

Enterprise Composite Applications, Applications Integration Architecture

*An Oracle White Paper
December 2008*

Enterprise Composite Applications, Applications Integration Architecture

Executive Overview	3
Oracle Application Integration Architecture.....	3
PROBLEM: Too Many Applications For Completing a Task Delivers Decreased Service Levels	4
SOLUTION: AIA Enterprise Composite Applications	5
CASE STUDY: Master Data management, Customer Match & Fetch Composite Application.....	6
Identifying Enterprise Composite Applications Patterns.....	8
Application Consolidation.....	8
Application Reuse	9
New Application	9
Natively Extended Application.....	10
Building Enterprise Composite Application: comprehensive support For UI Technologies	10
Oracle ADF Enterprise Composite Applications.....	12
Conclusion.....	13

Enterprise Composite Applications, Face of AIA

EXECUTIVE OVERVIEW

Most IT organizations today are grappling with the dual challenges of integrating their portfolio of heterogeneous applications and the costs of training users how to use multiple applications to complete their daily tasks. IT's top priority is to find a sustainable, low risk, low cost solution that enables companies to bring together their best of breed applications and use them as an integrated whole. Oracle's Application Integration Architecture (AIA) was created to fulfill the demand for such a solution. This whitepaper focuses on how AIA Enterprise Composite Applications empower IT to build user interfaces that ultimately enable it to run the business better.

ORACLE APPLICATION INTEGRATION ARCHITECTURE

Oracle AIA is best described as architecture in a box. It not only provides prebuilt integrations between applications in the Oracle portfolio, it also delivers the business process composition framework underneath the prebuilt integrations, thereby enabling users of AIA to extend the prebuilt integrations as well as to create and govern new integrations. This business process composition framework, also known as the Foundation Pack, provides a programming model, a methodology, and application independent objects and services to construct loosely coupled integrations that are flexible, extensible, scalable, low cost and low risk. In the absence of AIA, IT organizations resort to risky investments in brittle, high cost, high maintenance point-to-point integrations. Given that most companies today opt for a best of breed portfolio instead of the monolithic application suites of the past, the only way for such companies to ensure that applications in the portfolio are integrated and play well together is to adopt AIA.

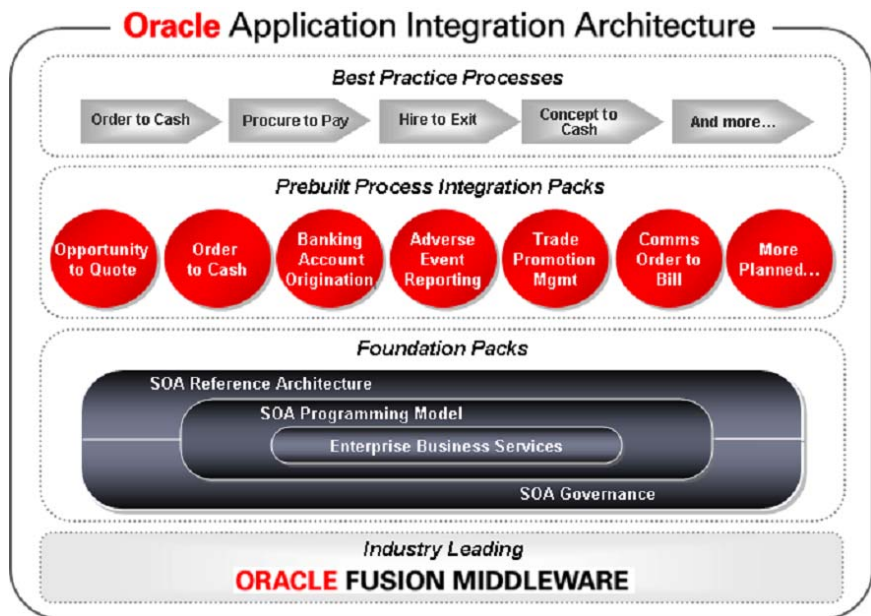


Figure 1: Oracle Applications Integration Architecture

Figure 1 illustrates that the prebuilt integrations offered by Oracle are end to end solutions involving multiple applications. These integrations are based on best practices as defined by the industry the processes represent. The integrations are built using the methodology defined by AIA, leverage the AIA programming model, and ultimately rest upon the application independent Enterprise Business Services to achieve loose coupling.

PROBLEM: TOO MANY APPLICATIONS FOR COMPLETING A TASK DELIVERS DECREASED SERVICE LEVELS

A lack of Composite Applications and the lack of a strategy and infrastructure for building such applications lead to sub-optimal solutions with low user adoption. One common characteristic of such solutions is that UI integrations are typically implemented independently of the associated business process. Even if IT is able to deliver a well-integrated business process, there is little or no improvement from the end-user’s point of view. Here are some examples that illustrate some of the stopgap measures taken:

Swivel Chair Integrations

In the absence of composite applications, IT generally deals with a heterogeneous portfolio by forcing users to master multiple systems. This leads to swivel-chair integrations where a user has to master multiple applications to carry out their daily tasks. This is a costly and risky path because applications are continually being added and dropped from the company’s portfolio, hence requiring the user to constantly learn new tricks to accomplish routine tasks, and new users are continuously joining or leaving the work force.

Intrusive Extensions

Some IT organizations may try to side-step swivel chair integrations by choosing a dominant application and extending it to aggregate the needed functionality in one place. While this is a better approach than a swivel-chair integration, it is still far from optimal because under the covers it tends to be a point-to-point integration with application(s) providing the new content. It requires customization to packaged applications, which can be costly to maintain. In the worst case scenario many applications in the portfolio will need to be augmented the same way, requiring intrusive and point-to-point changes for all affected applications – leading to little or no reuse. Furthermore, as new versions of the dominant application are released, there may not be a straightforward way to migrate the extensions made to prior versions.

Rip-and-Replace

In situations where IT has the option of building a net new application to aggregate and surface content from various applications in the portfolio, once again a point-to-point methodology is typically used which exposes the UI to any changes in the participating applications. These new applications are one-off projects. There is no consistent mechanism for dealing with transactions, security, logging, navigations etc. because there is no underlying infrastructure and framework that provides supports for such features. This is a costly solution, which is difficult to build, upgrade and maintain and it has to be repeated every single time a new application is needed.

SOLUTION: AIA ENTERPRISE COMPOSITE APPLICATIONS

The integration needs for an IT organization are best served by a well defined strategy that ensures there is a consistent, flexible and repeatable approach for building composite applications. These composite applications should be reusable, insulated from changes in provider systems and enable IT to augment existing applications as non-intrusively as possible.

This approach is embodied by AIA Enterprise Composite Applications. At the core of Enterprise Composite Applications (ECAs) are the application independent Enterprise Business Objects and Services. In the integrations developed using AIA architecture, the Enterprise Business Object data model serves as a common data abstraction across systems. This canonical data model is what enables the loose coupling between systems being integrated. Instead of creating point-to-point mapping between schemas of the systems participating in an integration, each system maps to the canonical data model provided by the Enterprise Business Objects. To ensure that the data model can support any-to-any integration and not just Oracle integrations, the Enterprise Business Object definitions are based on industry standards like OAGIS, UNCEFACT. In addition, they are rationalized against dozens of applications owned by Oracle. The result is a set of best-of-breed objects that are built and delivered by AIA. The Enterprise Business Services go hand-in-hand with Enterprise Business Objects. Enterprise Business Services represent application independent web service definition for performing a business

task. These services expose coarse-grained, message-driven interfaces for the purposes of exchanging data between applications.

The Enterprise Business Objects and Services are used to build application independent business processes and it is the same set of objects and services that can be leveraged to create composite applications. AIA's viewpoint is that composite applications and composite processes together make the business process integration and that the best way to achieve business process integration is to use the AIA methodology, objects and services.

AIA Enterprise Composite Applications thus have the following characteristics:

- They are enterprise-class applications that are durable and transactional with support for navigation, security, transactions, languages etc.
- They harness the power of Service Oriented Architecture and hence are loosely coupled and are application independent. As a result they are reusable applications that are future-proofed because there is little or no impact when the provider applications change.
- They provide a standard way of integrating with internal and external source applications through Enterprise Business Services.
- They are open, in that any UI (presentation) technology can be used to build an application on top of Enterprise Business Services.

It is important to note that Enterprise Composite Applications are not portals. Portals generally have a stitched-together feel. The user of ECA is unaware that the content is being provisioned by one or more non-native systems.

CASE STUDY: MASTER DATA MANAGEMENT, CUSTOMER MATCH & FETCH COMPOSITE APPLICATION

A customer owns Oracle E-Business Suite, Siebel CRM as well as the Oracle Universal Customer Master Data Hub. The business need is to support a co-existence MDM model where participating applications E-Business Suite and Siebel-CRM persist a full customer record and periodically synchronize the local record profile to the master record in the Customer Hub. In some cases the participating applications may not even have the golden customer record. This means that the ability for these applications to proactively search the Customer Hub for a specific customer and fetch the profile of that customer in real-time ensures that the most current version of the customer record is used to create or update the local customer record. By contrast, the lack of support for a Match & Fetch flow could result in duplicate customer records being created locally that are out of sync with the customer master and ultimately would need to be reconciled by the Customer Master that contains the golden record.

Enterprise Composite Applications are a great fit for this type of scenario where multiple systems want to augment the existing functionality and retrieve information and display it from another system in essentially the same manner.

Typical point-to-point solution

In this particular example, E-Business Suite and Siebel CRM want to retrieve the golden record from a customer hub in real-time and display the information for the end user. Once the user has selected the golden record, it is retrieved from the hub and synchronized with the local system. Without an ECA for Matching and Fetching the golden record, and an application independent process for synchronizing the record with the local system, applications in all three systems would have to be augmented intrusively. It will be expensive to maintain the augmentation as various applications go through various releases, there will be no reuse and there will also be the additional burden of keeping the intrusive UI extensions in sync with each other. When a new application like PeopleSoft (or another application that came in through an acquisition), wants to take advantage of the same functionality, it will also be forced to a point-to-point integration and also make intrusive changes to the existing application.

Figure 2 illustrates the inferior and brittle point-to-point integration between participating applications and the Customer MDM hub.

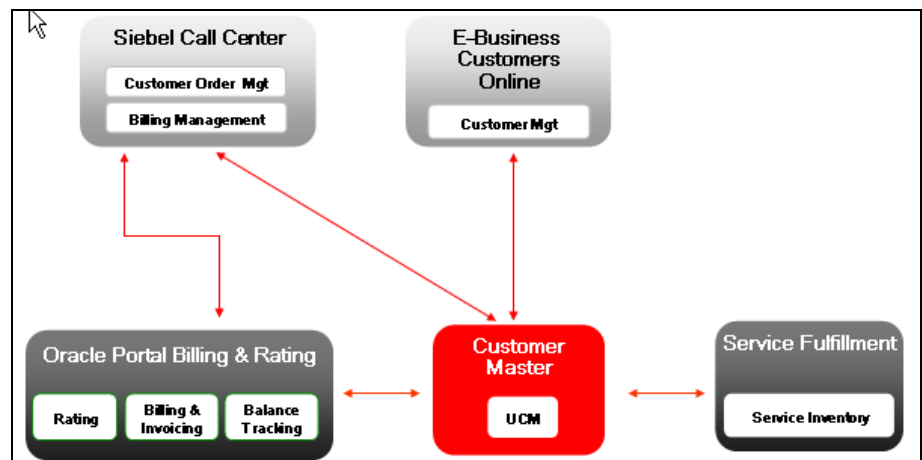


Figure 2: Siebel, E-Business-Suite and the UCM Customer Hub are tightly coupled. There is no abstraction at the process layer or the UI layer. Intrusive UI changes are made to UIs for both applications.

Recommended Solution

The superior approach to address this use case is to build an application independent process and a composite application by leveraging the AIA methodology. The resulting integration will use the Customer Enterprise Business Service to provide loose coupling between the participating applications and the hub. There will be an Enterprise Composite Application which uses the CustomerEBS to display the golden record to the user. This same application will be plugged into E-Business Suite and Siebel. In the future, when a new application,

say a PeopleSoft application, wants to leverage the same functionality, the same ECA can be reused. Similarly, the underlying business process for matching and fetching the record from the hub will also use the same Customer EBS. In the future if a different hub needs to be used it can be plugged in without changing the ECA and with little or no impact to the process.

Figure 3 illustrates the loosely coupled SOA integration approach following the AIA methodology. It also depicts how the same Match_&_Fetch Enterprise Composite Application is being used by both Siebel and E-Business Suite.

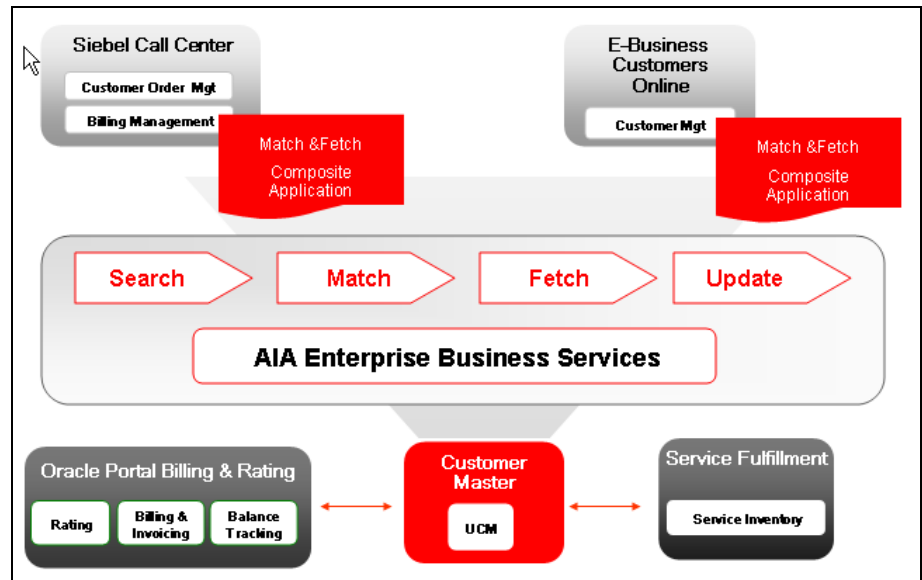


Figure 3: Siebel and E-Business are loosely coupled with the Customer Hub. The Match & Fetch Composite Application is developed on top of AIA and then plugged into Siebel and E-Business suite, ensuring minimal changes to the packaged apps and enabling reuse.

IDENTIFYING ENTERPRISE COMPOSITE APPLICATIONS PATTERNS

This section gives a set of examples of common patterns that will arise when constructing ECAs. Enterprise Composite Applications as defined by AIA cover a broad spectrum of usage patterns. The goal is to support creation of the right type of composite application for the business requirement. The patterns are not mutually exclusive; they can be combined to produce the desired type of composite application. The common underlying theme is application independence through Enterprise Business Services.

Application Consolidation

Multiple applications in the portfolio supply overlapping functionality. The content needs to be aggregated into a single composite application to create a one-stop

shop to enable the end-user to carry out all necessary tasks within the same application and context.

Scenario

A telecommunication vendor needs to create a one-stop shop for ordering DSL, cell phone, land-line and satellite services. The current solution consists of multiple applications, one for each offering. The agent has to master all of the applications to create an order. There is functionality that is common to all applications and requires the same data to be re-entered in each application. In addition, one or more of these applications has functionality that would not be needed if there were a common application to handle all types of ordering needs. Given that none of these systems is going to be retired, the best solution is to create a cross-application flow that is also application independent and to create a new composite application that combines the necessary functionality into a single UI that an agent can use to create one order. The same set of Enterprise Business Services will be used for building the process as well as the composite application.

Application Reuse

Multiple applications in the portfolio need access to the same information from one or more sources. The applications providing the source content are not static, new applications can be added to the mix and existing ones can be unplugged. A new Enterprise Composite Application is created to surface the needed content. The ECA is then plugged into the requesting applications.

Scenario

Acme Corp. has adopted a co-existence MDM model. Participating applications in the ecosystem continuously synchronize the local customer record profile to the master record in the Customer Hub. It is critical for all participating applications to proactively search the Customer Hub for a specific customer and fetch the profile of that customer in real-time. The lack of support for match and fetch could result in duplicate customer records being created locally that are out of synch with the customer master and would ultimately need to be reconciled. The ideal solution is to create an application independent process for synchronizing data between the participating applications and the hub. In addition, a composite application should be created for viewing data from the hub. This composite application can then be plugged into the current set of participating applications or into any future participating application interested in synchronizing with the hub in real-time.

New Application

The desired functionality is either distributed across multiple existing applications or doesn't exist but can be derived by leveraging the data stores of existing applications. There is no existing application that addresses the business need. The resulting composite application is a first-class citizen that is a peer to existing applications in the portfolio.

Scenario

Acme Corp. has recognized that delivering a high-quality customer experience using a 360 degree view of the customer adds essential strategic value its business. A 360 degree view enables the customer service agent to increase customer satisfaction, and it also significantly increases the agent's ability to up-sell and cross-sell. The necessary information is scattered throughout the ecosystem, but there is no application that the agent can use to get this view. The needed solution is a cross-application flow and a new composite application that consolidates information from multiple sources and also has the ability to add new sources of information as well as retire existing ones, without impact the user experience and the information being surfaced.

Natively Extended Application

There is an obvious dominant application that will not be replaced but needs to be augmented. The changes are intrusive but are seamless from the user's perspective.

Scenario

Acme Corp. has chosen Siebel CRM as the dominant application for the foreseeable future. Acme has two local billing systems. Customer account information from these systems needs to be surfaced in the Siebel CRM application. An application independent process should be created to fetch information from the local billing system. The Siebel CRM application is natively extended and uses the same application independent services to retrieve and display data in the CRM UI.

BUILDING ENTERPRISE COMPOSITE APPLICATION: COMPREHENSIVE SUPPORT FOR UI TECHNOLOGIES

AIA Enterprise Composite Applications can be built using any UI presentation technology, as long as it supports consumption of Web Services – examples include .NET/Visual Studio, Oracle Application Development Framework (ADF), MS Office Business Applications. The key is to use AIA's application independent services to ensure the composite application use an application independent canonical data model provided by AIA. The canonical data model enables reusability and also ensures the UI is insulated from changes to applications sourcing the content. Figure 4 illustrates the high level architecture of Enterprise Composite Applications.

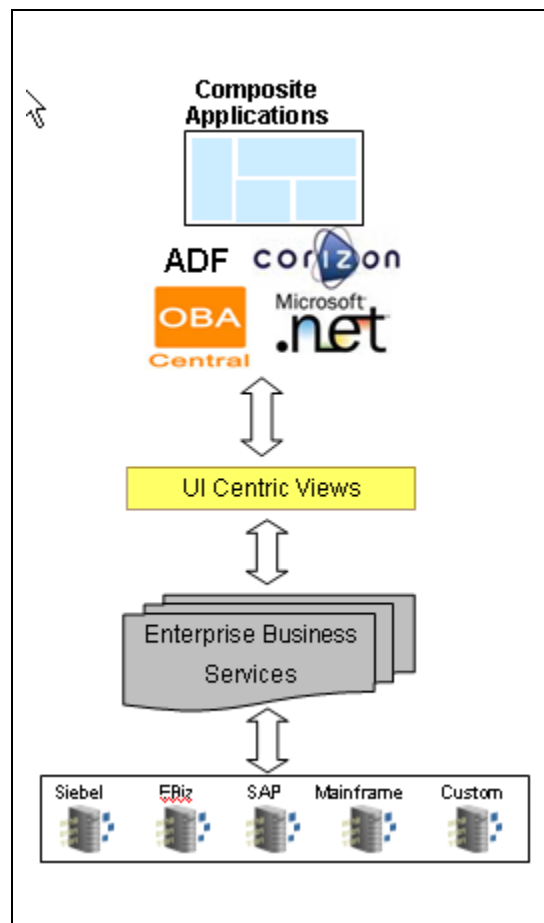


Figure 4: Enterprise Composite Applications

For details on how to build ECAs on .NET and MS-OBA, refer to the whitepaper “Building Enterprise Composite Applications with .NET, MS-OBA”. For more information on Corizon, visit <http://www.corizon.com>.

AIA’s Enterprise Composite Application solution has three key aspects:

- Enterprise Business Services: at the core are the coarse-grained, application independent Enterprise Business Services that furnish the canonical data model for Enterprise Composite Application UI.
- UI Centric Views: The UI technology being used for ECA will create these views and will wire them to the user interface. For example, for Oracle Application Development Framework (ADF) View Objects should be created on top of the Enterprise Business Services.
- Helper Service: AIA provides a Helper Service that automatically takes care of standard tasks that are needed for calling an Enterprise Business Service from an ECA. This service can be called from any UI technology being used to build the ECA.

Oracle ADF Enterprise Composite Applications

AIA provides a JDeveloper extension that is used to build Enterprise Composite Application in Oracle ADF. The extension provides a wizard to build ADF View Objects on top of Enterprise Business Services. The extension derives and displays the View Objects that can be created for the chosen service and also displays the operations available on the service. The developer can then choose which view objects and operations to use in the ECA. The extension automatically creates and maintains the wiring between the ADF artifacts and the Enterprise Business Service. It also provides a standard interface for calling operations on the service. The helper service provided by AIA is used along with the JDeveloper extension to automatically take care of standard tasks that are needed for calling an Enterprise Business Service from an ECA. Hence it enables rapid ECA development on top of Oracle's Fusion Applications Development framework – ADF. Figure 5 illustrates the ECA architecture for Oracle ADF.

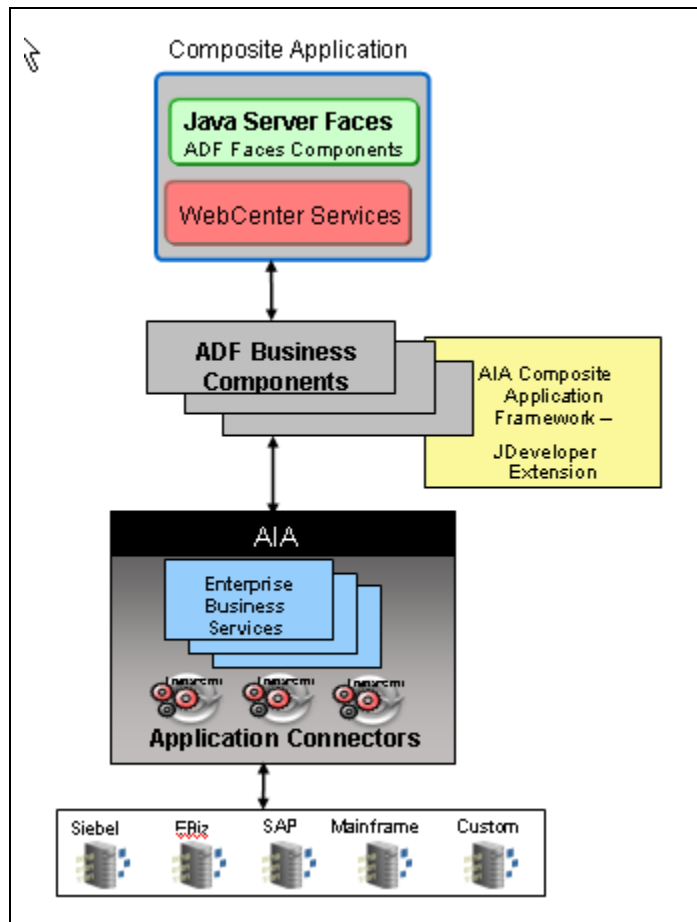


Figure 5: Enterprise Composite Applications on Oracle ADF

The Developer's Guide for how to build AIA Enterprise Composite Applications with Oracle ADF will be available early next year.

CONCLUSION

Oracle AIA offers a solid and proven architecture and methodology for delivering not only prebuilt packaged composite applications but also provides the infrastructure for building new composite applications. It brings SOA to all aspects of business process integration whether they are related to developing application independent, composite processes or user interfaces. ECA enables IT to preserve its chosen UI technology by opening up AIA canonicals for consumption in multiple presentation technologies. ECA also provides a sustainable, repeatable and low-cost, low risk methodology for building application independent applications. IT organizations no longer have to worry about building one-off, throw-away solutions and can offer a framework for augmenting the heterogeneous portfolio through composite applications that enable end-users to carry out their daily tasks in an optimal manner.



Face of AIA: Enterprise Composite Applications

December 2008

Author: Munazza Bukhari

Contributing Authors:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Copyright © 2003, Oracle. All rights reserved.

This document is provided for information purposes only
and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to
any other warranties or conditions, whether expressed orally
or implied in law, including implied warranties and conditions of
merchantability or fitness for a particular purpose. We specifically
disclaim any liability with respect to this document and no
contractual obligations are formed either directly or indirectly
by this document. This document may not be reproduced or
transmitted in any form or by any means, electronic or mechanical,
for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective owners.