

An Oracle White Paper  
October 2009

# Developing AIA Connector Services for SAP

## Introduction

Oracle Application Integration Architecture (AIA) enables the development of compelling industry-specific composite business processes leveraging existing software assets that are available in your IT infrastructure. AIA is a prebuilt, open, and complete integration infrastructure designed for orchestrating agile user-centric business processes across enterprise applications – Oracle, third-party, and custom applications.

This whitepaper will discuss how you can use AIA to integrate your Oracle and non-Oracle applications with SAP and build cross-functional business processes. It outlines how to connect to an SAP system using AIA and discusses how leveraging AIA prebuilt enterprise objects and services and a proven application integration methodology help achieve a faster time-to-market with a lower total cost of ownership and risk.

## AIA Approach to Integration

Today, most organizations have IT portfolios that consist of a number of heterogeneous applications that were never designed to work together. A traditional point-to-point integration seems simple and inexpensive at first; however, as more processes are enabled and more applications are added, it quickly becomes a complex web of integrations with no reusability. These point-to-point integrations lead to a costly, rigid, and brittle infrastructure.

In Oracle's approach, all applications are loosely coupled to the standards-based canonical model. This approach lowers maintenance costs by reducing the number of unique integration points and encourages reuse of already built SOA assets, such as enterprise services and objects. This approach results in agility, as you can quickly adjust your business processes to meet changing business needs and plug new applications into the standards-based canonicals. The AIA one-to-many integration approach is sustainable when implemented with consistent application-independent semantics. During an upgrade, it shields applications from the canonical layer and simplifies migration to new application versions.

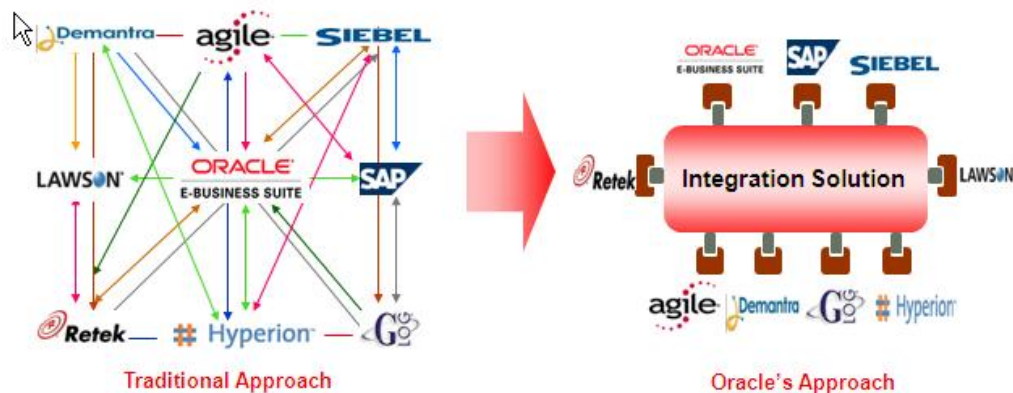


Figure 1: AIA approach to integration

AIA Enterprise Business Objects (EBOs) are application-agnostic, standardized representations of business data. EBOs provide a foundation of common semantics and reusable components

that are rationalized against various applications owned by Oracle. EBOs are based on industry-standards, such as the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), Core Components Technical Specification (CCTS) and the Open Applications Group Integration Specification (OAGIS). Enterprise Business Services (EBSs) provide the web service implementation of the EBOs. EBSs expose coarse-grained, message-driven interfaces for the purposes of exchanging data between applications. EBSs can be bound to various backend systems, but are only loosely coupled to them through Application Business Connector Services (ABCSs).

For more background on how the AIA canonical model fits well with SAP's data models on a semantic level and on how Oracle has built integrations with SAP applications please refer to the "Integrating SAP with Oracle AIA Foundation Pack" white paper.

## Application Business Connector Services

The ABCS is the only application-aware component in AIA. The ABCS exposes the business functions provided by the participating application in a representation that is agreeable to EBSs. ABCSs are necessary because every application has a different representation of objects, and any communication between applications necessitates the transformation of these objects to the canonical definition. The ABCS acts as a conduit to allow the participating application to invoke the application-agnostic EBSs.

Each ABCS is an implementation that is specific to a participating application. The requestor application that wants to leverage an action needs to define a requestor-specific ABCS. The primary task of the requestor ABCS is to act as a vehicle to allow the source application to invoke the EBS. It takes an application business message as input and after performing any optional validations and enrichment, transforms the message into an Enterprise Business Message (EBM) and invokes the EBS.

The primary task of the provider ABCS is to expose the business capabilities that are available in the provider application according to the EBS specification. Application APIs are typically fine-grained and different applications expose different interaction patterns. A provider ABCS takes an EBM as input and calls on a target application to service the business function. For additional details, please refer to AIA Foundation Pack documentation.

## Connecting with SAP

Implementing a business process that integrates with SAP is no different than integrating with any Oracle application. The basic principles and implementation practices are the same whether SAP is a service requestor or provider. The SAP-specific ABCS can connect with an SAP application using the Oracle Application Server (AS) Adapter for SAP using the standard Java Connector Architecture (JCA) and web service interfaces.

The Oracle AS Adapter for SAP Enterprise Resource Planning (ERP) is a remote function call adapter that provides a means to exchange real-time business data between an SAP system and other applications. This adapter can be deployed as both a web service servlet and as a JCA Resource Adapter. The web service servlet deployment is referred to as an Oracle AS Adapter

Business Services Engine (BSE). The JCA Resource Adapter deployment is referred to as an Oracle AS Adapter JCA.

The Oracle AS Adapter for SAP ERP provides XML schemas for the JCA Resource Adapter and the web service for BSE. It also provides Application Explorer, a design-time GUI tool that uses SAP ERP Business Object Repository (BOR) metadata to build XML schemas and web services to handle adapter requests or event data. Application Explorer exposes an adapter service as a JCA Resource Adapter or a web service for BSE in few simple clicks and enables rapid implementation. Users can browse or search for specific intermediate documents (IDocs), Business Application Programming Interfaces (BAPIs), and Remote Function Calls (RFCs).

SAP ERP provides many interfaces, such as RFC modules, BAPIs, and IDocs, which are used to support existing business processes. These business components and methods are available to the Oracle AS Adapter for SAP ERP at the request of SAP ERP and to the event adapter when SAP invokes its remote requests. These business components work as follows:

- BAPIs are interfaces within the business framework that are used to link SAP components to one another or to third-party components.
- RFC modules are SAP application interfaces that enable clients to invoke SAP technologies and receive responses.
- IDocs are the logical messages that correspond to different business processes. They enable different application systems to be linked by a message-based interface. The IDoc type indicates the SAP format to use to transfer the data for a business transaction.

BAPIs and RFCs are called synchronously by the adapter and always return data, either technical error information or a well-formed response document. IDocs are processed asynchronously.

### Benefits of Connecting with a JCA Resource Adapter Configuration

- JCA Resource Adapter deployed in a J2EE server.
- Exposes the adapter as a JCA interaction.
- Exposes the adapter as a service (WSDL/JCA) through the Adapter Framework.
- Metadata is stored in a file-based repository.
- Supports both inbound (event notification) and outbound (request-response) integration with BPEL Process Manager.
- JCA test servlet is used for testing.
- OC4J-RA.xml is used to specify configuration information.
- User name and password can be passed using the connection specification of the Common Client Interface (CCI).

### Benefits of Connecting with a BSE Configuration

- Servlet deployed in a J2EE server.
- Exposes adapter as a web service (WSDL/SOAP).

- Metadata is stored in a repository. Supports both file and database repositories.
- Used for integration with InterConnect.
- Supports only outbound (request-response) integration with BPEL Process Manager.
- BSE configuration page is used to specify configuration information.
- Provides web pages for testing.
- Capability to pass identity using the SOAP header.

As discussed, there are benefits in using both the configurations. At times, the option to choose one configuration over the other is based on how application interfaces required for integration are exposed by the SAP system. AIA discourages developing custom application interfaces for integrations. For performance and scalability, AIA recommends an asynchronous fire-and-forget message exchange pattern and a JCA configuration with IDocs. This may be a better choice for outbound communication. On the other hand, if certain operations need to be processed synchronously, using a BSE configuration calling a particular BAPI may be a better fit.

For example, the AIA Design to Release: Agile PLM to SAP Process Integration Pack (PIP) implementation has both inbound and outbound flows to SAP. To leverage the benefits of both adapter deployment options, we have used the BSE Adapter configuration for inbound requests to SAP and the JCA Adapter configuration for outbound requests from SAP. At the same time, integration processes in other AIA PIPs use a mixed approach.

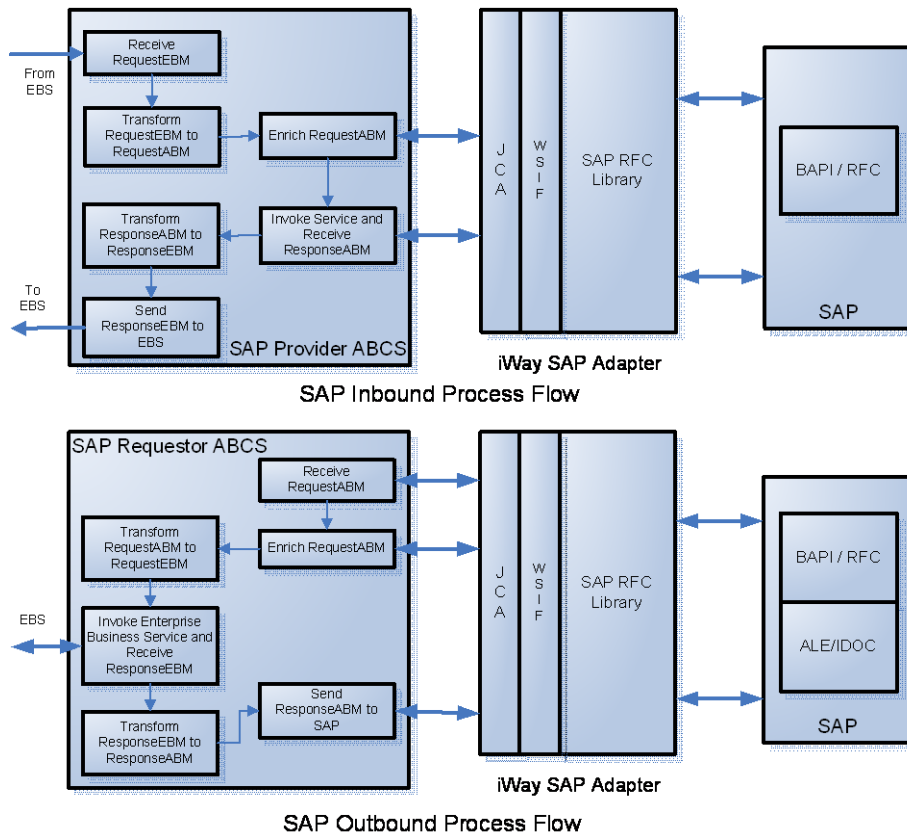
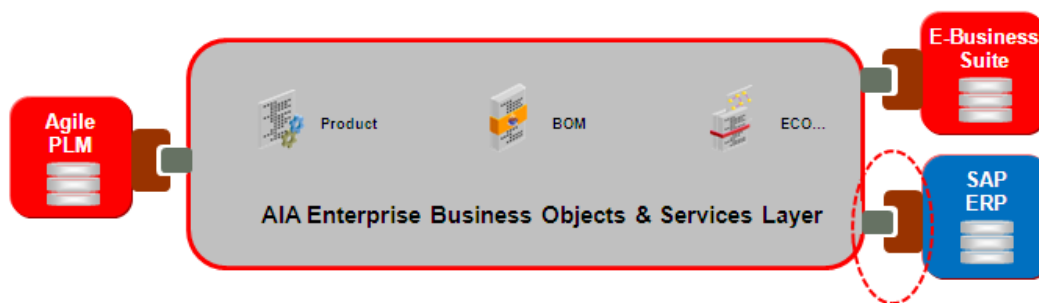


Figure 2: Conceptual view of SAP connector services

For complete details on configuring the Oracle AS Adapter for SAP ERP and creating web services, please refer to the Oracle AS Adapter for SAP ERP user guide.

## SAP Integration Simplified Through Reuse

AIA promotes loose coupling and reusability of services, which facilitates significant savings when developing new integration processes. You can take an existing AIA PIP that integrates Oracle applications and quickly develop integration with SAP by leveraging existing PIP code. The only new development necessary is the creation of an SAP-specific ABCS. This approach enables the reuse of proven business process models, EBSs, EBOs, and the ABCS for one of the participating applications. The functional mapping of attributes from the existing PIP can be leveraged to map EBOs to SAP objects.



**Figure 3: Integration with SAP leveraging AIA prebuilt processes**

The Design to Release: Agile PLM to SAP PIP is a proven example of such reuse. The Design to Release: Agile PLM to E-Business Suite PIP was introduced in AIA 2.2 release. This PIP was leveraged to create the Design to Release: Agile PLM to SAP PIP in AIA 2.3 release with Oracle co-development partner developing the SAP connector services only. This resulted in a 70% savings in development costs and a 50% reduction in time-to-deliver.

## Summary

AIA helps build best-of-breed SOA solutions for customers that are looking to accelerate their SOA implementations by leveraging AIA's prebuilt integrations. The AIA canonical model approach is a proven and scalable integration pattern. It provides a new level of agility and allows customers to plug new applications into standards-based canonicals to quickly change their business processes. Reusing AIA prebuilt integrations to integrate with SAP provides a quicker time-to-market and a greater ROI.



Developing AIA Connector Services For SAP  
October 2009  
Author: Rohit Agarwal

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  
oracle.com



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

Copyright © 2009, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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