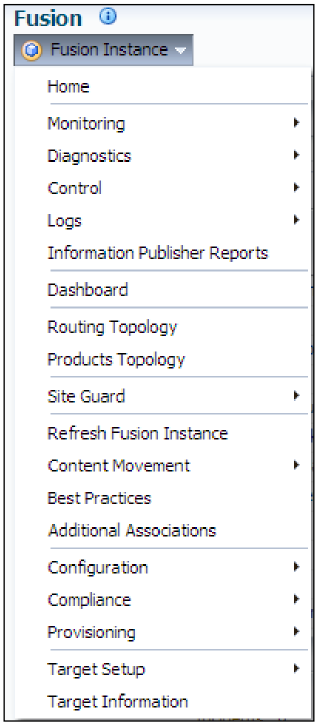
Compliance Summary

Compliance Standards Members

View Trends

Name	Violations ⊗	⚠	⚠	Average Score	Last Evaluation Date
No data to display					

On the Fusion Instance home page, you will get information about the target summary, target status, and summary. It also provides patch recommendation for Fusion Instance and Compliance summary. Using the Fusion Instance drop-down menu, you can further go into the details for **Monitoring**, **Diagnostic**, **Compliance**, **Logs**, and so on. The following screenshot shows the list of options available from the Cloud Control menu:



Using these options, the following monitoring activities can be carried out for Oracle Fusion Applications using OEM Cloud Control:

- Managing Oracle Fusion Applications product families
- Managing Oracle Fusion Middleware components and WebLogic Server
- Diagnosing availability and performance issues
- Proactive middleware monitoring
- Business transaction management
- Reviewing routing and product topology
- Incident management and support workbench
- Viewing log messages and configuring log archive locations
- Analyzing Java EE and SOA applications using Application Dependency and Performance
- Using Java Diagnostic to diagnose the performance problems of Java applications

Summary

In this chapter we discussed the Oracle Fusion Applications Diagnostic Framework. We reviewed the logging options and steps to enable logging. We also looked at QuickTrace and the areas where this functionality can be used. We also discussed the steps to view log messages and to correlate messages. We also discussed the Cloud Control functionality provided by Oracle Enterprise Manager for Oracle Fusion Applications.

7

Managing Fusion Applications

Oracle Fusion Applications have a complex foundation of technology components and functional domains. There are strong dependencies between components. There are maintenance and administrative tasks that need to be performed to manage the application.

In this section we will look at starting and stopping Oracle Fusion Applications, as well as several common administrative functions that need to be performed. The patching process will be looked at, along with the various automation options for patching.

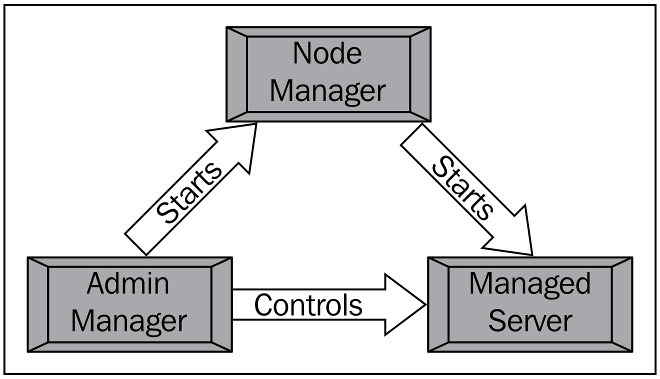
We will look at backup and recovery, as well as the cloning of Oracle Fusion Applications. We will also look at the high-level extension options for Oracle Fusion Applications.

Starting and stopping Oracle Fusion Applications

Oracle Fusion Applications consists of several product components. The primary components that can be stopped and started are:

- Oracle Databases
- Oracle WebLogic Server
- Oracle SOA Suite
- Oracle Identity Management Suite
- Oracle Business Intelligence

Each component can be started and stopped individually. For the WebLogic Server, a combination of Node Manager, Admin Manager, and Managed Server is required to start/stop the server. The high-level relationship between these components is shown in the following figure:



The option available to start/stop each component is listed in the following table:

Component	Options
Database	Oracle Enterprise Manager
	Command line (`svrmgrl`)
	Shell scripts
WebLogic Server	WebLogic Server console
	Command line (`opmnctl`)
	`fastartstop`
SOA Suite	Oracle Enterprise Manager
OIM Suite	Command line
OBIEE Suite	OBIEE Management Suite

Start and stop sequence

There is a specific sequence in which Oracle Fusion Applications must be started/stopped. This sequence is important as there are dependencies between the various technology components of Oracle Fusion Applications. Shutting down or starting up a component in the wrong order will result in errors.

The startup sequence is as follows:

1. Start Oracle Databases for all components
2. Start the OIM Suite
3. Start Node Managers
4. Start WebLogic-Server-based fusion components

The corresponding shutdown sequence is as follows:

1. Stop WebLogic-Server-based fusion components
2. Stop OIM Suite
3. Stop databases

The fastartstop utility

Starting and stopping each component of the Oracle Fusion Applications stack can be time consuming. It requires logging in to different consoles, and using different users and file system locations at the command line. To simplify the process of starting up the Oracle Fusion Applications WebLogic domains, the `fastartstop` utility has been provided.

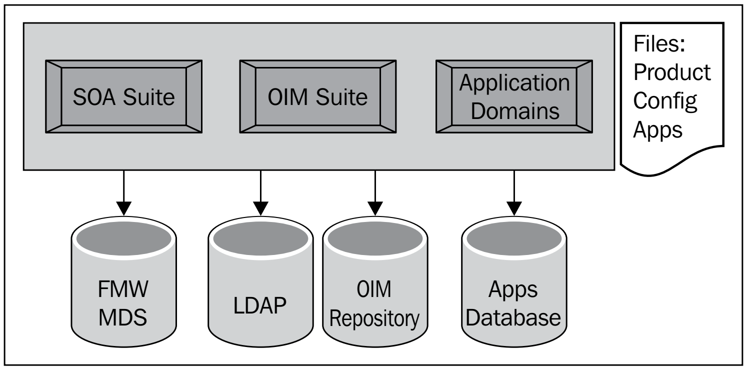
This utility will start/stop all the Oracle Fusion Applications domains, including the Common domain. This is a much more efficient approach for restarting an environment when there are multiple domains. The database and OIM Suite still need to be started separately.

Backing up the Oracle Fusion Applications

All ERP applications require a comprehensive backup and recovery mechanism. The ability to back up an application provides insurance against application failure. This can happen for several reasons:

- Hardware failure
- Database corruption
- Application failure due to patching

A comprehensive backup allows the application to be recovered to a point in time at which the entire application was in an integral state. Since Oracle Fusion Applications have multiple components, the backup of Oracle Fusion Applications spans multiple layers. There are primarily two areas of backup with multiple subcomponents. The high-level layers are shown in the following figure:

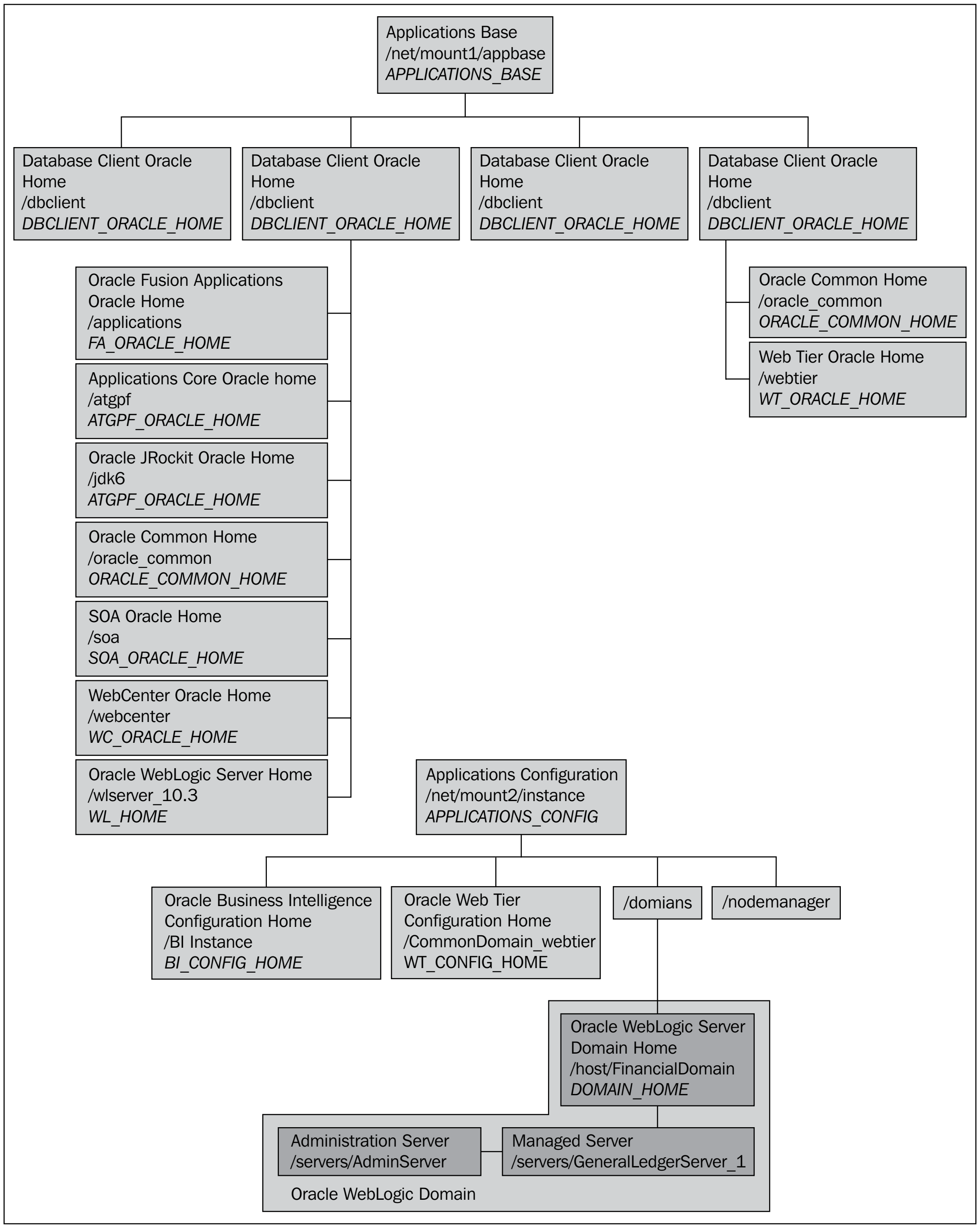


Filesystem backup

The filesystem has a copious amount of files. Some of these files are static and hence a single backup for these files would suffice. Other files change based on the application setup and usage. These files need to be backed up on a regular basis. In general, the following categories of files are backed up:

- **Binary files**: These files are static in nature. They include the application server binaries as well as the application logic components. These files do not change often, and are usually impacted during a major version upgrade.
- **Configuration files**: These files contain configuration information regarding the application. These files change often and should be backed up on a regular basis.

The files themselves include technology stack binaries, application logic files, and application configuration files. These files are stored across servers and nodes and reside in several directory structures. A high-level view (source: *Oracle*) of the file system layout is provided in the following figure:


Applications Base
/net/mount1/appbase
APPLICATIONS_BASE
Database Client Oracle Home
/dbclient
DBCLIENT_ORACLE_HOME
Database Client Oracle Home
/dbclient
DBCLIENT_ORACLE_HOME
Database Client Oracle Home
/dbclient
DBCLIENT_ORACLE_HOME
Database Client Oracle Home
/dbclient
DBCLIENT_ORACLE_HOME
Oracle Fusion Applications Oracle Home
/applications
FA_ORACLE_HOME
Applications Core Oracle home
/atgpf
ATGPF_ORACLE_HOME
Oracle JRockit Oracle Home
/jdk6
ATGPF_ORACLE_HOME
Oracle Common Home
/oracle_common
ORACLE_COMMON_HOME
SOA Oracle Home
/soa
SOA_ORACLE_HOME
WebCenter Oracle Home
/webcenter
WC_ORACLE_HOME
Oracle WebLogic Server Home
/wlserver_10.3
WL_HOME
Oracle Common Home
/oracle_common
ORACLE_COMMON_HOME
Web Tier Oracle Home
/webtier
WT_ORACLE_HOME
Applications Configuration
/net/mount2/instance
APPLICATIONS_CONFIG
Oracle Business Intelligence Configuration Home
/BI Instance
BI_CONFIG_HOME
Oracle Web Tier Configuration Home
/CommonDomain_webtier
WT_CONFIG_HOME
/domians
/nodemanager
Oracle WebLogic Server Domain Home
/host/FinancialDomain
DOMAIN_HOME
Administration Server
/servers/AdminServer
Managed Server
/servers/GeneralLedgerServer_1
Oracle WebLogic Domain

Database backup

The following databases need to be backed up as part of the Oracle Fusion Applications backup process:

- **Oracle FMW MDS repository**: This contains the Fusion-Middleware-specific metadata
- **Oracle LDAP repository**: This is the LDAP database that contains user accounts and related attributes
- **Oracle OIM repository**: This contains the security policies
- **Oracle Application database**: This is the primary database for all transactions and master data

Each of these databases can be backed up using standard database backup tools.

Backup options

Oracle Fusion Applications are intended to be an enterprise grade application. The applications cannot be shut down for extended periods (if at all) for a backup operation. Accordingly, there are two backup options available for Oracle Fusion Applications:

- **Offline backup**: With this approach, the entire application is shut down. This includes the Managed Server as well as the databases. This is a time-consuming approach and is usually chosen to set up a baseline backup. Incremental future backups would use this baseline to update from. This backup is also used when there is a significant upgrade or patching milestone.
- **Online backup**: With this approach, the application is available while the backup is taken. This approach is usually used for incremental and daily operational backups. Incremental database backup techniques are used at the database level for this kind of backup. At the file system level, only file changes would be backed up. An offline backup approach usually requires an initial baseline backup.

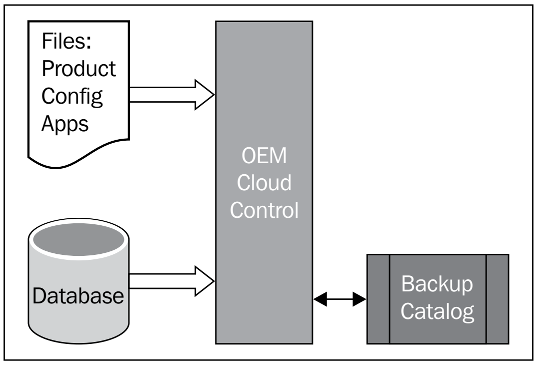
Cloud Control backup

Oracle Fusion Applications contains several database and file systems that need to be backed up. There are several industry-leading options for both backing up databases and file systems. Product vendors have leading technologies to instantly capture changes and record them from a backup perspective.

However, from an Oracle Fusion Applications perspective, several backup sets need to be brought together to create an integral backup that can be recovered. This requires aligning of time stamps and other operational features, and does get complicated from an orchestration perspective.

Oracle provides Oracle Enterprise Manager Cloud Control, which combines the backup orchestration across all components of Oracle Fusion Applications. Using Cloud Control, a single integrated snapshot of the Oracle Fusion Applications can be created at a certain point in time. It provides an extremely powerful approach for managing the backup and recover of Oracle Fusion Applications.

The following figure shows the conceptual approach used by Cloud Control for Oracle Fusion Applications backups:



Component configuration catalog

It is important to keep track of the system configuration variables for the Oracle Fusion Applications environment. This information is vital if the application has to be recovered. It is also useful for ensuring that all backup information for a viable recovery has been captured.

The components of an Oracle Fusion Applications catalog are as follows:

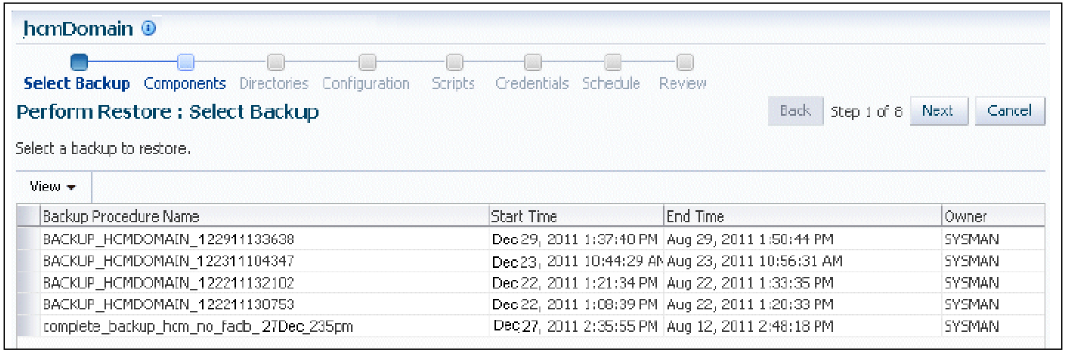
Component	Aspects
Hardware	• Server hardware details • Operating system • Hostname / IP address • Ports used • Patch levels for the hardware
Application	• Version of application and underlying components • Application topology • Access URLs and login information • Filesystem details • Significant patch details • Language and localization details
Database	• Database version • Database name (SID) • Character set
Backup	• Time of backup • Backup location • Backup type

Recovering Oracle Fusion Applications

The recovery of Oracle Fusion Applications requires a recovery at both the database and file system levels. The file systems need to be recovered from the backup device and recreated. The databases need to first be recovered from the backup device and only then can the database-specific recovery be carried out. This includes the recreation of database control files as well as the application of pending archive logfiles.

The Cloud Control based on Oracle Enterprise Manager provides a comprehensive recovery tool for all components of Oracle Fusion Applications. If Cloud Control has been used for backing up Oracle Fusion Applications, a catalog of backups is available.

Using the Cloud Control panel shown in the following screenshot, a specific backup set can be selected and Cloud Control then recovers the entire application based on that backup snapshot. This approach highly simplifies the recovery process and makes it much more efficient:



Perform Restore : Select Backup

Select a backup to restore.

Backup Procedure Name	Start Time	End Time	Owner
BACKUP_HCMDOMAIN_122911133638	Dec 29, 2011 1:37:40 PM	Aug 29, 2011 1:50:44 PM	SYSMAN
BACKUP_HCMDOMAIN_122311104347	Dec 23, 2011 10:44:29 AM	Aug 23, 2011 10:56:31 AM	SYSMAN
BACKUP_HCMDOMAIN_122211132102	Dec 22, 2011 1:21:34 PM	Aug 22, 2011 1:33:35 PM	SYSMAN
BACKUP_HCMDOMAIN_122211130753	Dec 22, 2011 1:08:39 PM	Aug 22, 2011 1:20:33 PM	SYSMAN
complete_backup_hcm_no_facb_27Dec_235pm	Dec 27, 2011 2:35:55 PM	Aug 12, 2011 2:48:18 PM	SYSMAN

Patching

One of the most important and time-consuming aspects of an Oracle Fusion admin's job is applying patches to Oracle Fusion Applications. Patches may be required to resolve problems with the application code, fix production issues, install/add new features, or upgrade Oracle Fusion Applications components. Patching is not a simple, one-step process, but requires careful research in order to determine all of the prerequisite steps, patching steps, and postpatching steps required as part of patching process.

A typical Oracle Fusion Applications patch contains two types of artifacts:

- Middleware artifacts
- Database artifacts

Cloning

Cloning is the process used to create a copy of existing Oracle Fusion Applications; for example, for creating a test or development environment from the production system. This process involves creating a master image of the source environment, copying into the target environment, and rewiring the target environment. The cloning process introduces a new terminology, abstract hostnames, which is basically the source environment hostnames that are present in the file system structure, MDS, and Oracle Fusion Applications configuration files. As part of the cloning process, you would not able to modify abstract hostnames on the target environment. The cloning process only allows you to modify external hostnames that are used by users to access the environment.

Before proceeding with cloning, please consider the following key points:

- Make sure your source environment version supports cloning.
- Review release notes and *Oracle Support Note#1523300. 1* for the list of patches that need to be applied to the source environment.
- Make sure the target environment's operating system and storage configuration are similar to the source environment's.
- Make sure the list of ports used in the source environment is available in the target environment.
- The number of hosts on a target environment should be the same as the source environment's hosts. You can downgrade hardware compared to the source environment if required, but make sure the hardware is configured sufficiently to support the target user base.
- Install JDK on the target environment before extracting source files
- Abstract names / internal hostnames should not be placed in the DNS; these entries should be part of the `/etc/hosts` file.

The following are the high-level steps that need to be performed to create the test environment from production:

- Collect all the relevant information that you have supplied during the provisioning of Oracle Fusion Applications, such as host details, web tier configuration, ports, topology, domain configuration, database details, and IDM configuration details. The Oracle document provides links to the *Discovery Workbook*, which should be utilized to collect all the relevant information required for cloning.
- Take a complete tar backup of the source environment and copy it in the target environment.
- Restore the database on the target server using the RMAN duplicate or any other preferred method.
- Extract the tar backup using the `faclone.sh` file on the target servers and rewire the components with target server details.
- Validate the new clone environment.
- Perform post-clone steps, such as changing passwords, masking data, reintegrating with third-party applications, or other custom applications.

Patching steps

There are several steps involved in patching Oracle Fusion Applications. In this section we will discuss each of these steps:

- Preparing to patch
- Reporting patches
- Validating patches
- Applying patches
- Monitoring and resolving patching issues
- Postpatching steps

Preparing to patch

Before applying a patch, carefully examine the readme file provided with the patch. The readme file will list all the steps required by the patch. It will contain information about prerequired patches or steps, installation steps, and post-installation steps vital to the successful completion of the patch.

A full backup of the application needs to be taken before any patching activities are carried out. Oracle Fusion Applications does not allow any automatic steps to roll back a patch. Before modifying any artifacts, Oracle Fusion Applications does take a backup of file in the `$FA_ORACLE_HOME/admin/pbackup` folder, but it's highly recommended to take a complete backup as a fallback option.

Before you apply a patch or validate a patch, ensure that the snapshot is current for the environment. To take a snapshot, you can use the `adadmin.sh` utility located under the `$FA_ORACLE_HOME/lcm/ad/bin` directory.

You can run the Patch Impact report to understand what the impact of the patch is on your environment. For more information, please review the *Patch reporting* section.

If you are planning to apply multiple patches, please make sure you review the readme files for all the patches and all the mentioned steps have been followed.

Patch reporting

Patch reporting is used to determine whether or not a specific patch has already been applied to your environment, the impact of applying patches on your environment, or a list of installed product families and their version and diagnostic report to view the status of the current patch session. The following table provides description of the syntax:

Syntax	Description
`fapmgr.sh report -patchimpact -patctop <<PatchTopDirectory>>`	This provides information on how the patch impacts on your environment.
`Fapmgr.sh report -listcomps`	This provides the installed product families' version details.
`Fapmgr.sh report -isapplied -patch`	This tells you if the patch has already been applied to your environment or not.
`Fapmgr.sh report -patchprogress`	This provides a diagnostic report of the current patch session. This report is also automatically generated at the end of the patch session.

Validating patches

Validating the patch before applying it provides better visibility on patch compatibility with the current application state. This step is also performed when you apply a patch, but running beforehand provides greater visibility on impact and required downtime and provides a list of steps to take care of to avoid issues when you apply a patch.

The syntax for validating the patch is as follows:

```
fapmgr.sh validate -patchtop <<patchtop Directory>> -online
```

The previous command performs the following high-level steps:

- Validates the OPatch version
- Validates the database connection
- Validates the taxonomy URL
- Checks prerequisite patches
- Validates the impact of applying database artifacts

- Validates the impact of applying middleware artifacts
- In the online mode, it validates prerequisites for deployment and verifies if the Managed Server is down and also if the domain is in edit mode

Applying patches

After understanding the impact and performing all the prerequisite steps, you can begin the process of applying patches. There are two modes to apply a patch:

- **Online mode**: In this mode, the Patch Manager automates the impacted manager server reboot and deploys middleware artifacts

> Online mode will not reboot the impacted server if it's already down during the process of applying the patch.

- **Offline mode**: In this mode, you have to manually reboot the impacted servers and deploy middleware artifacts

The administrative tool used to apply application patches is `fapmgr.sh` (Patch Manager), which internally calls the following three utilities:

- **Adpatch.sh**: This is an AutoPatch utility used to apply database artifacts
- **Adop**: This is an online patching tool that provides a cut-over-based approach and reduces downtime
- **OPatch**: This utility is used to apply middleware artifacts

The following is the syntax for applying a patch:

```
Fapmgr.sh apply -patchtop <<patchtopDirectory>> -online -stoponerror -
Workers 2
```

The following are the high-level steps performed while applying the patch:

- Run the patch validation steps mentioned in the earlier sections
- File comparison
- Back up files that are modified by the patch
- Copy files
- Apply database components
- Deploy middleware artifacts

- Stop and start Managed Server and applications
- Update the patch information to the database (update the ad_bugs and ad_applied_patches tables)
- Run the diagnostic report

The Patch Manager (fapmgr.sh) apply options are as follows:

Option	Purpose
Stoponerror	The patching session fails after a postapply error is reported.
Workers	Specified to increase or decrease the number of workers when running database components.
Loglevel	The Patch Manager's default value is INFO, but if you would like to increase the logging level, you can use this option.
Online	By default, the Patch Manager is in the Offline mode. In the Online mode, the process will take care of starting and stopping of the Managed Server and its automatic deployment.

Monitoring and resolving patching problems

Patching problems manifest themselves in many different ways. There could be issues from the database side or JVM or there could be a performance issue. The first step to diagnose issues is to view the Patch Manager logfiles, and for database artifacts you can monitor the patching progress through the adctrl utility.

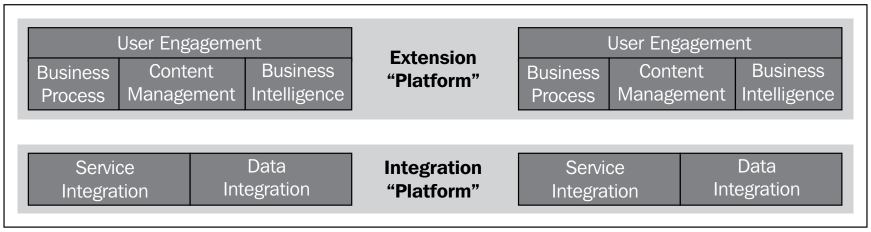
- **Viewing logfiles**: You can find the Patch Manager logfiles under $FA_ORACLE_HOME/ admin/$APPL_TOP/log. If the patch contains middleware artifacts, you can find the OPatch logs under $FA_ORACLE_HOME/cfgtools/opatch. For the database artifacts, you can find the adpatch logs under $ATGPF_ORACLE_HOME/admin/$APPL_TOP/log. You can also find adadmin and worker (adctlr) logfiles under $ATGPF_ORACLE_HOME/admin/$APPL_TOP/log.
- **Using AD Control**: The administrative tool used to manage the adpatch workers is AD Control (adctrl). Workers will frequently fail or hang, which will require the Fusion admin to interface with adctrl.

Post-patching steps

Many patches require postpatching steps to be executed to complete the patching process. If there were any invalid object after postpatching, you can use the adadmin utility to recompile the database objects. In the offline mode, you have to take care of the deployment and start/stop processes for Managed Server or other processes.

Extending Oracle Fusion Applications

Oracle Fusion Applications have been built with the intention of co-existing with the Oracle ERP suite of products as well as integration with other ERP and custom applications. To provide this kind of co-existence, the application has been built with a layered approach. The following figure outlines the extensibility approach that is implicit in Oracle Fusion Applications:



Extension types

There are two aspects of extension and both are provided by Oracle Fusion Applications. The aspects are process extension and application integration.

Process extension

Process extension provides the ability to extend the application to include processes that span multiple applications. The objective here is to provide a seamless end-to-end experience for the end user. The end user may be performing a process that requires functionality provided by different underlying applications (including Oracle Fusion Applications). Oracle Fusion Applications would be able to provide this end-to-end process orchestration without the user realizing that they are spanning multiple applications.

This capability is provided both by the UI elements and the underlying process orchestration components. The Oracle-WebCenter-based frontend allows the extension of Oracle Fusion Applications to include additional UI components that would be required by external applications. The end-to-end process flow is orchestrated by Oracle SOA Suite components.

Data integration

There is comprehensive integration at the data level provided by Oracle SOA Suite. This provides compelling integration functionality to integrate Oracle Fusion Applications with other systems. This provides seamless back-and-forth integration at the backend with multiple applications that would be required for end-to-end processing.

Summary

In this chapter we reviewed the administration and maintenance aspects of Oracle Fusion Applications. The basics of starting and stopping the application were discussed, along with the use of the `fastartstop` utility.

Details of backup and recovery along with various options and approaches were also discussed. All the components involved in the backup and recovery process were reviewed. We also looked at the Component Catalog to ensure that the specifications of an installation were captured.

An in-depth review of patching Oracle Fusion Applications was carried out. This included the patching process as well as pre- and post-patching steps. Finally, the extension of Oracle Fusion Applications and the options available were discussed.

Index

Thank you for buying
Oracle Fusion Applications Administration Essentials

About Packt Publishing

Packt, pronounced 'packed', published its first book "Mastering phpMyAdmin for Effective MySQL Management" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: www.packtpub.com.

About Packt Enterprise

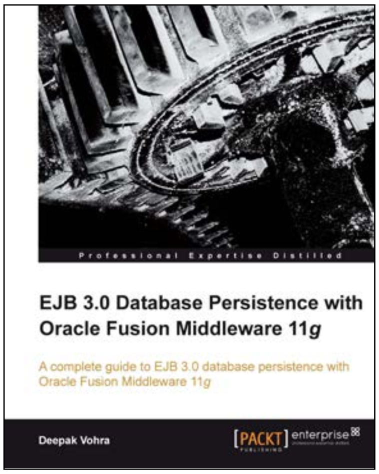
In 2010, Packt launched two new brands, Packt Enterprise and Packt Open Source, in order to continue its focus on specialization. This book is part of the Packt Enterprise brand, home to books published on enterprise software – software created by major vendors, including (but not limited to) IBM, Microsoft and Oracle, often for use in other corporations. Its titles will offer information relevant to a range of users of this software, including administrators, developers, architects, and end users.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to author@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

[PACKT] enterprise
professional expertise distilled
PUBLISHING


Professional Expertise Distilled
EJB 3.0 Database Persistence with Oracle Fusion Middleware 11*g*
A complete guide to EJB 3.0 database persistence with Oracle Fusion Middleware 11*g*
Deepak Vohra
[PACKT] enterprise

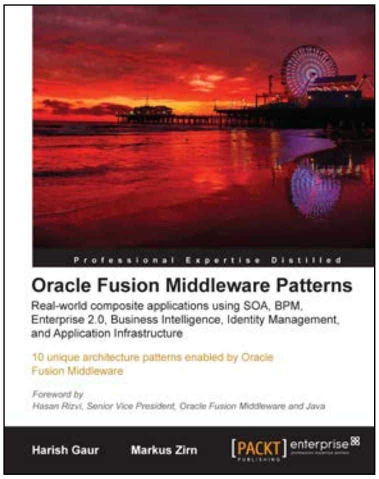
EJB 3.0 Database Persistence with Oracle Fusion Middleware 11*g*

ISBN: 978-1-84968-156-8 Paperback: 448 pages

A complete guide to EJB 3.0 database persistence with Oracle Fusion Middleware 11*g*

1. Integrate EJB 3.0 database persistence with Oracle Fusion Middleware tools: WebLogic Server, JDeveloper, and Enterprise Pack for Eclipse
2. Automatically create EJB 3.0 entity beans from database tables
3. Learn to wrap entity beans with session beans and create EJB 3.0 relationships

---


Professional Expertise Distilled
Oracle Fusion Middleware Patterns
Real-world composite applications using SOA, BPM, Enterprise 2.0, Business Intelligence, Identity Management, and Application Infrastructure
10 unique architecture patterns enabled by Oracle Fusion Middleware
Foreword by
Hasan Rizvi, Senior Vice President, Oracle Fusion Middleware and Java
Harish Gaur
Markus Zirn
[PACKT] enterprise

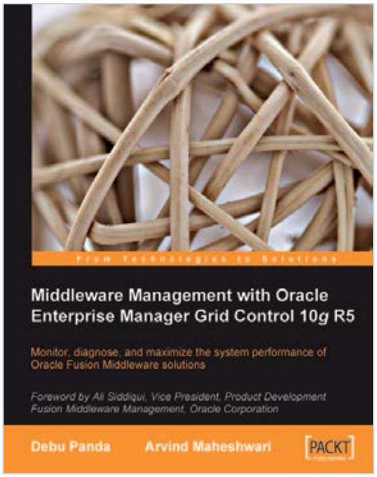
Oracle Fusion Middleware Patterns

ISBN: 978-1-84719-832-7 Paperback: 224 pages

10 unique architecture patterns enabled by Oracle Fusion Middleware

1. First-hand technical solutions utilizing the complete and integrated Oracle Fusion Middleware Suite in hardcopy and e-book formats
2. From-the-trenches experience of leading IT Professionals
3. Learn about application integration and how to combine the integrated tools of the Oracle Fusion Middleware Suite - and do away with thousands of lines of code

Please check **www.PacktPub.com** for information on our titles

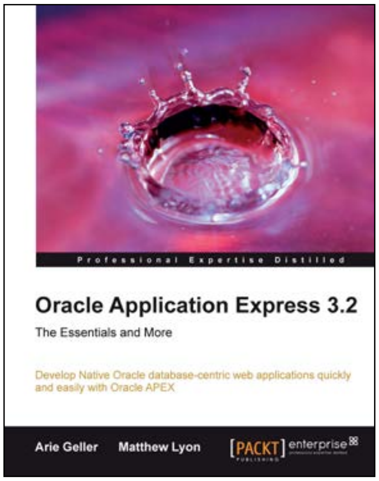

From Technologies to Solutions
Middleware Management with Oracle Enterprise Manager Grid Control 10*g* R5
Monitor, diagnose, and maximize the system performance of Oracle Fusion Middleware solutions
Foreword by Ali Siddiqui, Vice President, Product Development Fusion Middleware Management, Oracle Corporation
Debu Panda
Arvind Maheshwari
[PACKT]

Middleware Management with Oracle Enterprise Manager Grid Control 10*g* R5

ISBN: 978-1-84719-834-1 Paperback: 350 pages

Monitor, diagnose, and maximize the system performance of Oracle Fusion Middleware solutions

1. Manage your Oracle Fusion Middleware and non-Oracle middleware applications effectively and efficiently using Oracle Enterprise Manager Grid Conrol
2. Implement proactive monitoring to maximize application performance
3. Best practices and troubleshooting tips to manage your middleware and SOA applications for optimal service levels and reduced down time from Oracle Product Managers


Professional Expertise Distilled
Oracle Application Express 3.2
The Essentials and More
Develop Native Oracle database-centric web applications quickly and easily with Oracle APEX
Arie Geller
Matthew Lyon
[PACKT] enterprise

Oracle Application Express 3.2

ISBN: 978-1-84719-452-7 Paperback: 644 pages

Develop Native Oracle database-centric web applications quickly and easily with Oracle APEX

1. Grasp the principles behind APEX to develop efficient and optimized data-centric native web applications, for the Oracle environment
2. Gain a better understanding of the major principles and building blocks of APEX, such as the IDE and its modules
3. Learn to Globalize APEX applications, including Right-To-Left support

Please check **www.PacktPub.com** for information on our titles