

Oracle® Application Integration Architecture
Foundation Pack 11g Release 1 (11.1.1.2.0):
Getting Started with the Oracle® AIA Foundation
Pack and Demo

Release 1 (11.1.1.2.0)

Part No. E17359-01

April 2010

ORACLE®

Part No. E17359-01

Copyright © 2010 Oracle and/or its affiliates. All rights reserved.

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

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

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

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

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Preface	5
Oracle AIA Guides	5
Additional Resources	5
1. Introduction	7
2. Introduction to Application Integration Architecture	9
2.1. Overview of Application Integration Architecture Foundation Pack	12
2.2. Overview of Application Integration Architecture Lifecycle Management	12
2.3. Overview of the Application Integration Architecture Reference Architecture	14
2.3.1. Enterprise Business Objects	14
2.3.2. Enterprise Business Services	14
2.3.3. Enterprise Business Messages	14
2.3.4. Application Business Connector Services	15
2.3.5. Composite Business Processes	16
2.3.6. Enterprise Business Flows	16
3. Introduction to the Application Integration Architecture Foundation Pack Demo	17
3.1. Overview of the Foundation Pack Demo Order Booking Business Process	17
3.1.1. Customer Synchronization	17
3.1.2. Order Processing	18
3.1.3. B2B Order Processing	18
3.2. Overview of the Foundation Pack Demo Architecture	18
4. Creating the Foundation Pack Demo	23
4.1. Modeling the Business Process	23
4.2. Decomposing the Business Process	24
4.3. Creating the Enterprise Business Services and Messages	26
4.4. Creating the Application Business Connector Services	28
4.5. Creating the Composite Business Process	32
4.6. Harvesting the AIA Demo Artifacts	33
4.7. Deploying the Foundation Pack Demo	34
5. Running the Foundation Pack Demo	37
5.1. Initiating the Customer Synchronization Flow	37
5.2. Placing an Order	38

5.3. Orchestrating the B2B Order Flow	40
6. Extending the Foundation Pack Demo.....	43
6.1. Extending an Enterprise Business Object.....	43
6.2. Extending a Process Flow.....	44
6.2.1. Creating an Application Business Connector Service	44
6.2.2. Extending an Enterprise Business Service	45
7. Introduction to Application Integration Architecture Infrastructure Components.....	47
7.1. Overview of the Oracle Enterprise Repository.....	47
7.2. Overview of the Composite Application Validation System	48
7.3. Overview of the Error Handling Framework.....	50
8. Appendix: Foundation Pack Demo Artifacts: Supported Concepts and Features	53
Index	57

Preface

Welcome to the *Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.2.0): Getting Started with the Oracle AIA Foundation Pack and Demo Guide*.

Oracle Application Integration Architecture (AIA) provides the following guides and resources for this release:

Oracle AIA Guides

- *Oracle Application Integration Architecture Foundation Pack: Installation Guide*
- *Oracle Application Integration Architecture Foundation Pack: Getting Started with the Oracle AIA Foundation Pack and Demo*
- *Oracle Application Integration Architecture Foundation Pack: Concepts and Technologies Guide*
- *Oracle Application Integration Architecture Foundation Pack: Development Guide*
- *Oracle Application Integration Architecture Foundation Pack: Infrastructure Components and Utilities Guide*
- *Oracle Application Integration Architecture Foundation Pack: Reference Process Model Guide*
- *Oracle Application Integration Architecture Foundation Pack: Migration Guide for Foundation Pack 2.X to Foundation Pack 11gR1 (11.1.1.2.0)*

Additional Resources

The following resources are also available:

Resource	Location
Oracle Application Integration Architecture Foundation Pack: Product-to-Guide Index	My Oracle Support: https://support.oracle.com/
Known Issues and Workarounds	My Oracle Support: https://support.oracle.com/
Release Notes	Oracle Technology Network: http://www.oracle.com/technology/
Documentation updates	My Oracle Support: https://support.oracle.com/

1. Introduction

"It is apparently easier to say that a firm will adopt SOA than it is to make specific plans and follow through on them."

Randy Heffner, "Planned SOA Usage Grows Faster Than Actual SOA Usage." Forrester, March 2007.

Many organizations plan to adopt service-oriented architectures (SOAs) to achieve more agile and flexible system landscapes. To substantially support them, Oracle offers customers and partners its own extensive experience with application integration and SOA adoption in the form of Oracle Application Integration Architecture (AIA) for reuse in many application process flows.

Oracle AIA delivers a comprehensive, service-oriented foundation for adaptive integration of applications. Oracle's development organization relies on this architecture to implement Process Integration Packs (PIPs) for a number of industry-specific, out-of-the-box integrations between applications in Oracle's portfolio and also third party applications such as SAP. Oracle AIA also offers customers and partners multiple options to adjust existing integrations to unique business requirements or to even build new integrations.

Regardless of the approach, the AIA Foundation Pack substantially supports organizations with a SOA methodology, a well-established common vocabulary, and a rich set of infrastructure components required for a successful SOA-based integration running on Oracle's best-in-class middleware suite.

This document is targeted mainly at integration architects who want to extend or adjust prebuilt integrations shipped by Oracle, or who plan to build new SOA-based integrations based on AIA. After reading this document, you should be able to understand the AIA approach and the reference architecture.

We will demonstrate these concepts and components using the AIA Foundation Pack Demo. This demo is the AIA adaptation of the Oracle SOA Suite 11g Fusion Order Demo, with which many readers may already be familiar. After reading this document, you should also have a solid understanding of the workings of the AIA Foundation Pack Demo and the main AIA concepts and technologies.

2. Introduction to Application Integration Architecture

Today, many organizations are planning to or are already starting to transform their IT landscapes into service-oriented architectures (SOAs) to meet growing business challenges that demand more and more flexibility and agility. They need to be able to respond faster than their competitors to new market demands and need more accurate and real-time information about their customers and their business processes.

However, experience shows that SOA adoption is no easy task. Organizations are facing several key issues. First, business processes are often poorly understood and do not follow best-practice industry approaches. Second, implementation requires a big set of technologies, while often no reference architecture or standard business objects are available to rely on. And finally, implementations often lack a mature methodology and a profound governance model to effectively design, operate, and maintain a SOA.

Oracle's Application Integration Architecture (AIA) delivers many valuable features to master these challenges. AIA is a business-centric approach because it always starts with best-practice industry processes such as Order-to-Cash and Opportunity-to-Quote. From there, Oracle defines a common, application-agnostic vocabulary to uniquely standardize business objects and their relevant abstract operations.

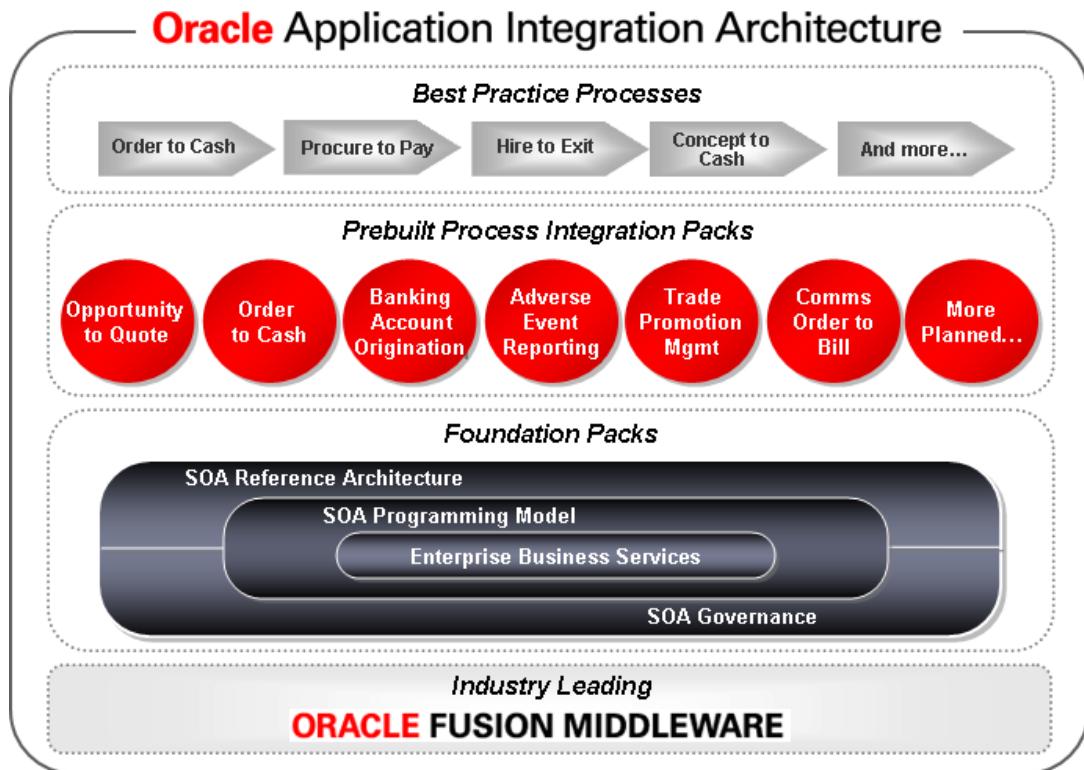


Figure 1: AIA overview

Relying on this foundation of standard industry processes and canonical data models, Oracle identifies Process Integration Packs (PIPs). These PIPs model specific business process flows in the context of a certain industry, Order-to-Bill in the communications industry, for example. These PIPs also contain all required integration aspects in a specific application constellation to cover the business process. For Order-to-Bill, for example, this means the synchronization of customers, products, orders, and other entities between Oracle Siebel CRM and Oracle Billing and Revenue Management.

AIA defines a common vocabulary of common business entities and their corresponding services.

As stated previously, AIA defines a common vocabulary across applications and industries. Enterprise Business Objects (EBOs) are the key elements in this context. They canonically describe standard business entities such as an order or an invoice. Based on these generic business entities, AIA delivers other artifacts such as Enterprise Business Services (EBSs), Enterprise Business Messages (EBMs), Enterprise Business Flows (EBFs), Composite Business Processes (CBPs), and Application Business Connector Services (ABCs). We will describe the purposes of these artifacts and their relationships to each other later when we explain the AIA reference architecture. The combination of all of these SOA artifacts makes up the adaptable and extensible service-oriented reference architecture that AIA stands for.

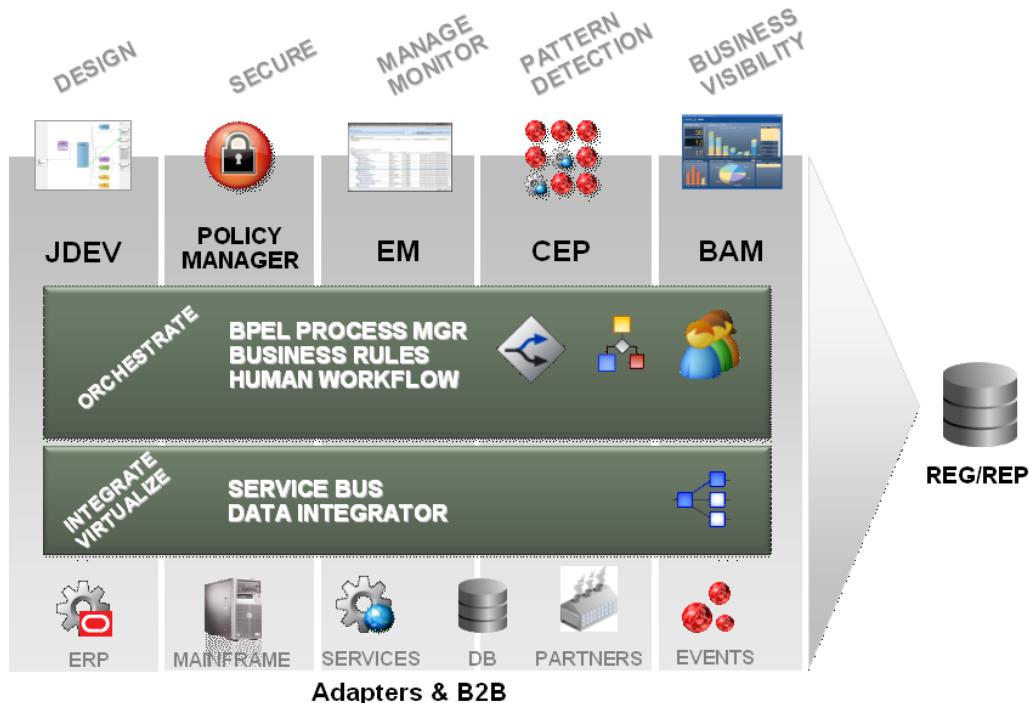


Figure 2: AIA runs on Oracle SOA Suite 11g

The AIA infrastructure enables organizations to successfully manage their SOA.

Based on Oracle Fusion Middleware, AIA provides several additional key infrastructure components that make up an even more powerful SOA stack:

- Installation and deployment

AIA comes with an installation infrastructure that simplifies the deployment of AIA by providing a one-step installation procedure to deploy all AIA components.

- Reference Process Models

The AIA Reference Process Models shipped with AIA Foundation Pack are a rich collection of best-practice business processes modeled with the Oracle Business Process Analysis Suite. These Reference Process Models provide multiple levels of detail to document processes for core processes across business domains and for various industries.

- Oracle Enterprise Repository

The Oracle Enterprise Repository is Oracle's new standard for the management and governance of SOA artifacts. AIA provides additional taxonomies beyond the standard Oracle Enterprise Repository taxonomies to support the AIA artifact types in an optimal way.

- AIA Project Lifecycle Workbench

The new AIA Project Lifecycle Workbench allows integration architects to decompose functional integration requirements into business tasks and identify the AIA artifacts needed for their implementation. The collected information becomes available downstream for the service implementation and for the deployment of the artifacts.

- Service Constructor

Service Constructor is an extension to Oracle JDeveloper 11g, Oracle's design-time development tool for the entire middleware stack. Service Constructor helps jumpstart the development of AIA artifacts, such as ABCSs.

- Composite Application Validation System.

With the Composite Application Validation System (CAVS), customers get a powerful tool to define and test single services, as well as entire end-to-end flows that make up a particular integration flow. By providing simulation capabilities, CAVS enables testing even when participating applications are not in place.

- Error resolution and logging.

Oracle provides a unified error handling and logging solution that enables the identification and resolution of issues across applications and across integration patterns.

- PIP Auditor

The PIP Auditor provides a tool to measure the compliance of AIA artifacts with AIA guidelines. The PIP Auditor can help to improve the quality of development by identifying and visualizing all artifacts that don't follow AIA best practices and programming guidelines.

- Business-to-business support

AIA enables interaction with trading partners by establishing programming guidelines for Business-to-Business (B2B) Connector Services (B2BCS) and by leveraging the Oracle 11g B2B Server for the translation and transport of a variety trading message standards, such as Electronic Data Interchange (EDI).

All of these SOA artifacts and infrastructure components need a solid technical foundation on which to run. With Oracle Fusion Middleware 11g on WebLogic Server, Oracle delivers the best-in-class middleware suite available today. The parts of the Oracle Fusion Middleware that are most relevant to AIA are the BPEL Process Manager, Mediator, Oracle Business Rules, Business Activity Monitoring (BAM), and Policy Manager, as well as the adapters, which are used to connect to virtually any application.

2.1. Overview of Application Integration Architecture Foundation Pack

The AIA Foundation Pack contains everything you need to build custom SOA integrations on a solid foundation.

The same foundation that Oracle uses to develop PIPs is now publicly available as the Oracle AIA Foundation Pack. It provides customers and partners with a powerful infrastructure with which to implement strategic SOA-based integrations without the need to reinvent SOA methodologies, common vocabularies, and key infrastructure components. The AIA Foundation Pack offers the flexibility needed to adapt to business changes rapidly and in a governed way.

The AIA Foundation Pack contains exactly what organizations need to develop and maintain SOA-based integrations:

- A proven service-oriented integration approach to implement agile and adaptable business processes.
- A robust, proven, and reliable reference architecture for SOA-based integrations.
- A ready-to-use set of common data models for frequently required business entities and their related services and messages.
- A rich set of additional infrastructure components to enable streamlined design, implementation, operation, maintenance, and governance of SOAs.
- Best-in-class middleware from Oracle to run the best-in-class SOA tools.

The Application Integration Architecture Foundation Pack delivers everything organizations need to implement state-of-the-art integrations to better support their business processes. Leveraging service-oriented concepts along with a sophisticated governance model provides the means to achieve and maintain the agility businesses require today and tomorrow.

2.2. Overview of Application Integration Architecture Lifecycle Management

Application Integration Architecture Lifecycle Management covers the whole SOA lifecycle and all players contributing to the solution.

To provide a streamlined development and maintenance approach for integrations, AIA Foundation Pack provides a rich set of tools to cover the entire lifecycle of an AIA-based

integration, starting with the identification and modeling business requirements to the installation and maintenance of process integrations.

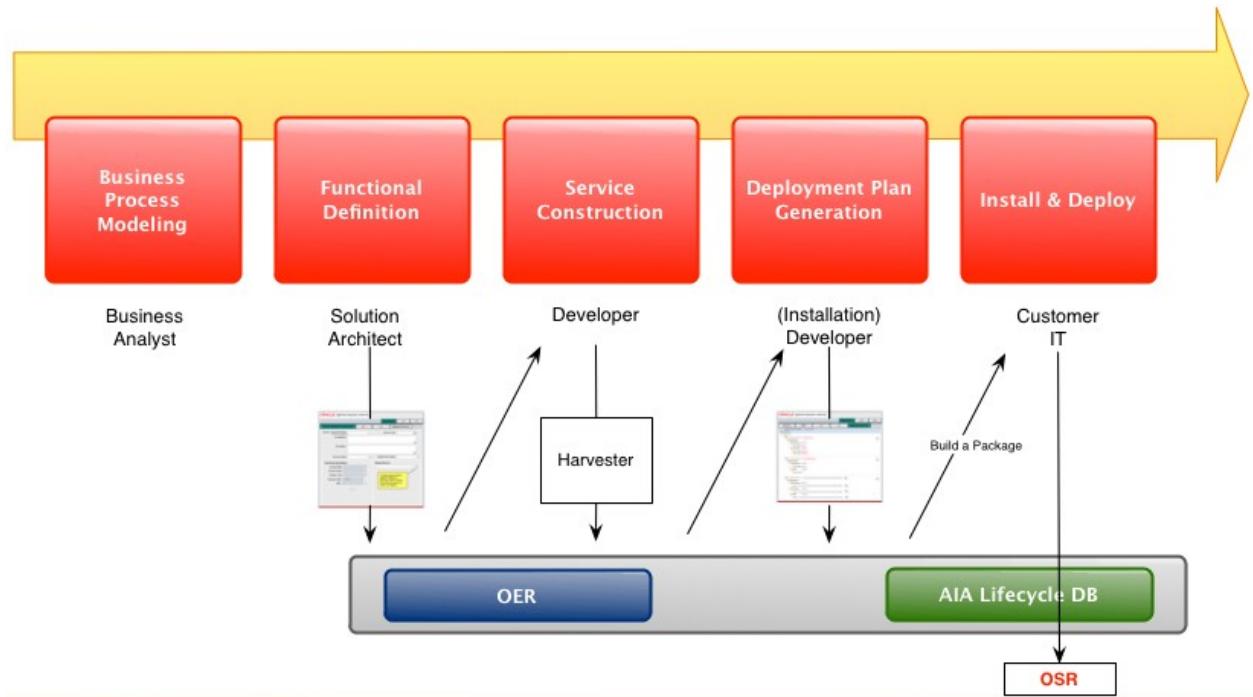


Figure 3: AIA Lifecycle Management

Figure 3 depicts the lifecycle phases from requirements gathering to the deployment of an integration solution. The AIA lifecycle consists of the following phases:

- **Business Process Modeling:** Gathering and modeling business requirements to understand and document the business process.
- **Functional Decomposition:** Decompose the business process into functions and identify reusable and new artifacts required to realize these functions.
- **Service Construction:** Implement the identified new SOA artifacts. Based on best-practice templates so that developer can focus on the specific logic of a service while the template provides the implementation of the standard AIA features such as Error Handling support or enablement for the CAVS testing framework.
- **Deployment Plan Generation:** Create a complete deployment plan based on the service details that were gathered in the Functional Decomposition and Service Construction phases.
- **Install and Deploy:** Use the deployment plan to deploy the complete solution to any Oracle Fusion Middleware environment.

To ensure consistency across all phases, the AIA Lifecycle Management approach ensures that results from each phase are reused in subsequent phases. For example, the outcome of the functional decomposition is accessible through the Service Constructor, enabling developers to not only refer to the information from that phase, but also actually operate on the same set of data and enrich that data for the next phases of the cycle.

We will cover each of these phases in more detail in [Creating the Foundation Pack Demo](#).

2.3. Overview of the Application Integration Architecture Reference Architecture

A proven reference architecture is one of the most important elements of the AIA Foundation Pack. It provides customers with very clear guidance and a mature architectural blueprint by which to build a best-practice SOA.

2.3.1. Enterprise Business Objects

You can think of EBOs as canonical, application-agnostic representations of frequently used business entities.

The architecture starts with the concept of an EBO. EBOs can be considered to be application-independent representations of key business entities. Examples of some of the EBOs included in the AIA Foundation Pack are:

- CustomerParty
- Item
- SalesOrder
- Invoice
- InstalledProduct

Developing the definition of such a common view of business data is always a major challenge in integration projects. Customers now can rely on Oracle's experience with thousands of application implementations when using these EBOs.

2.3.2. Enterprise Business Services

An EBS provides the generic operations that an EBO should have.

EBSs are the centerpiece of the AIA Reference Architecture. They implement the required operations on EBOs in the right coarse-grained granularity that SOAs demand. For every EBO, AIA also ships generic service definitions to cover standard operations such as create, update, query, delete, and sync. Furthermore, an EBS may additionally provide several other business-specific operations, such as GetAccountBalance for the CustomerPartyEBS. Only with these application-agnostic EBSs can the loose-coupling of systems—a necessity in a true SOA—become a reality.

2.3.3. Enterprise Business Messages

An EBM is the message format that is specific to the input or output of an EBS operation.

EBS operations require specific message formats called EBMs for service requests and responses. For instance, the request message for the `QueryCustomerParty` operation requires only a unique customer number. However, the response message needs to carry the entire `CustomerPartyEBO` structure as part of the message because this is what the caller of the service expects as the result when calling `QueryCustomerParty`.

2.3.4. Application Business Connector Services

An ABCS implements a particular service operation in the context of a specific application.

To enable applications to integrate into these generic, application-independent structures, you can implement ABCSs. Such an ABCS calls or is called by the appropriate EBS, depending on whether the application is in the requester or provider role in a particular scenario. The main responsibilities of ABCSs include:

- Conversion between the generic EBO and the application-specific format.
- Cross-referencing of key attributes and message validation.
- Conversation with the specific application.

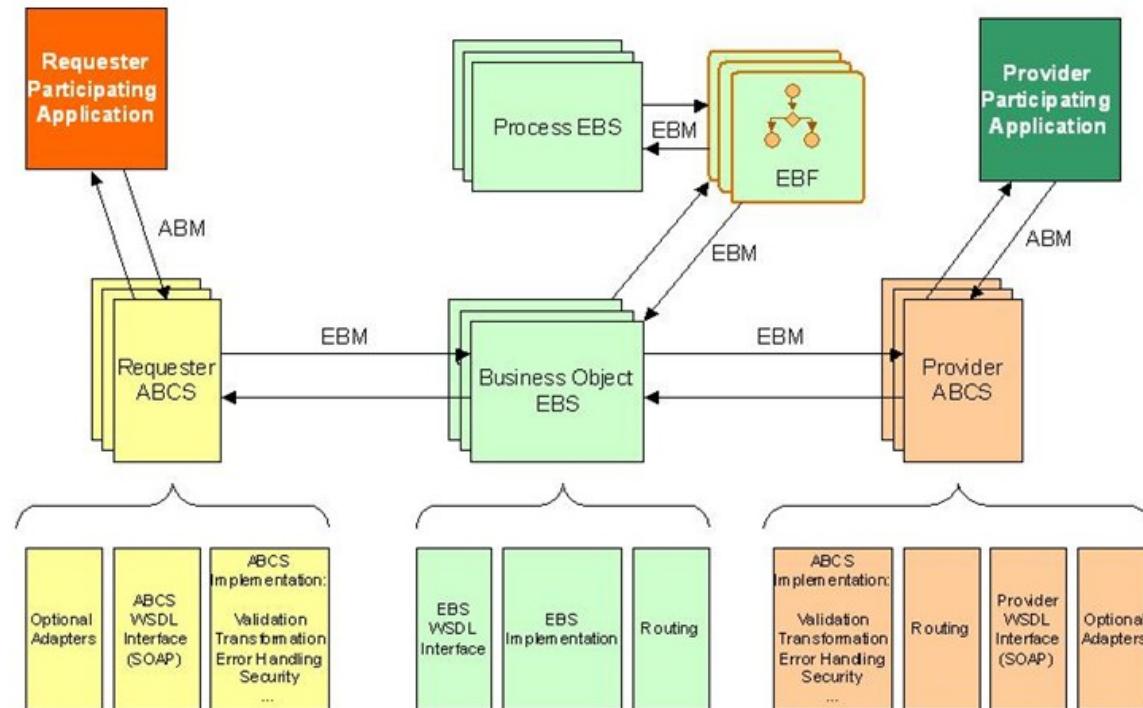


Figure 4: AIA Foundation Pack Reference Architecture

Figure 4 illustrates the way in which these components fit together. While the requesting application talks to the requester ABCS, the requester ABCS transforms the request into an EBM and initiates the EBS. The EBS now calls the provider ABCS to query the response from the provider application. The response finally flows back through the same services to the requester application. On the way back, transformation of the provider's application-specific format to the EBM and back to the requester's format must be performed.

2.3.5. Composite Business Processes

A CBP is a potentially long-running orchestration process eventually leveraging human workflow.

In many cases, there is a need to orchestrate multiple EBSs to implement a business process. As an example, the employee hiring process could be a long-running process with interactions with multiple systems such as a core Human Resources system, a payroll application, and several other systems that need to be set up for every new employee. This process may also include human workflows so that people can perform required action (provide a badge, for example) or approve privileges (provide access to certain systems, for example).

These orchestration processes correlate to a business process in a process model describing the process from a functional view. CBPs typically leverage human workflow and business rules and interact with multiple EBSs in a fully automated manner.

2.3.6. Enterprise Business Flows

An EBF orchestrates a number of EBSs to implement a complex integration flow.

EBFs also orchestrate multiple EBSs to implement a certain business function however, unlike CBPs, EBFs are not long-running processes. They are rather short-lived and correlate to business activities in process models. Therefore, they never leverage human workflow and usually also do not leverage business rules.

Both EBSs and CBPs may invoke EBFs to achieve the overall business process logic. An EBF is the implementation of a reusable business activity requiring some degree of orchestration logic, while a CBP implements the logic of the overall business process, such as the order orchestration in the AIA Foundation Pack Demo.

3. Introduction to the Application Integration Architecture Foundation Pack Demo

The Application Integration Architecture (AIA) Foundation Pack Demo is designed to present the most important components and approaches of Oracle AIA. It is based on the Oracle SOA Suite Fusion Order Demo, which was built to showcase particular aspects of the Oracle SOA Suite. In the context of a simplified business scenario, the Foundation Pack Demo integrates a reasonable number of lightweight applications involved in an order fulfillment process. The AIA Foundation Pack Demo implements the same business flow, but additionally leverages the entire AIA Foundation Pack.

3.1. Overview of the Foundation Pack Demo Order Booking Business Process

The AIA Foundation Pack Demo implements a simplified Order-to-Fulfillment business process in the context of a fictitious retail company called Global Company, which offers electronic equipment in an online shop for their customers.

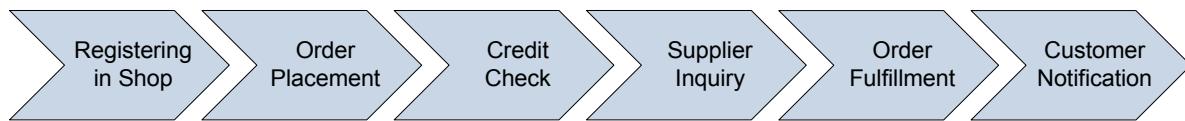


Figure 5: Order-to-Fulfillment Process Diagram

In the following sections, we will describe the processes that need to be supported to support Global Company's business model.

3.1.1. Customer Synchronization

Global Company's customers can register themselves in the provided shop application to be able to purchase items. Each time a new customer account is registered, there is a requirement to propagate this information to Global Company's CRM application. This automated synchronization process is implemented as part of the Foundation Pack Demo to provide a sample for a straightforward integration flow.

Note. While the synchronization is fully functional, users will not be able to log on with these newly generated accounts as this requires propagation to the LDAP-based authentication provider. This propagation is not implemented with the current version of the AIA Foundation Pack Demo.

3.1.2. Order Processing

Customers can select products while browsing through Global Company's offerings in the online shop. After logging in or registering as new customers, they can select electronic items in the shop application. After order submission, the backend system automatically initiates a business flow to process the order through all stages of the flow.

Once the order has been propagated to the Order Management application, it collects basic information about the customer and requests an external credit card check. If the credit card validation is successful, the process queries an internal warehouse and a supplier for prices for the ordered items. The process selects the best price and proceeds to the order fulfillment stage.

For simplicity, the order fulfillment consists of placing a shipment request with a logistics provider and finally notifying the customer about the shipment.

3.1.3. B2B Order Processing

The AIA Foundation Pack Demo also includes a showcase for business-to-business (B2B) integration. The same order processing as previously described in [Order Processing](#) can be initiated by an EDI 850 message containing details for an order from one of Global Company's trading partners.

The only real difference in this B2B scenario is that order status updates are not pushed to the shop application, as these orders don't originate from there. Instead, another interaction with the Oracle B2B server takes place to respond to the trading partner with an EDI 855 message carrying the order status.

3.2. Overview of the Foundation Pack Demo Architecture

The AIA Foundation Pack Demo flow integrates a number of loosely coupled applications. The AIA Foundation Pack Demo flow needs to orchestrate communication between participating applications accordingly to realize the required functionality.

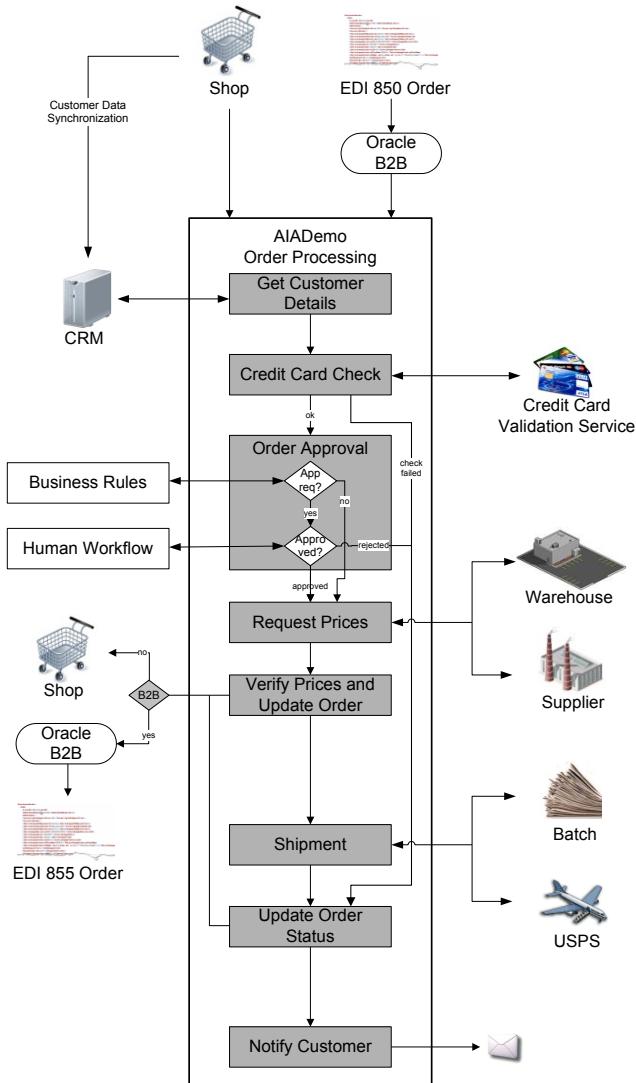


Figure 6: AIA Foundation Pack Demo order processing flow

Figure 6 outlines the business flow in more detail and shows the participating applications and data flows that make up the Order-To-Fulfillment process.

At a more technical level, the following diagram shows the technical details for the Credit Card Check step of the process:

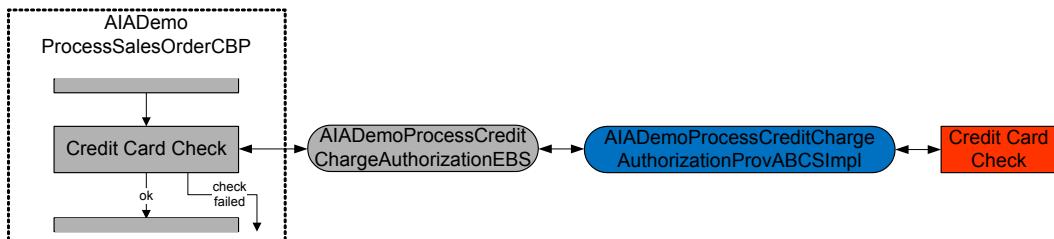


Figure 7: AIA SOA artifacts for the Credit Card Check process step

The Credit Card Check process step, as shown in Figure 7, leverages a service to validate the credit card information provided by Global Company's customer. The AIA Foundation Pack Demo implementation leverages the ReceivedPaymentEBS, as this canonical service supports the ProcessCreditChargeAuthorization method, which suits the AIA Foundation Pack Demo process requirements for verifying credit card details with the CreditCardValidation service.

According to the AIA reference architecture, an Application Business Connector Service (ABCS) is required to establish the communication with the external service, which is a very simple web service implementation, in the case of the AIA Foundation Pack Demo. It is the responsibility of the AIADemoProcessCreditChargeAuthorizationABCSImpl service to transform the incoming ProcessCreditChargeAuthorizationEBM payload into a message structure that is expected by the CreditCardValidation web service.

After invoking this service, the ABCS transforms the service response into the ProcessCreditChargeAuthorizationResponseEBM structure and sends it back to the calling EBS implementation called AIADemoProcessCreditChargeAuthorizationEBS. Finally, the Composite Business Process (CBP) handles the evaluation of the response and proceeds to either process or cancel the order, depending on the outcome of the credit card check.

Interactions with applications in AIA follow this same architectural approach. Figure 8 illustrates the AIA Foundation Pack Demo architecture for the customer synchronization flow, while Figure 9 shows the details of the AIA Foundation Pack Demo's order processing flow.



Figure 8: Demo architecture for artifacts implementing the customer synchronization flow

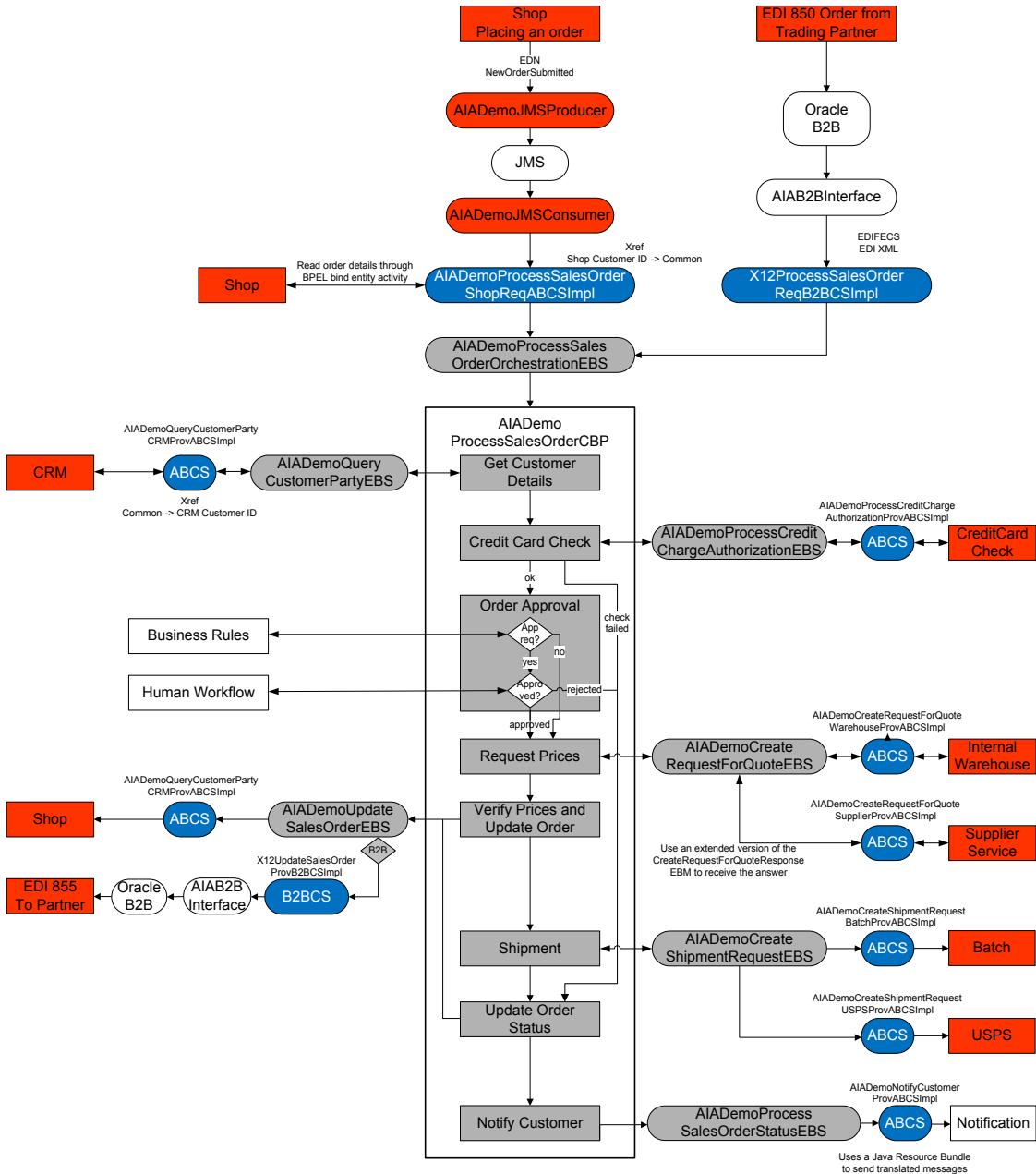


Figure 9: Demo architecture for the artifacts implementing the order processing

4. Creating the Foundation Pack Demo

To implement the Application Integration Architecture (AIA) Foundation Pack Demo, we will follow all of the primary aspects of the AIA reference architecture. In particular, this requires the following development effort:

- Design and implement the Enterprise Business Services (EBSs) based on the following Enterprise Business Objects (EBOs): CustomerPartyEBO, SalesOrderEBO, ReceivedPaymentEBO, RequestForQuoteEBO, and ShipmentRequestEBO.
- Design and implement the Application Business Connector Services (ABCSs) to implement the service operations in the context of the participating applications.
- Design and implement the Composite Business Process (CBP) to orchestrate all of the EBSs to realize the AIA Foundation Pack Demo flow.

The following sections describe the implementation of these AIA artifacts on a conceptual level.

For more information about the AIA methodology and underlying technology, see the *Oracle Application Integration Architecture Foundation Pack: Development Guide*.

4.1. Modeling the Business Process

Always start with modeling the business requirements before thinking about implementation details.

Every integration project starts with a requirements gathering and design phase. This phase is typically handled by business analysts and users who primarily focus on the business process that needs to be supported by an integration. Keep in mind that the sole purpose of application integration is to best support a business process.

The AIA methodology recommends using the Oracle Business Process Analysis Suite for modeling these business processes in a multi-level approach. We start at Level 0, with the identification of impacted functional areas, such as Order Management. We then move to other levels that provide more and more detailed descriptions of the process. At Level 3, we expect to include references to the potential AIA services being required for the implementation.

Figure 10 illustrates Level 3 of the Foundation Pack Demo process model, including high-level implementation details, which include references to the EBSs that need to be leveraged.

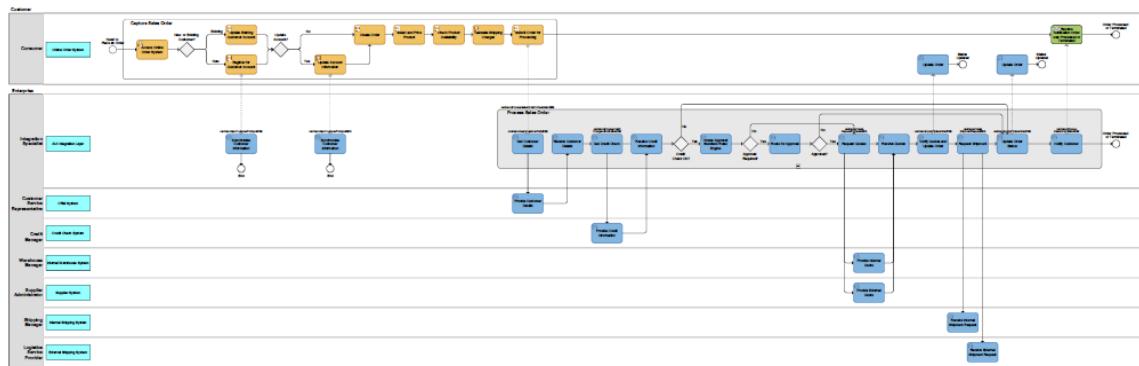


Figure 10: Level 3 process model of the AIA Foundation Pack Demo

4.2. Decomposing the Business Process

Functional decomposition helps map business requirements to the integration artifacts that are required for their implementation.

AIA Foundation Pack now provides the AIA Project Lifecycle Workbench to support the process of decomposing a business process into smaller functional definitions called business tasks. Each business task describes an ideally reusable functional task, such as Customer Synchronization, Get Customer Details, and so forth. The interaction of these business tasks forms the overall business process implemented by the integration.

Typically a solution or integration architect creates the functional decomposition of a business process. This person must be familiar with both the business process to be implemented and the AIA reference architecture.

Figure 11 shows the functional decomposition of the Foundation Pack Demo business process.

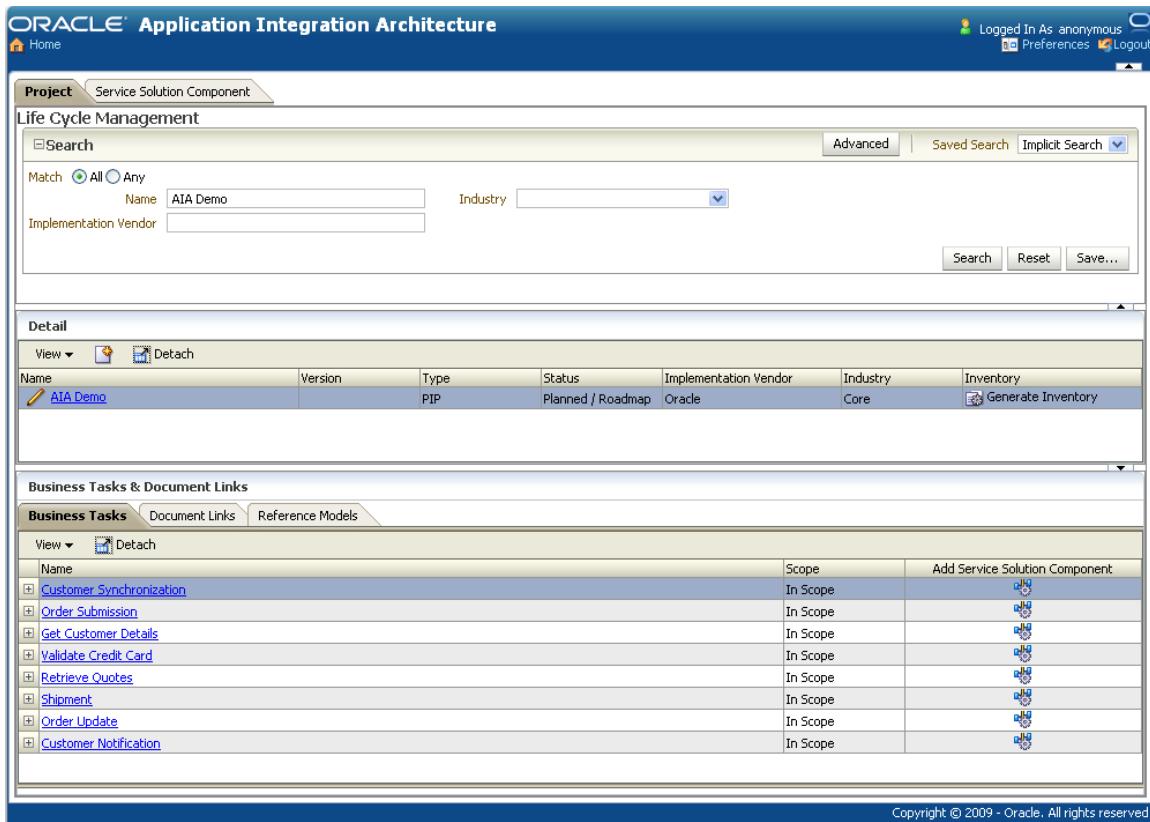


Figure 11: Functional decomposition of the AIA Foundation Pack Demo

In the next step, the architect identifies the AIA artifacts that are required to implement a given business task. Figure 12 shows the service solution components for the Update Order business task. The service solution component actually implements the functionality to update the order in the shop to display the outcome of the order processing to the customer.

For the implementation of this business task, the architect identified the need for a provider ABCS and an EBS, as well as the relationship to the CBP orchestrating the order processing.

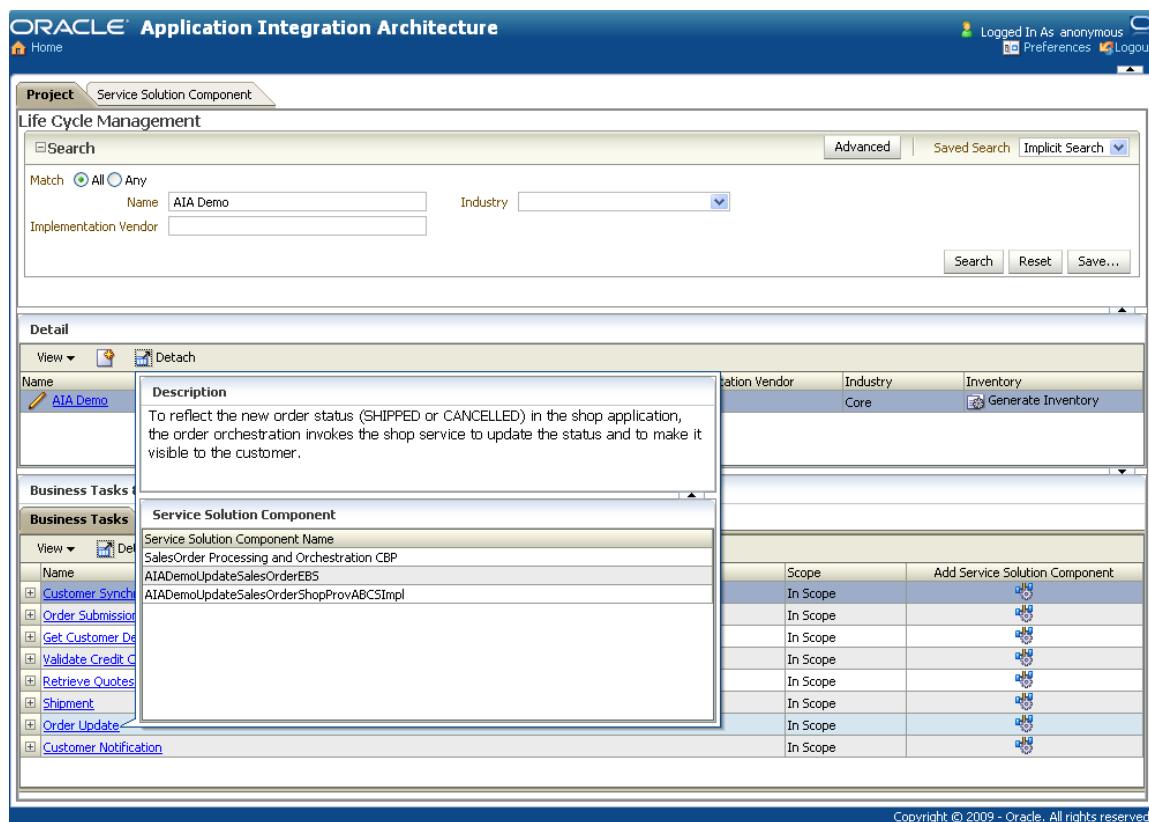


Figure 12: AIA artifacts for the Business Task “Update Order”

4.3. Creating the Enterprise Business Services and Messages

Always start with identifying the EBSs that you need to implement your integration flow.

For the AIA Foundation Pack Demo, we have identified the following AIA -delivered EBOs as being relevant to the business flow:

- CustomerPartyEBO: Represents the customer purchasing items.
- SalesOrderEBO: Contains all items in a customer’s purchase.
- ReceivedPaymentEBO: Used for the credit card validation.
- RequestForQuoteEBO: Carries the information to request item prices from multiple sources.
- ShipmentRequestEBO: Used to send a shipment request to a logistics provider.

To handle these EBOs, we need the following EBSs with their corresponding operations and messages, as shown in the following table:

EBO	EBS Implementation	Enterprise Business Messages
CustomerPartyEBO	AIADemoSyncCustomerPartyListEBS	SyncCustomerPartyListEBM
	AIADemoQueryCustomerPartyEBS	QueryCustomerPartyEBM, QueryCustomerPartyResponseEBM
SalesOrderEBO	AIADemoProcessSalesOrderOrchestrationEBS	ProcessSalesOrderEBM
	AIADemoUpdateSalesOrderEBS	UpdateSalesOrderEBM
	AIADemoProcessSalesOrderStatusEBS	ProcessSalesOrderEBM
ReceivedPaymentEBO	AIADemoProcessCreditChargeAuthorizationEBS	ProcessCreditChargeAuthorizationEBS, ProcessCreditChargeAuthorizationResponseEBS
RequestForQuoteEBO	AIADemoCreateRequestForQuoteEBS	CreateRequestForQuoteEBM, CreateRequestForQuoteResponseEBM
ShipmentRequest	AIADemoCreateShipmentRequestEBS	CreateShipmentRequestEBM

Note that some of the implemented operations follow the asynchronous fire-and-forget pattern (for example, CreateShipmentRequest) while others are synchronous request-reply operations that immediately deliver a response for a request. This is why some of the operations do not need a response Enterprise Business Message (EBM), whereas operations for the request-reply interaction pattern (for example, QueryCustomerParty) require response messages.

EBSs are always implemented as SOA composites leveraging a Mediator component for the routing of messages to provider ABCSs.

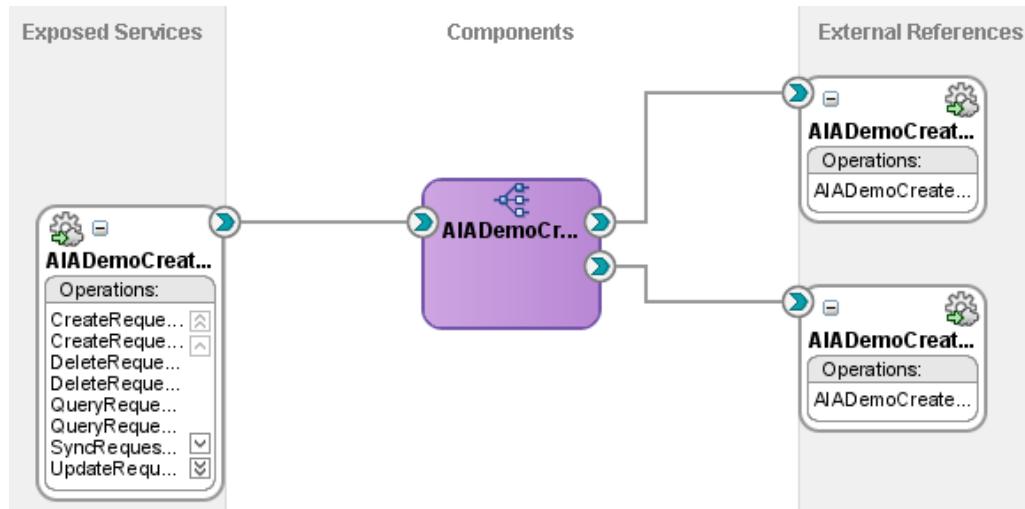


Figure 13: SOA composite AIADemoCreateRequestForQuoteEBS

Figure 13 illustrates the AIADemoCreateRequestForQuoteEBS implementation. As there are two service providers (Warehouse and Supplier), the EBS needs to route to the appropriate service provider based on the actual message payload.

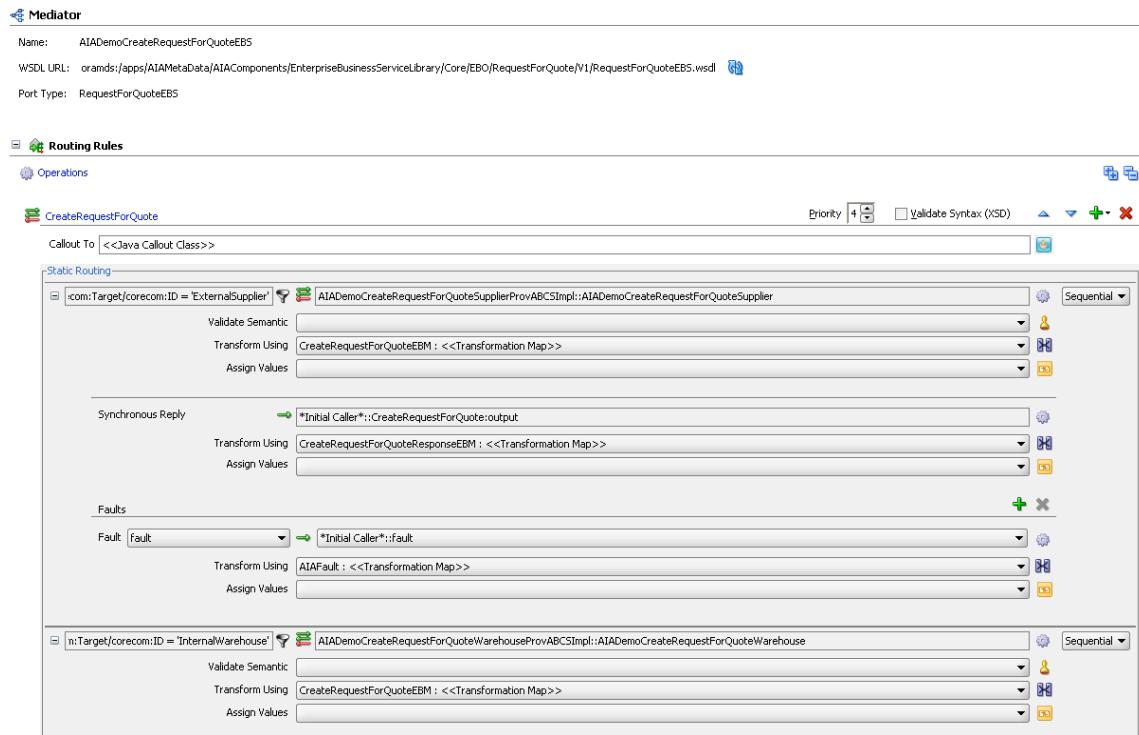


Figure 14: Routing rules for Mediator component AIADemoCreateRequestFor-QuoteEBS

Figure 14 shows the routing rules for the AIADemoCreateRequestForQuoteEBS Mediator component. There are two static routing rules, one of them routing to the AIADemoCreateRequestForQuoteWarehouseProvABCImpl and the other one pointing to AIADemoCreateRequestForQuoteSupplierProvABCImpl.

Also note the conditions for each routing rule, like “..ID='ExternalSupplier'” for the first routing rule. The AIA Foundation Pack Demo leverages existing structures in the EBM header to store routing information so that the EBS can determine the correct service to call.

4.4. Creating the Application Business Connector Services

Create Application Business Connector Services (ABCSS) to implement the services identified in the context of specific applications.

For every operation of the generic EBSs that we identified, we now need to implement the respective ABCSSs. ABCSSs take the EBMs as input and output (for request-reply) and transform them into application structures and method invocations.

For the AIA Foundation Pack Demo, we have the following ABCSSs along with the participating applications with which they are communicating:

ABCS	Role	Participating Application or Service
Customer Synchronization		
AIADemoSyncCustomerPartyListShopReqABCSImpl	Requester	Shop
AIADemoSyncCustomerPartyListCRMProvABCSImpl	Provider	CRM
Order Processing		
AIADemoProcessSalesOrderShopReqABCSImpl	Requester	Shop
AIADemoQueryCustomerPartyCRMProvABCSImpl	Provider	CRM
AIADemoProcessCreditChargeAuthorizationABCSImpl	Provider	Credit Card Validation Service
AIADemoCreateRequestForQuoteWarehouseProvABCSImpl	Provider	Warehouse
AIADemoCreateRequestForQuoteSupplierProvABCSImpl	Provider	Supplier
AIADemoUpdateSalesOrderShopProvABCSImpl	Provider	Shop
AIADemoCreateShipmentRequestUSPSProvABCSImpl	Provider	USPS Service
AIADemoCreateShipmentRequestBatchProvABCSImpl	Provider	Shipment Batch
AIADemoNotifyCustomerProvABCSImpl	Provider	Notification System
B2B Order		
X12ProcessSalesOrderReqB2BCSImpl	Requester	B2B
X12UpdateSalesOrderProvB2BCSImpl	Provider	B2B

As previously stated, an ABCS can have either the requester or provider role in an integration flow. For instance, the requester ABCS AIADemoProcessSalesOrderShopReqABCSImpl is invoked by the Shop application to process a new order. The ABCS then transforms the order into the EBM containing the SalesOrderEBO structure and invokes the AIADemoProcessSalesOrderOrchestrationEBS.

On the other hand, a provider ABCS, such as AIADemoQueryCustomerPartyCRMProvABCSImpl, is called by AIADemoQueryCustomerPartyEBS. It receives the request EBM (including a unique customer key) to read customer details from the CRM application. It transforms this payload into the format the embedded database adapter requires to read the customer record from the CRM_CUSTOMER database table. The ABCS then takes care of transforming the retrieved record into the CustomerPartyEBO structure and sends this back to the requester embedded into a QueryCustomerPartyResponseEBM message.

Figure 15 illustrates the SOA composite for AIADemoQueryCustomerPartyCRMProvABCSImpl. The overall logic is implemented in the embedded BPEL process and the Mediator component is responsible for the interaction with the database adapter.

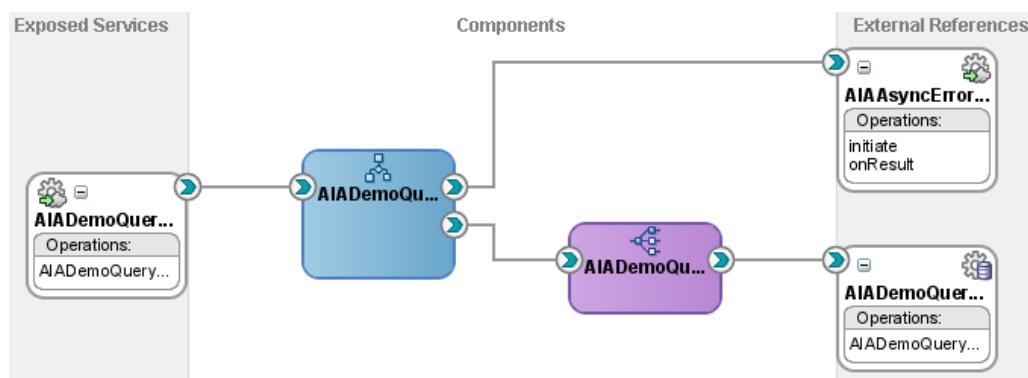


Figure 15: SOA composite AIADemoQueryCustomerPartyCRMProvABCSImpl

Drilling down into the BPEL process, Figure 16 illustrates the details of the implemented logic in the BPEL process. It mainly performs the following tasks:

- Receives the QueryCustomerPartyEBM (including the customer identifier) and transforms it into the format expected by the database adapter service.
- Invokes the database adapter service implemented as a Mediator component.
- Transforms the response into the QueryCustomerPartyResponseEBM (including the CustomerPartyEBO) and responds to the service caller, AIADemoQueryCustomerPartyEBS.

The remaining parts of the BPEL process are responsible for a proper integration into the AIA Error Handling Framework and the AIA Composite Application Validation System (CAVS) testing framework.

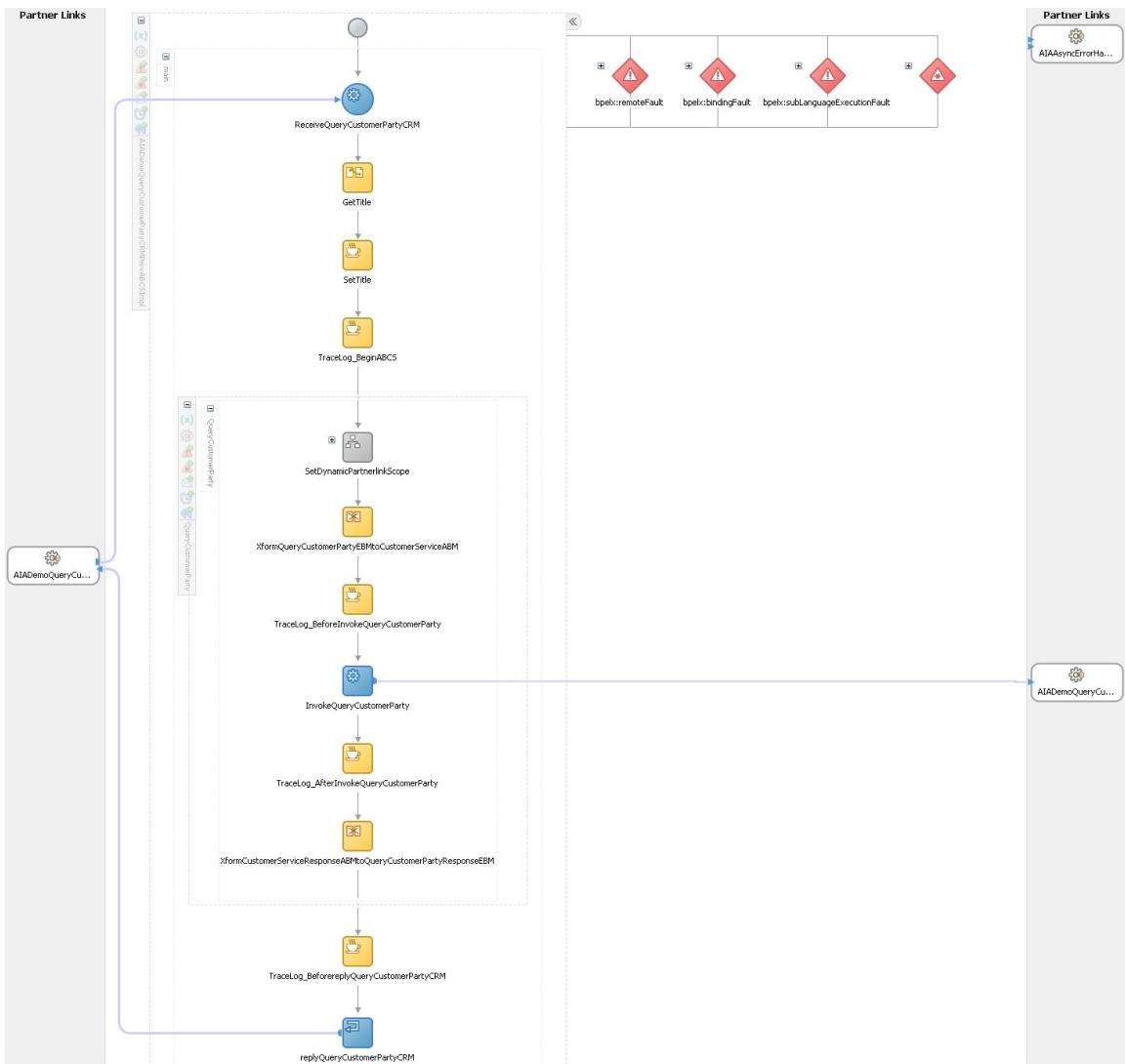


Figure 16: BPEL process implementing the ABCS logic for Query CustomerParty

The AIA Foundation Pack Service Constructor is an extension to JDeveloper 11g that can help jumpstart the development of ABCSs. A wizard guides developers through the process of providing information required to build a SOA composite, such as service name, role (requester or provider), interface definition, and so forth.

Based on the information gathered, the Service Constructor creates a new SOA composite with an embedded BPEL process that adheres to AIA guidelines and which already includes the required code to fully support AIA best practices, such as support for error handling, trace logging, CAVS, extensibility, and so forth.

Figure 17 illustrates the use of Service Constructor to build the ABCS for the Credit Card Validation Service.

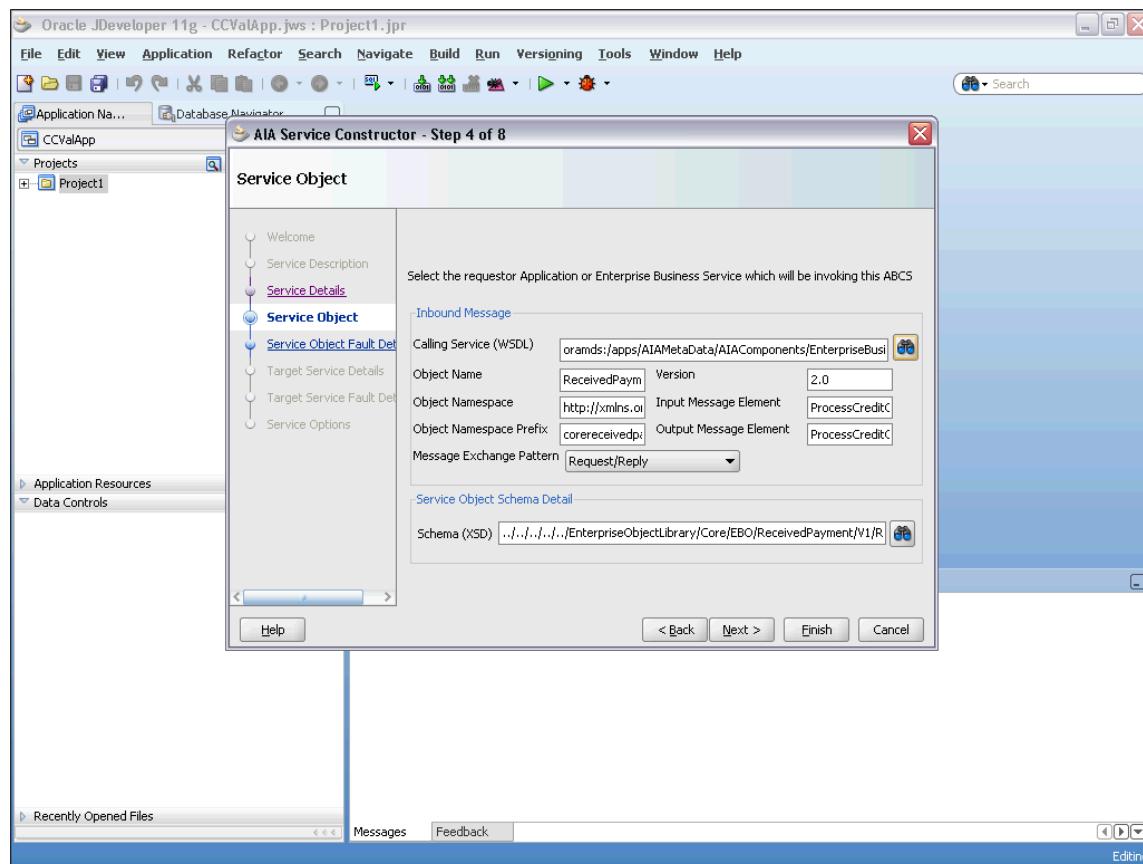


Figure 17: Service Constructor wizard for creating an ABCS

4.5. Creating the Composite Business Process

The Composite Business Process (CBP) is the BPEL process that orchestrates the EBSs to implement the required business process.

After defining the EBSs and ABCSs, we have everything in place that we need to orchestrate the entire Order-to-Fulfillment business flow. For the orchestration of process activities, we need a CBP. In the AIA Foundation Pack Demo, the BPEL-based composite, `AIADemoProcessSalesOrderCBP`, has this responsibility.

In essence, the `AIADemoProcessSalesOrderCBP` implements a series of sequential and parallel calls to all of the EBSs we have implemented so far. The EBF itself is launched by the `AIADemoProcessSalesOrderOrchestrationEBS`. It then invokes other EBS operations such as `AIADemoQueryCustomerPartyEBS` or `AIADemoProcessCreditChargeAuthorizationEBS`, along with several other operations as required by the business process.

Figure 18 illustrates the part of `AIADemoProcessSalesOrderCBP` that requests prices from the Warehouse and a Supplier in parallel and then selects the best offered price.



Figure 18: The part of the BPEL implementation of AIADemoProcessSalesOrderCBP that requests prices for the items

With the implementation of the CBP, we have completed the implementation of the AIA Foundation Pack Demo. We are now ready to deploy all services built to an Oracle AIA Foundation Pack instance to see how the integration flow works.

4.6. Harvesting the AIA Demo Artifacts

The AIA Harvester introspects the developed artifacts and publishes information to the AIA Project Lifecycle Workbench and optionally, to Oracle Enterprise Repository.

After completing development of the required AIA artifacts, such as ABCSs, EBSs, Adapter Services, and so forth, harvesting these design time artifacts is the next logical step in the AIA lifecycle. The AIA Harvester tool takes care of the following:

- Introspecting the SOA composites and their annotations in particular to enrich information in the AIA Project Lifecycle Workbench.
- Optionally publishing the artifacts to Oracle Enterprise Repository for all SOA governance purposes.

The quality of information to be published to the AIA Project Lifecycle Workbench is based on the annotations provided by the developer when creating the artifacts. Only if developers annotate their services according to the guidelines provided in the *Oracle Application Integration Architecture Foundation Pack: Development Guide*, can the AIA Harvester publish useful information to the AIA Project Lifecycle Workbench to be consumed in the subsequent deployment phase of the project.

The Foundation Pack Demo composites have been properly annotated so that the harvested information can be used later in the AIA lifecycle to create a deployment plan that includes the following instructions to:

- Create database objects.
- Configure the Oracle Fusion Middleware environment. For example, to create JMS resources required by the Foundation Pack Demo at runtime.
- Deploy the Foundation Pack AIA artifacts.

The AIA Harvester is a command-line tool located in `AIA_HOME/Infrastructure/LifeCycle/AIAHarvester` and can be executed as shown in Figure 19.

```
AIAHarvest.sh/.bat -settings <AIA_INSTANCE>/Infrastructure/LifeCycle/
Install/PIPHarvest/AIADemoLocalHarvesterSetting.xml
```

Figure 19: Executing the AIA Harvester to harvest the Foundation Pack Demo artifacts

The Foundation Pack Demo includes a harvester settings file that enables all Foundation Pack Demo SOA composites to be harvested by executing a single statement. Note that this harvester command publishes both to the AIA Project Lifecycle Workbench and Oracle Enterprise Repository in one step.

For more information about annotating AIA artifacts, see *Oracle Application Integration Architecture Foundation Pack: Development Guide*.

4.7. Deploying the Foundation Pack Demo

Foundation Pack integrations can be deployed in one step by leveraging the deployment plan that is generated by the AIA Project Lifecycle Workbench and the AIA Deployment Plan Generator.

The AIA Lifecycle Application uses information defined during the functional decomposition phase and the technical details captured by the AIA Harvester to generate a deployment plan for the entire integration project or a selected subset of it.

This deployment plan includes not only the SOA composites that were built for the project, but also accounts for related deployments and configurations, such as database scripts, JEE applications, and middleware configurations, such as messaging, adapter configuration, and so forth.

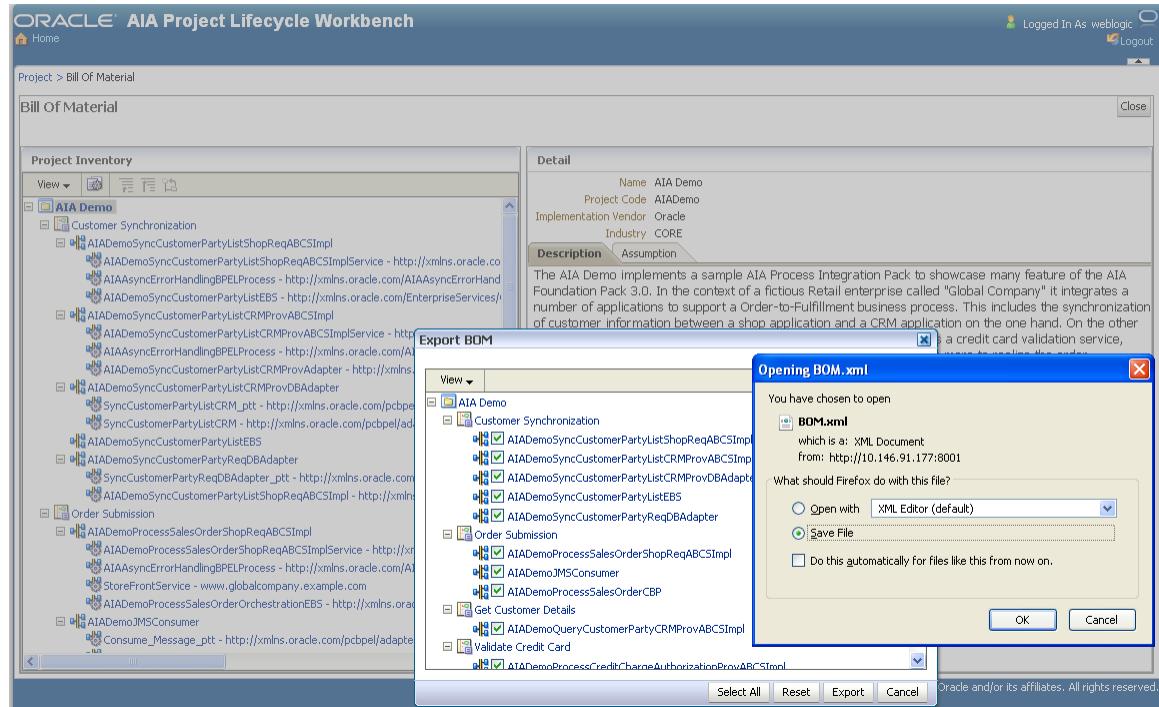


Figure 20: Generating the deployment plan from the AIA Project Lifecycle Workbench

Figure 20 shows the extraction of the bill of material information from the AIA Project Lifecycle Workbench. The bill of material can finally be converted into a fully functional deployment plan using the AIA Deployment Plan Generator. This tool is located in `<AIA_HOME>/util/DeploymentPlanGenerator` and can be executed as shown in Figure 21.

```
ant -Dinput=<Path to file>/AIADemoBOM.xml  
-DDeploymentPlan=<Some path>/AIADemoDP_generated.xml  
-DHarvesterSettings=< Some path>/AIADemoHS_generated.xml
```

Figure 21: Creating a deployment plan based on the bill of material exported from the AIA Project Lifecycle Workbench

Now that we have generated a deployment plan for the Foundation Pack Demo, we can deploy the entire integration project by leveraging the AIA Installation Driver as shown in Figure 22:

```
ant -f <AIHome>/Infrastructure/Install/AID/AIAInstallDriver.xml  
-DPropertiesFile=<AI_INSTANCE>/config/AIAInstallProperties.xml  
-DDeploymentPlan=<Path to file>/AIADemoDP generated.xml
```

Figure 22: Deploying the Foundation Pack Demo using the generated deployment plan

Note. For the user's convenience, the Foundation Pack Demo ships with deployment plans to deploy and undeploy the entire Foundation Pack Demo in a single step. These deployments are located in <AIA_HOME>/samples/AIAutoDemo/config and can be easily initiated as shown in Figure 23.

```
ant -f (un)deployDemo.xml
```

Figure 23: Simplified deployment of the Foundation Pack demo with provided scripts

For more information about deploying AIA artifacts, see the *Oracle Application Integration Architecture Foundation Pack: Development Guide*.

5. Running the Foundation Pack Demo

The Foundation Pack Demo reuses many parts of the Oracle SOA Suite 11g Fusion Order Demo. As such, the Foundation Pack Demo uses the same online store application to place new orders and to kickoff the demo order processing.

You can access this application on the Oracle SOA Suite instance with the deployed Foundation Pack Demo using the URL <http://<server>:<port>/StoreFrontModule/faces/home>.

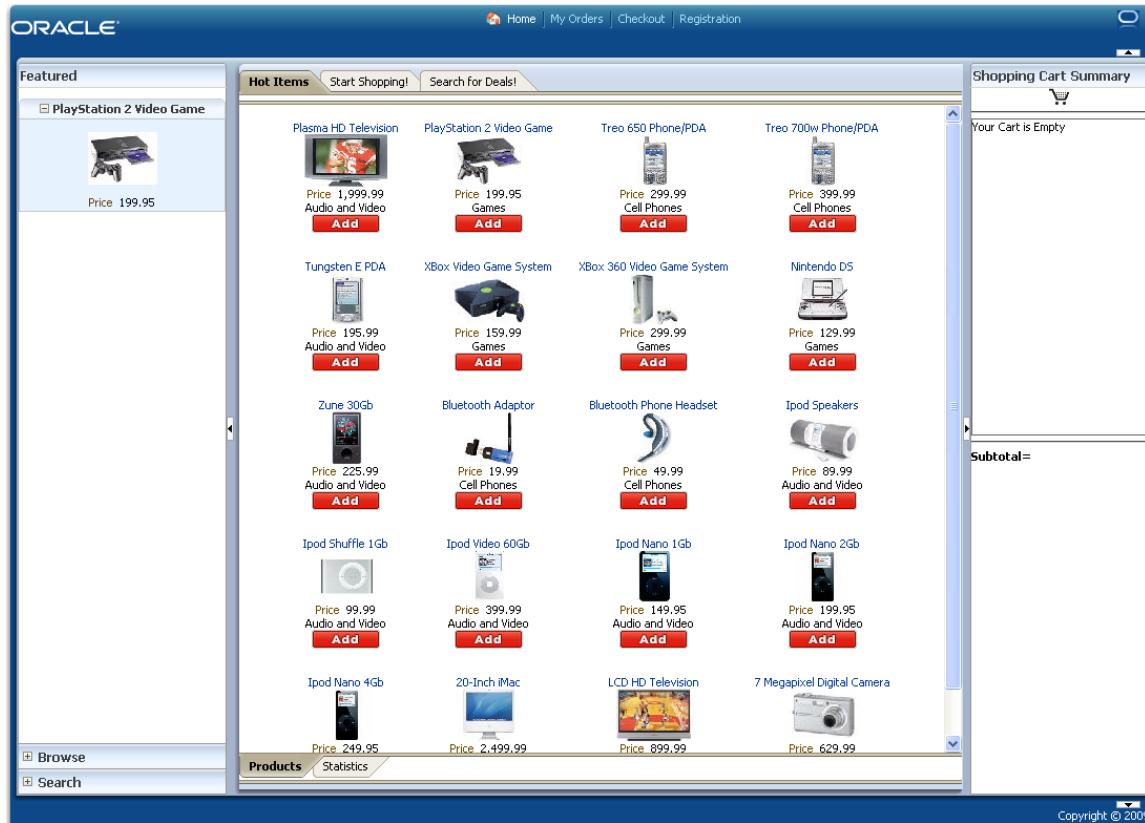


Figure 24: Foundation Pack Demo Online Store application

5.1. Initiating the Customer Synchronization Flow

To initiate the customer synchronization flow, click the “Registration” link at the top of the screen and follow the registration wizard pages. On the first page, ensure to not only provide information for all mandatory fields, but also for “First Name” and “Last Name” as these values are required as a part of the flow.

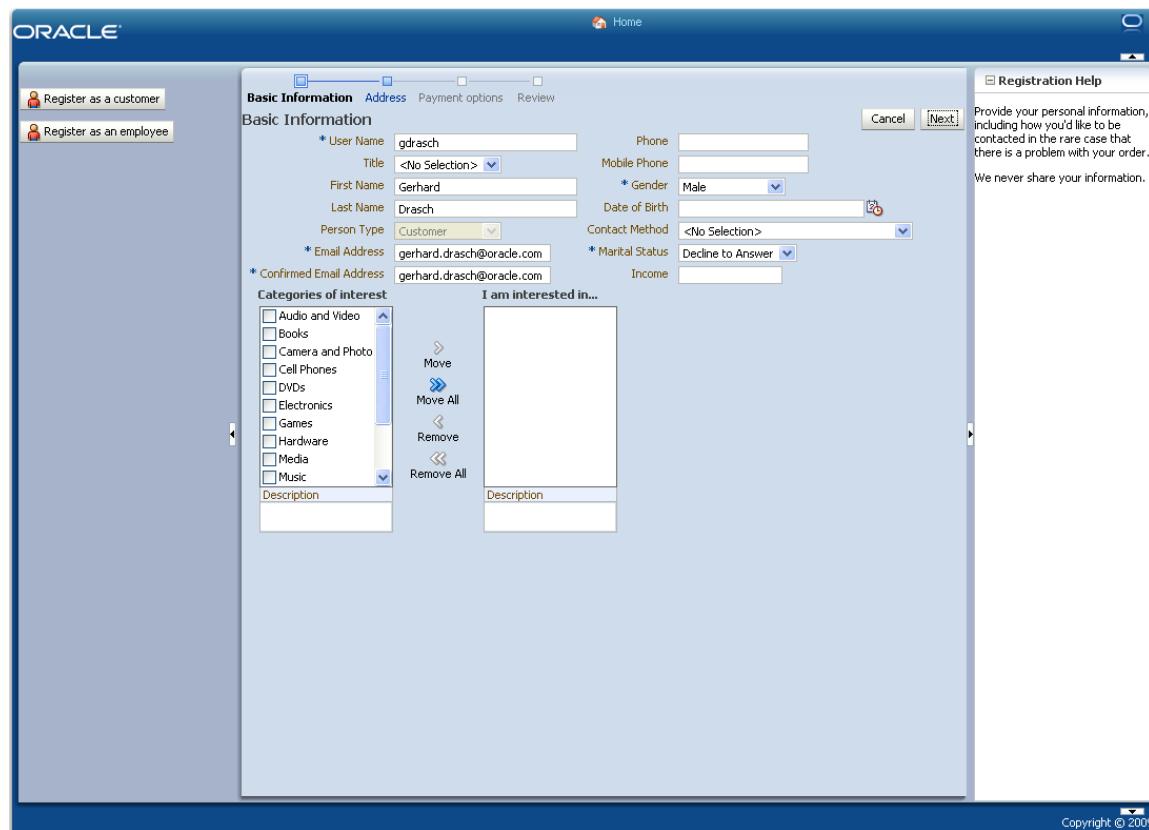


Figure 25: Account registration in the shop application

The information on the following pages for addresses and payment options can be skipped. Once you click “Register” on the final page of the registration wizard, the Oracle Application Integration Architecture (AIA) integration flow will be triggered and the account information will be synchronized to the CRM application, including the maintenance of cross-referencing records that are used to map the customer account across applications.

You can review the initiated SOA composites in the Oracle Fusion Middleware Enterprise Manager console. You can click any of the initiated instances to view the entire process flow with all AIA artifacts involved.

5.2. Placing an Order

To place an order in the shop application for further processing, select items on the shop main page (see Figure 24) by either clicking one of the “Add” buttons or by dragging items to the shopping cart on the right.

When you are done selecting items, click the “Checkout” link at the top of the screen and provide the login credentials “ngreenbe” with a password of “welcome1” on the next page. On the following order review page, click “Submit Order” to kickoff order processing in the AIA layer.

ORACLE®

Exit and Continue Shopping

Welcome ngreenbe Logout

Shipping Details

Customer Information

General Information

Customer Name: Nancy Greenberg
Member Since:
Email Address: NGREENBE
Mobile Phone:
Phone Number: 865.555.0102

Primary Address

Address Line 1: 2549 Yonge Street
City: Toronto
Postal Code or ZIP: M4P 2H9
State / Province: ON
Country: CA

Order Information - #1381

Shipping Information

Ship to:
Shipping Address: 100 N Peach St Philadelphia PA 19139 US
Phone Number:

Shipping Options

Shipping Option Code: Standard Shipping (3-5 business days) Two-Day Shipping One-Day Shipping Pick-up

Gift Options

Gift Wrapping Message: None

Payment Options

Payment Option Code: 4111111111111111 VISA

Discounts

Coupon Code:

Order Summary

Bluetooth Adaptor
Cell Phones
19.99 -quantity: 1

Ipod Speakers
Audio and Video
89.99 -quantity: 1

Nintendo DS
Games
129.99 -quantity: 1

Treo 700w Phone/PDA
Cell Phones
399.99 -quantity: 1

Items: \$639.96
Shipping & Handling: \$39.09
Discounts: \$20.00
Your Total: \$659.05

Submit Order

Copyright © 2009

Figure 26: Order submission

After placing the order, you can check on the initiated SOA composites in the Oracle Fusion Middleware Enterprise Manager console. You can click any of the initiated instances to view the entire process flow with all AIA artifacts involved.

Flow Trace 

This page shows the flow of the message through various composite and component instances. 

ECID 00001Hgf2DMBHBAnv~r2F11Ar54o00002q:47980
Started Oct 20, 2009 8:02:08 AM

Faults (0)
Faults
Select a fault to locate it in the trace view.
Error Message
No faults found

Trace
Click a component instance to see its detailed audit trail.
Show Instance IDs 

Instance	State	Type	Time	Composite Instance
AIADemoProcessSalesOrderShopReqABCImpl	Completed	BPEL Component	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderShopReqABCImpl of 1240070
StoreFrontService	Completed	Reference	Oct 20, 2009 8:02:05 AM	AIADemoProcessSalesOrderShopReqABCImpl of 1240070
AIADemoProcessSalesOrderOrchestrationEBS	Completed	Reference	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderOrchestrationEBS of 1240071
AIADemoProcessSalesOrderOrchestrationEBS_2	Completed	Service	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderOrchestrationEBS of 1240071
AIADemoProcessSalesOrderOrchestrationEBS	Completed	Mediator Component	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderOrchestrationEBS of 1240071
AIADemoProcessSalesOrderCBP	Completed	Reference	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderOrchestrationEBS of 1240071
AIADemoProcessSalesOrderCBP_2	Completed	Service	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderOrchestrationEBS of 1240071
AIADemoProcessSalesOrderCBP	Completed	BPEL Component	Oct 20, 2009 8:02:09 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoQueryCustomerPartyEBS	Completed	Reference	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoQueryCustomerPartyEBS_2	Completed	Service	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoQueryCustomerPartyEBS	Completed	Mediator Component	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoQueryCustomerPartyCRMProvABC	Completed	Reference	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoQueryCustomerPartyCRMProvV	Completed	Service	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoQueryCustomerPartyCRMProvV	Completed	BPEL Component	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoQueryCustomerPartyCRMEE	Completed	Mediator Component	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoQueryCustomerPartyCRMEE	Completed	Reference	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoProcessCreditChargeAuthorizationEBS	Completed	Reference	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240072
AIADemoProcessCreditChargeAuthorizationEBS	Completed	Service	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240075
AIADemoProcessCreditChargeAuthorizationEBS	Completed	Mediator Component	Oct 20, 2009 8:02:07 AM	AIADemoProcessSalesOrderCBP of 1240075
AIADemoProcessCreditChargeAuthorization	Completed	Reference	Oct 20, 2009 8:02:06 AM	AIADemoProcessSalesOrderCBP of 1240075

Figure 27: Reviewing SOA artifacts for order processing in Enterprise Manager

5.3. Orchestrating the B2B Order Flow

To initiate the business-to-business (B2B) flow of the AIA Foundation Pack Demo, just copy a sample EDI 850 document into the directory where the B2B configuration expects to locate incoming trading documents. By default, in Linux/Unix-based environments, this is the directory “/tmp/GlobalChips_File_Inbound”.

A sample EDI 850 document with name “GlobalChips_123.dat” can be found in this directory: <AIA_HOME>/samples/AIAuto/B2BSamples/X12ProcessSalesOrderReqB2BCSImpl. By copying this file into the incoming trading documents directory, the Oracle B2B Server translates the message and kicks off the order processing in AIA.

As a part of the order processing, the flow generates a B2B 855 response message. By default, in Linux/Unix-based environments, the Oracle B2B Server generates this response message in the directory “/tmp/GlobalChips_File_Outbound”.

ORACLE B2B

Administration | Partners | Reports | Metrics | Help | Logout

Logged in as weblogic

Business Message | Wire Message | Application Message | Error | Conversation

Business Message

Search

Match: All (Any)

Sender Contains:

Receive Time Stamp: Greater Than (UTC-08:00) US Pacific Time

Receiver Contains:

State Equals:

Agreement Contains:

Message Id Contains:

Send Time Stamp: Greater Than 10/19/2008 12:00:00 AM (UTC-08:00) US Pacific Time

Advanced | Saved Search | Default

Search | Reset | Save...

Result

Details	State	Document Type	Agreement	Sender	Receiver	Receive Time Stamp	Send Time Stamp
MSG_ERROR						Monday, October 19, 2009 5:17	Monday, October 19, 2009 5:17
MSG_COMPLETE	997	Acme_to_GlobalChips_997_401 Acme		GlobalChips		Monday, October 19, 2009 5:17	Monday, October 19, 2009 5:17
MSG_COMPLETE	850	GlobalChips_to_Acme_850_401 GlobalChips		Acme		Monday, October 19, 2009 5:14	Monday, October 19, 2009 5:14
MSG_ERROR						Friday, October 9, 2009 12:42	Friday, October 9, 2009 12:42
MSG_COMPLETE	997	Acme_to_Advantage_997_501 Acme		AdvantageCorp		Friday, October 9, 2009 12:42	Friday, October 9, 2009 12:42
MSG_COMPLETE	850	Advantage_to_Acme_850_501 AdvantageCorp		Acme		Friday, October 9, 2009 12:41	Friday, October 9, 2009 12:41
MSG_ERROR						Friday, October 9, 2009 12:35	Friday, October 9, 2009 12:35
MSG_COMPLETE	997	Acme_to_GlobalChips_997_401 Acme		GlobalChips		Friday, October 9, 2009 12:35	Friday, October 9, 2009 12:35
MSG_COMPLETE	850	GlobalChips_to_Acme_850_401 GlobalChips		Acme		Friday, October 9, 2009 12:35	Friday, October 9, 2009 12:35
MSG_COMPLETE	850	GlobalChips_to_Acme_850_401 GlobalChips		Acme		Friday, October 9, 2009 12:28	Friday, October 9, 2009 12:28
MSG_COMPLETE	997	Acme_to_GlobalChips_997_401 Acme		GlobalChips		Friday, October 9, 2009 12:28	Friday, October 9, 2009 12:28
MSG_COMPLETE	997	Acme_to_GlobalChips_997_401 Acme		GlobalChips		Friday, October 9, 2009 12:03	Friday, October 9, 2009 12:03
MSG_COMPLETE	850	GlobalChips_to_Acme_850_401 GlobalChips		Acme		Friday, October 9, 2009 12:03	Friday, October 9, 2009 12:03
MSG_COMPLETE	997	Acme_to_GlobalChips_997_401 Acme		GlobalChips		Friday, October 9, 2009 11:54	Friday, October 9, 2009 11:54
MSG_COMPLETE	850	GlobalChips_to_Acme_850_401 GlobalChips		Acme		Friday, October 9, 2009 11:54	Friday, October 9, 2009 11:54

Purge | ReSubmit App Message | ReSubmit Wire Message

Copyright © 2004, 2009, Oracle and/or its affiliates. All rights reserved.

Figure 28: Oracle B2B Console showing incoming and outgoing trading messages

You can verify incoming and outgoing EDI messages using the Oracle B2B Server console available at <http://<server>:<port>/b2b>. Click “Reports” and search for recent business messages. You should be able to see records related to the EDI 850 and EDI 855 messages types.

6. Extending the Foundation Pack Demo

Extensibility is a key principle in the design of Application Integration Architecture (AIA).

One of the key principles for the design of Oracle Application Integration Architecture (AIA) is its extensibility model. AIA offers extensibility in different shapes.

First, the design of the Enterprise Business Objects (EBOs) includes mechanisms to extend these generic objects in an upgrade-safe manner by providing hooks to plug in additional industry-specific or customer-specific information.

Second, the entire architecture is prepared to consistently add additional applications to an existing integration process by allowing the plugging in of additional Application Business Connector Services (ABCSs).

The following sections describe typical extensions to existing integrations. We will cover how the AIA Foundation Pack offers ways to tailor common objects such as EBOs and Enterprise Business Messages (EBMs) according to your unique business requirements.

We will also show you how to consistently plug other applications into a new or existing integration process. For this purpose, we will change the fulfillment process to choose the distributor depending on the actual amount of the order. To achieve this extension, we implement another ABCS for the additional distributor service and wire that into the existing `AIADemoCreateShipmentRequestEBS`.

6.1. Extending an Enterprise Business Object

Oracle has invested heavily in creating the proper definitions of EBOs, such as `SalesOrderEBO` and `CustomerPartyEBO`. These prepackaged canonical representations of typical business entities were created by considering both standard approaches, such as Open Applications Group Integration Specification (OAGIS), as well as the rich set of Oracle-owned applications, such as Oracle E-Business Suite, Siebel, PeopleSoft, and others.

However, these definitions should never contain any attributes that are specific to certain industries or customers. This is why the AIA Foundation Pack provides the means to extend EBOs in a nonintrusive way. These extensions will survive any patches and upgrades to protect your investments.

For this purpose, AIA provides a custom schema definition file (XSD) for every EBO that is shipped with AIA. These custom schemas are incorporated into the actual EBOs and are never touched by patches or upgrades.

The custom extensions of EBOs are preserved across patches and upgrades.

In the context of the AIA Foundation Pack Demo business process, we use this mechanism to extend the CreateRequestForQuoteResponseEBM message structure to simply allow us to return an overall price for the items requested in the message CreateRequestForQuoteEBM. As there is no attribute for this in the standard message, we can add a custom attribute for this integration using the AIA extensibility model.

6.2. Extending a Process Flow

In this section, we drill down into what is required to extend an existing business flow. For the AIA Foundation Pack Demo, let's assume that Global Company decides to use different logistic providers for orders to be shipped outside of the United States. In this case we want the order to be shipped by *NewShipper*. For domestic shipments, the *USPS* service should be used. Figure 30 illustrates how this changes the flow. Note that any shipment request should still be routed to the batch system regardless of the address.

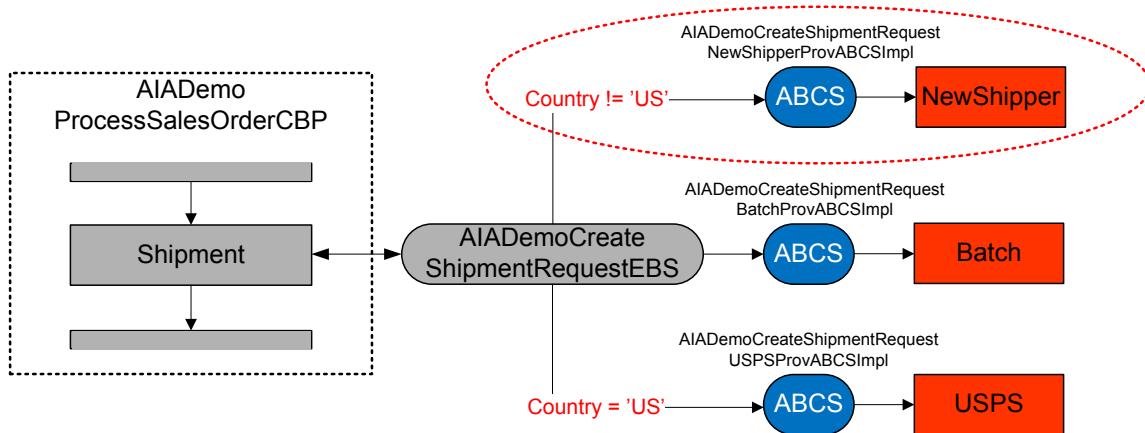


Figure 30: Extending the flow with another application (*NewShipper*)

To achieve this desired change to the flow, we need to identify and understand the services that make up the flow. In particular, we need to know the relevant CBPs, EBOs, and ABCSs of the integration.

For the extension of our AIA Foundation Pack Demo, we first require a new ABCS implementation for the NewShipper service called `AIADemoCreateShipmentRequestNewShipperProvABCSImpl`. Then, we will implement a moderate change to the `AIADemoCreateShipmentRequestEBS` to include the new application in the existing flow, as shown in Figure 30.

6.2.1. Creating an Application Business Connector Service

First, we want to implement the ABCS for the *NewShipper* logistics provider and call this `AIAutoCreateShipmentRequestNewShipperProvABCImpl`. To do this, we again leverage the AIA Service Constructor to jumpstart the development of this ABCS.

For more information about how to use the AIA Service Constructor, see the *Oracle application Integration Architecture Foundation Pack: Development Guide*.

6.2.2. Extending an Enterprise Business Service

As a second step, we need to include the AIADemoCreateShipmentRequestNewShipperProvABCImpl in the business flow. To achieve this, we need to extend the respective Enterprise Business Service (EBS), which is AIADemoCreateShipmentRequestEBS. EBSs are usually implemented as SOA composites with embedded Mediator components that take care of routing messages to the provider ABCSs.

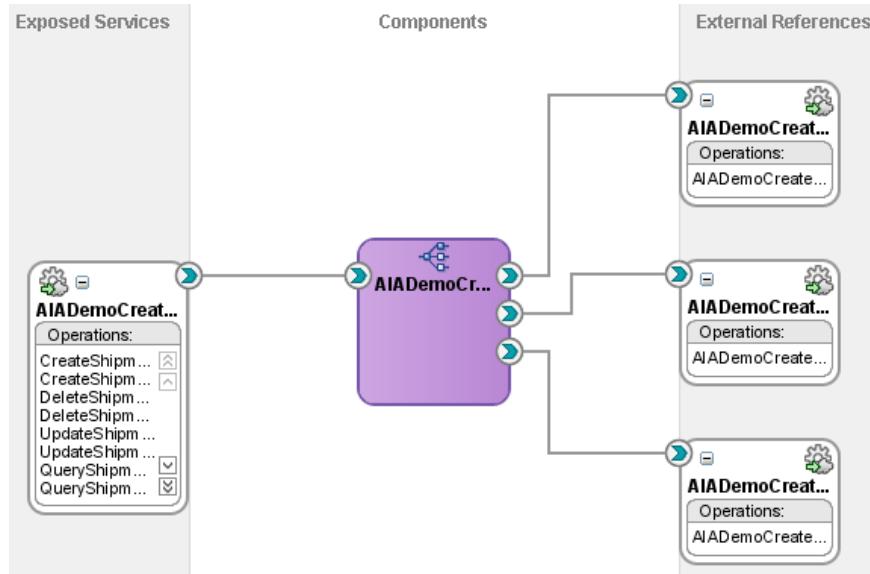


Figure 31: Extension of the AIADemoCreateShipmentRequestEBS SOA composite to now communicate with three provider ABCSs

While the basic implementation of the AIADemoCreateShipmentRequestEBS directly routes the fulfillment message to the AIADemoCreateShipmentRequestUSPSProvABCImpl and the AIADemoCreateShipmentRequestBatchProvABCImpl, the logic now needs to change.

With the availability of another provider ABCS, the content-based routing rules need to define which ABCSs to call for a particular order. For this purpose, we assign the following XPath filter criteria to the appropriate routing rules within the AIADemoCreateShipmentRequestEBS.

- USPS:

```
/CreateShipmentRequestEBM/.../ShipToPartyReference/.../CountryCode = 'US'
```

- NewShipper:

```
/CreateShipmentRequestEBM/.../ShipToPartyReference/.../CountryCode != 'US'
```

In an actual business implementation, this decision would not be hardcoded, but rather would be based on configurable business rules defined in Oracle Business Rules, a component of the Oracle SOA Suite. You can find an example of the uptake of business rules in AIADemoProcessSalesOrderOrchestrationCBP. The business rules are used to determine whether or not human approval is required for an order.

After deploying the new ABCS and the extended EBS to Oracle SOA Suite, we are done and the new flow is ready to go. You can now verify that new orders are shipped by *USPS* or *NewShipper*, depending on the amount of the order.

7. Introduction to Application Integration Architecture Infrastructure Components

Service-oriented architectures (SOAs) require a special set of infrastructure components to successfully manage the lifecycles of SOA artifacts such as services, message definitions, integrations scenarios, and so forth. The remainder of this document explains the Application Integration Architecture (AIA) infrastructure components in more detail and shows how organizations can benefit from them.

For more information about infrastructure components, see the *Oracle Application Integration Architecture Foundation Pack: Infrastructure Components and Utilities Guide*.

7.1. Overview of the Oracle Enterprise Repository

Service-oriented environments usually consist of hundreds of technically independent services. Organizations can quickly get lost without actual and consistent information about their SOA assets. They absolutely need a single source of truth to answer questions such as:

- Does a service exist that I can reuse for a certain purpose?
- What is the impact of a change to a service?
- How can I plug an additional service into an existing business process?

Oracle Enterprise Repository provides deep insight into an organization's SOA artifacts.

Oracle Enterprise Repository delivers this single source of truth that organizations need to manage their SOAs. The Oracle Enterprise Repository enables users to easily browse and search the catalog of SOA artifacts and get deep insight into the characteristics of each artifact. Oracle Enterprise Repository can handle all relevant artifact types such as SOA composites, service contracts (WSDLs), and schema definitions (XSDs). Furthermore, Oracle Enterprise Repository understands the relationship between these artifacts and provides useful views into this information.

In addition to the out-of-the-box functionality of Oracle Enterprise Repository, AIA adds specific taxonomies that clearly identify AIA asset types in Oracle Enterprise Repository. AIA Foundation Pack adds new taxonomies such as AIA:EBS, AIA:ProviderABCS, and so forth to enable Oracle Enterprise Repository users to easily identify AIA artifacts and their types.

Figure 32 illustrates the way in which users can use Oracle Enterprise Repository to search for relevant AIA artifacts. In this example, the user is searching for Enterprise Business Services (EBSs) available for the search term “item.” For this query, Oracle Enterprise Repository delivers the matching “ItemEBS” along with comprehensive details, such as the operations supported by the EBS and references to the WSDL and XSD artifacts that make up the EBS.

The screenshot shows the Oracle Enterprise Repository interface. The left sidebar has 'Assets' selected. The search bar at the top contains 'item'. The results table shows 100 of 359 items found. The first item in the list is 'ItemEBS' with the following details:

Name	Version	Type	Status
ItemEBS	1.0	Adapter Configuration	Registered

The 'Description' panel on the right shows the following details for 'ItemEBS':

- Registration Status: Registered
- AIA Namespace: <http://xmlns.oracle.com/ABCServiceImpl/Core/AIAQueryCustomerPartyCRMProvABCServiceImpl>
- AIA Service Name: AIAQueryCustomerPartyCRMProvABCServiceImpl
- AIA Version: 1.0
- Operations: AIA:ProviderABCS: {http://xmlns.oracle.com/ABCServiceImpl/Core/AIAQueryCustomerPartyCRMProvABCServiceImpl}AIAProviderABCS/AIAQueryCustomerPartyCRMPro

Figure 32: Accessing AIA artifact details in Oracle Enterprise Repository

7.2. Overview of the Composite Application Validation System

Developing and testing a SOA-based system has several unique requirements that need to be addressed by the SOA infrastructure, especially when the integrations are complex and involve external services. Because AIA process flows consist of several services calling each other, you must be able to test against a single service or an orchestration of services to be able to validate the health of the entire integration.

With Composite Application Validation System (CAVS), you can define, perform, and evaluate test invocations against any web service.

For exactly this purpose, AIA provides the CAVS. With CAVS, you can declaratively define tests on any kind of service. These tests can be performed at any time to ensure that services are working as expected.

Even if participating applications aren't available, CAVS can test single services or entire process flows by simulating these applications. By substituting applications with simulators, CAVS also allows the testing of single services or groups of services in isolation because the simulator removes the undesired dependency on specific behaviors of real services.

CAVS simulators can emulate any web service. You can develop and test integrations even when dependent services or applications are not in place.

The simulator functionality is also very useful in the development cycle, enabling you to create services even if dependent applications or services are not yet available. With simulators in place, developers can build their services in parallel, following a true SOA development methodology.

The AIA Foundation Pack Demo ships with a rich set of predefined test definitions and simulator definitions. Users can start using CAVS to verify the Foundation Pack Demo artifacts and flows by simply importing the test and simulator definitions in the AIADemo_CAVS_Definitions.xml file, which is delivered with the Foundation Pack Demo.

Figure 33: AIA Demo CAVS test and simulator definitions

The section covers a typical use case for using CAVS to verify an end-to-end integration. Specifically, we'll cover using CAVS to verify the customer synchronization flow of the Foundation Pack Demo. Figure 34 illustrates the message flow of this scenario.

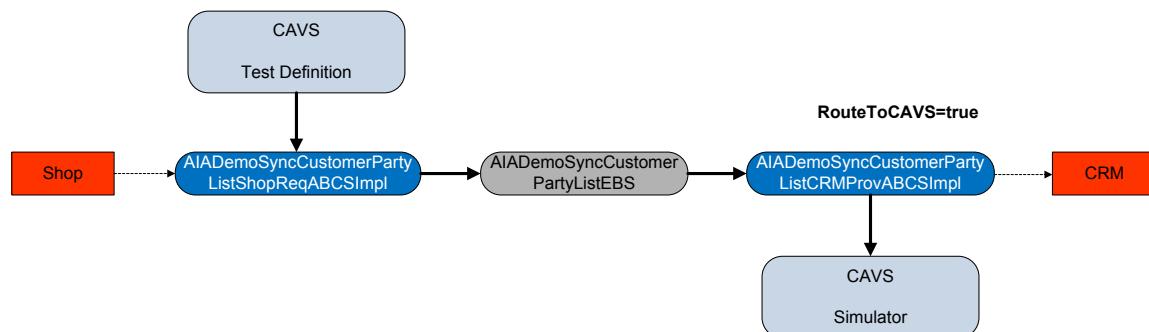


Figure 34: Testing an end-to-end flow in CAVS, including the use of a simulator

The CAVS test definition is configured to invoke the AIADemoSyncCustomerPartyListShopReq-ABCSImpl service and pass a valid payload to that service. The requester Application Business Connector Service (ABCS) operates in the usual way and invokes the EBS. The EBS itself routes the message to the provider ABCS called AIADemoSyncCustomerPartyListCRMProvABCSImpl.

The internal logic within the provider ABCS that enables support for CAVS checks the AIAConfigurationProperties.xml file to determine whether the services should operate in CAVS mode or not. In this case, we have enabled CAVS mode for the provider ABCS on the Configuration page of the AIA Console. For this reason, the provider ABCS does not invoke the adapter to the CRM application, but rather passes the message to the CAVS simulator endpoint.

Finally, the CAVS simulator engine identifies the right simulator definition based on the message payload and the available simulator definitions. The simulator definition can also contain criteria to verify the message payload. With this setup in place, we are able to use CAVS to verify that the end-to-end integration works as expected.

In a request-reply integration pattern, as opposed to the fire-and-forget scenario shown here, the test definition would also take care of evaluating the response message after invoking a service.

Note that when testing this scenario in Figure 34, you must change the message payload of the test definition to include unique values for personId and the firstName and lastName pair to avoid collisions in the cross-referencing table that is used during this flow.

7.3. Overview of the Error Handling Framework

A centralized error handling approach across applications, technologies, and integration patterns is key to being able to successfully operate a SOA.

The AIA Foundation Pack provides multiple approaches for handling errors in a straightforward and consistent way. AIA guidelines include best practices with regard to the implementation of error handling for AIA artifacts, while Foundation Pack provides additional capabilities on top of the Oracle SOA Suite to address error handling requirements in the application integration space.

The AIA Error Handling Framework includes rule-based error notifications. These error notification routing rules can include various parameters such as the integration flow, the name of the failing AIA artifact, the error code, and so forth. The error handling administrator can use these parameters to distinguish between business faults and system faults so that error notifications reach the people who are best suited to resolve the issue.

For example, an error notification about a system error, such as an application being unexpectedly unavailable, is more appropriately routed to a system administrator, while an error notification about a business error, such as failed order processing due to wrong payment information, is more appropriately routed to a business person. The Foundation Pack Demo handle use cases for both types of errors out-of-the-box and includes preconfigured notification routing rules to support both. Both types of errors can be triggered by providing “incorrect” payment options when submitting an order from the Foundation Pack Demo shop application.

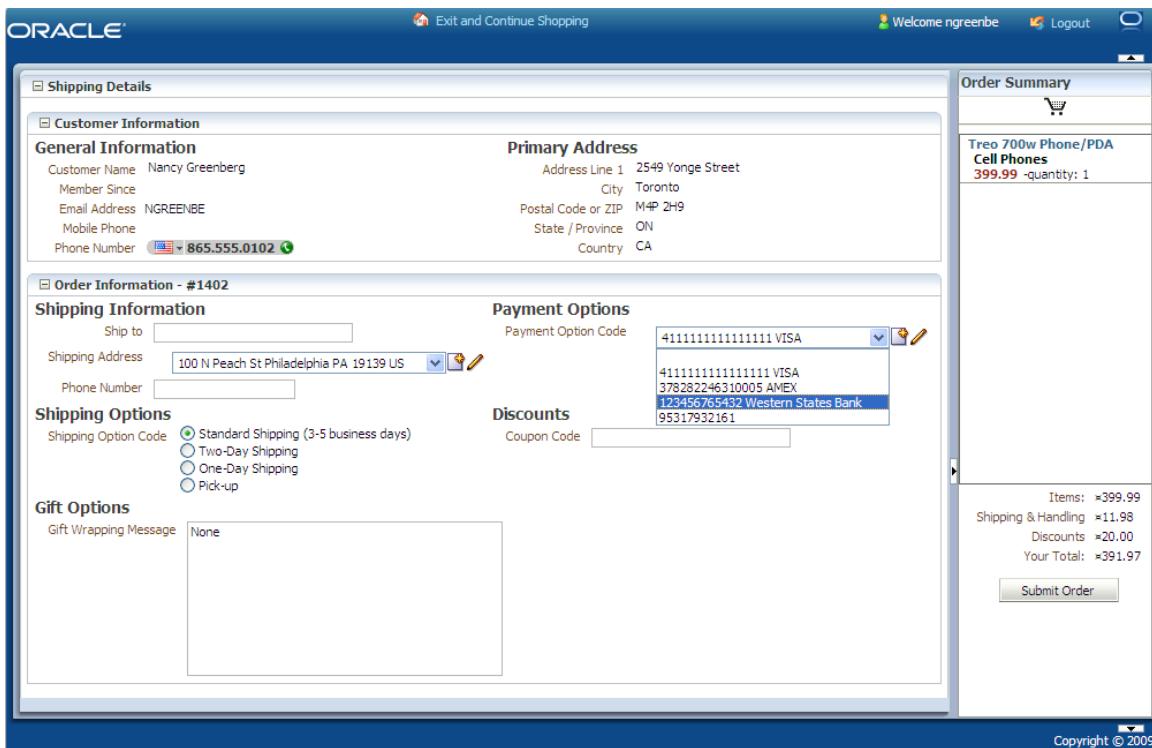


Figure 35: Providing order payment options designed to produce errors

The credit card validation service is implemented to reject credit cards with a card number starting with 1, 2, or 3, while credit card numbers starting with 4 or higher are considered valid. Also, the credit card validation service replies with a fault if no credit card type is supplied for the validation. The latter behavior can be triggered when selecting a bank account in the shop application's payment options for an order.

Therefore, you can provoke a business error by selecting a credit card number starting with 3 or less, or a system error and by providing a bank account as a payment option. Figure 35 shows payment selection before submitting an order.

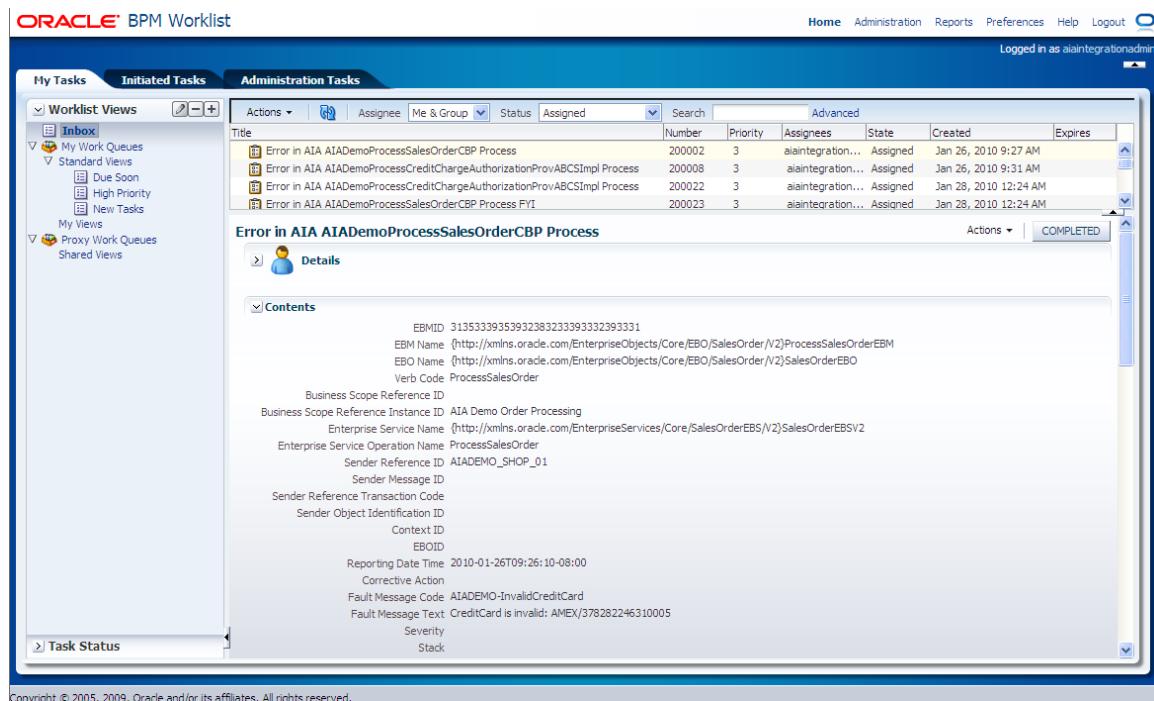


Figure 36: Error tracking within the Oracle BPM Worklist application

Figure 36 shows the Oracle BPM Worklist application, which is the default tracking tool for error notifications sent through AIA error handling.

The AIA Error Handling Framework is designed for extensibility in multiple ways:

- Define custom error handlers to take any type of action when an error occurs. For example, a custom error handler can be used to enrich the error message with additional details.
- Define additional error listeners to route error notifications to any tracking application.

Both approaches are implemented and visible in the Foundation Pack Demo. The custom error handler “AIADemoErrorHandlerExtImpl” is triggered each time an error occurs in the ABCS for the credit card validation. This custom error handler simply adds the entire payload of the failing service to the actual error message.

The custom error listener called “AIADemoErrorListener” is a simple SOA composite that subscribes to the AIA error handling JMS topic. In doing so, the process invoked by the custom error listener to route error notifications to another tracking application can be invoked each time a particular error occurs in the system. In this case, we want it to be invoked only for errors related to the credit card validation explained above. This is achieved through selection of specific criteria in the JMS adapter configuration of the AIADemoErrorListener SOA composite. The actual Foundation Pack Demo implementation of this custom error listener simply stores the error message in a file, but users can use this approach to plug in existing error tracking applications.

For more information about the AIA Error Handling Framework, see the *Oracle Application Integration Architecture Foundation Pack: Infrastructure Components and Utilities Guide*.

8. Appendix: Foundation Pack Demo Artifacts: Supported Concepts and Features

This appendix lists the Oracle Application Integration Architecture (AIA) Foundation Pack Demo artifacts that are delivered to support specific AIA architectural concepts and infrastructure features.

Supported Architectural Concept	Artifacts
Cross-Referencing	<p>Population:</p> <ul style="list-style-type: none"> • AIADemoSyncCustomerPartyListShopReqABCImpl • AIADemoSyncCustomerPartyListCRMProvABCImpl <p>Lookup:</p> <ul style="list-style-type: none"> • AIADemoProcessSalesOrderShopReqABCImpl • AIADemoQueryCustomerPartyCRMProvABCImpl
Domain Value Mapping	AIADemoCreateShipmentRequestUSPSPProvABCImpl
Fire-and-Forget Message Exchange Pattern	<ul style="list-style-type: none"> • AIADemoCreateShipmentRequestUSPSPProvABCImpl • AIADemoCreateShipmentRequestBatchProvABCImpl • AIADemoUpdateSalesOrderShopProvABCImpl • AIADemoNotifyCustomerProvABCImpl • AIADemoSyncCustomerPartyListShopReqABCImpl • AIADemoSyncCustomerPartyListCRMProvABCImpl • X12ProcessSalesOrderReqB2BCImpl • X12UpdateSalesOrderProvB2BCImpl
Synchronous Request-Reply Message Exchange Pattern	<ul style="list-style-type: none"> • AIADemoQueryCustomerPartyCRMProvABCImpl • AIADemoCreateRequestForQuoteWarehouseProvABCImpl • AIADemoCreateRequestForQuoteSupplierProvABCImpl • AIADemoProcessCreditChargeAuthorizationProvABCImpl
Enterprise Business Object Extension	<p>RequestForQuoteEBO/CreateRequestForQuoteResponseEBM used in:</p> <ul style="list-style-type: none"> • AIADemoProcessSalesOrderCBP • AIADemoCreateRequestForQuoteEBS • AIADemoCreateRequestForQuoteWarehouseProvABCImpl

Supported Architectural Concept	Artifacts
	<ul style="list-style-type: none"> AIADemoCreateRequestForQuoteSupplierProvABCImpl
XSL Extensibility	<ul style="list-style-type: none"> AIADemoProcessSalesOrderShopReqABCImpl AIADemoSyncCustomerPartyListShopReqABCImpl X12ProcessSalesOrderReqB2BCSImpl X12UpdateSalesOrderProvB2BCSImpl
Reference Abstract WSDL stored in Oracle Metadata Services (MDS) repository	All AIA artifacts (Adapters, Application Business Connector Services [ABCs], Enterprise Business Services [EBSs], Composite Business Processes [CBPs])
Application Business Connector Service and Adapter in the Same Composite	AIADemoQueryCustomerPartyCRMProvABCImpl
Uptake of Oracle SOA Suite 11g Business Rules	AIADemoProcessSalesOrderCBP
Uptake of Oracle SOA Suite 11g Human Workflow	AIADemoProcessSalesOrderCBP
Uptake of Oracle SOA Suite 11g Event Delivery Network	AIADemoJMSProducer
Uptake of Oracle SOA Suite 11g BPEL Bind Entity Feature	AIADemoProcessSalesOrderShopReqABCImpl
Content-Based Routing	AIADemoUpdateSalesOrderEBS
Message Queuing	<ul style="list-style-type: none"> AIADemoJmsProducer AIADemoJmsConsumer

Supported Infrastructure Feature	Artifacts
Business-to-Business	<ul style="list-style-type: none"> X12ProcessSalesOrderReqB2BCSImpl X12UpdateSalesOrderProvB2BCSImpl
Tracing/Logging	All ABCs
Error Handling	<p>All ABCs:</p> <ul style="list-style-type: none"> BPEL Error Handlers <p>All ABCs, EBSs, and CBPs:</p> <ul style="list-style-type: none"> Fault policy assignment <p>AIADemoProcessCreditChargeAuthorizationProvABCImpl (System Fault)</p> <p>AIADemoProcessSalesOrderCBP (Business Fault)</p>
Custom Error Handler	Class oracle.apps.aia.demo.AIADemoErrorHandlerExtImpl in project AIADemoErrorHandler: Adds full service payload to fault message
Custom Error Topic Subscriber	AIADemoErrorListener: Subscribes to fault type

Supported Infrastructure Feature	Artifacts
	AIADEMO_CREDITCARD_FAILURE and writes the fault message to a file
Composite Application Validation System (CAVS) Support	All ABCSs
Pre-seeded CAVS Test and Simulator Definitions	All ABCSs and AIADemoProcessSalesOrderOrchestrationEBS
Translation Support	AIADemoNotifyCustomerProvABCSImpl: <ul style="list-style-type: none"> Embedded Java code uses Java resource bundle oracle.apps.aia.demo.AIADemoNotificationResourceBundle
Service Constructor Uptake	All ABCSs

Index

ABCSSs. *See* Application Business Connector Services

AIA. *See* Application Integration Architecture

AIA Demo Artifacts

- harvesting, 33

Application Business Connector Services, 15

- creating, 28, 44

Application Integration Architecture

- infrastructure components, 47
- introduction, 9
- reference architecture, 14

CAVS. *See* Composite Application Validation System

CBPs. *See* Composite Business Processes

Composite Application Validation System, 48

Composite Business Processes, 16

- creating, 32

EBFs. *See* Enterprise Business Flows

EBMs. *See* Enterprise Business Messages

EBOs. *See* Enterprise Business Objects

EBSs. *See* Enterprise Business Services

Enterprise Business Flows, 16

Enterprise Business Messages, 14

- creating, 23, 24, 26

Enterprise Business Objects, 14

- extending, 43

Enterprise Business Services, 14

creating, 23, 24, 26

extending, 45

error resolution, 50

Foundation Pack

- infrastructure components, 47
- overview, 12

Foundation Pack Demo, 17

- architecture, 18
- creating, 23
- creating Application Business Connector Services, 28
- creating Composite Business Processes, 32
- creating Enterprise Business Services and messages, 23, 24, 26
- deploying, 34
- extending, 43
- harvesting AIA Demo Artifacts, 33
- order booking process, 17
- running, 37
- supported concepts, 53
- supported features, 53
- infrastructure components, 47
- logging, 50
- Oracle Enterprise Repository, 47
- process flows
- extending, 44

reference architecture, 18