Oracle® Application Integration Architecture Foundation Pack 2.1: Release Notes

Release 2.1

Part No. E12824-01

July 2008

ORACLE

Oracle Application Integration Architecture Foundation Pack 2.1: Release Notes

Part No. E12824-01

Contents

Release Notes for Oracle Application Integration Architecture Foundation Pack 2.1 Preface

This document provides an overview of the value proposition that is associated with the new features and enhancements that are planned for Oracle Application Integration Architecture Foundation Pack 2.1. It is a roadmap that is intended to help you assess the business benefits of the Foundation Pack and to plan your information technology (IT) projects and investments.

This preface discusses:

- Release information publications
- Additional resources

Oracle Application Integration Architecture Foundation Pack Release Information Publications

This section discusses four publications that provide in-depth technical and functional information that is available at the time of publication:

1. Statement of direction
2. Release value proposition
3. Prerelease notes
4. Release notes

Statement of Direction

The statement of direction is published 9 to 12 months before a release. It provides a high-level overview of the major focus of product development efforts, enabling high-level business decision makers to begin preliminary upgrade planning.

Release Value Proposition

The release value proposition provides more functional details than the statement of direction, identifies major enhancements, and articulates the expected business benefit. This document is designed to help you to determine whether new product features might warrant upgrading from an old release or embarking on a new PeopleSoft implementation. With this information, managers will be able to initiate preliminary budget planning and begin putting together a project team to further evaluate specific PeopleSoft products. The release value proposition is published 6 to 9 months before a release.

Prerelease Notes

Prerelease notes provide more functional and technical details than the release value proposition. This document describes how each enhancement functions within the context of the greater business process. This added level of detail should enable project teams to answer the following questions:

- What out-of-the-box functionality will change?
- What customizations may be affected?
- How will an upgrade or new implementation affect other systems?
- How will these changes affect the organization?

After the project team has reviewed and analyzed the prerelease notes, business decision makers should be in the position to determine whether to allocate budget and initiate implementation plans.

Prerelease notes are published approximately 3 months before a release.

Release Notes

Release notes are published when the products are generally available (GA) and validate the final scope of the release. The release notes discuss the features and enhancements that are available with the GA release of each product, describing the finalized functional and technical details that will enable project teams to confirm budgets and complete implementation plans.

Oracle Application Integration Architecture Foundation Pack Value Proposition

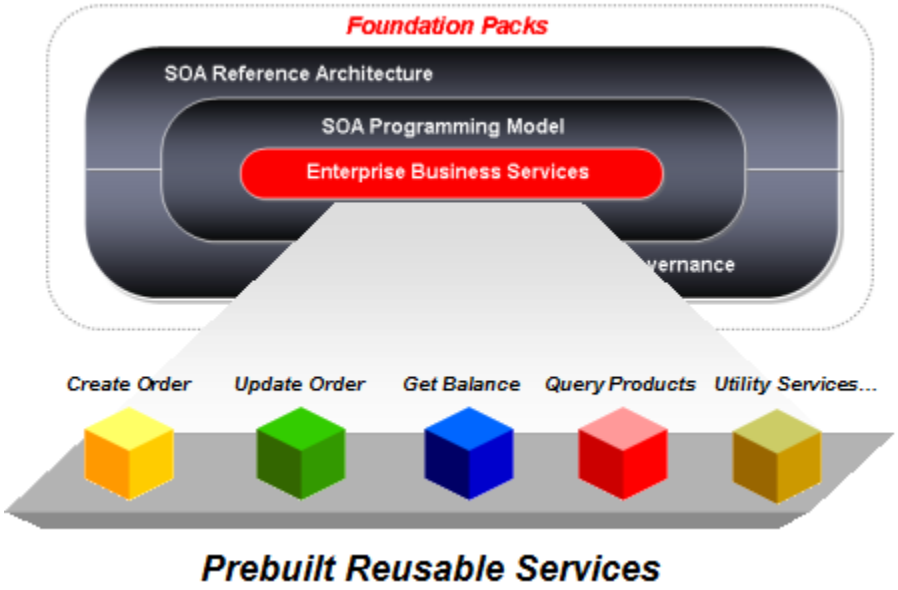
The new features and enhancements that are included in this release are grouped by release themes then by product area. Our goal is to help organizations leverage technology to its fullest and increase the efficiency and effectiveness of operations. Please note that the final release may not have every feature that is discussed in this document, and a specific feature may become part of a different application or have a product name that is different from those cited in this document.

Oracle proudly announces Oracle Application Integration Architecture Foundation Pack 2.1. This new release demonstrates Oracle's commitment to:

- Reducing total cost of ownership
- Mitigating risks using proven methodology
- Accelerating adoption and migration to enterprise SOA

Reducing Total Cost of Ownership

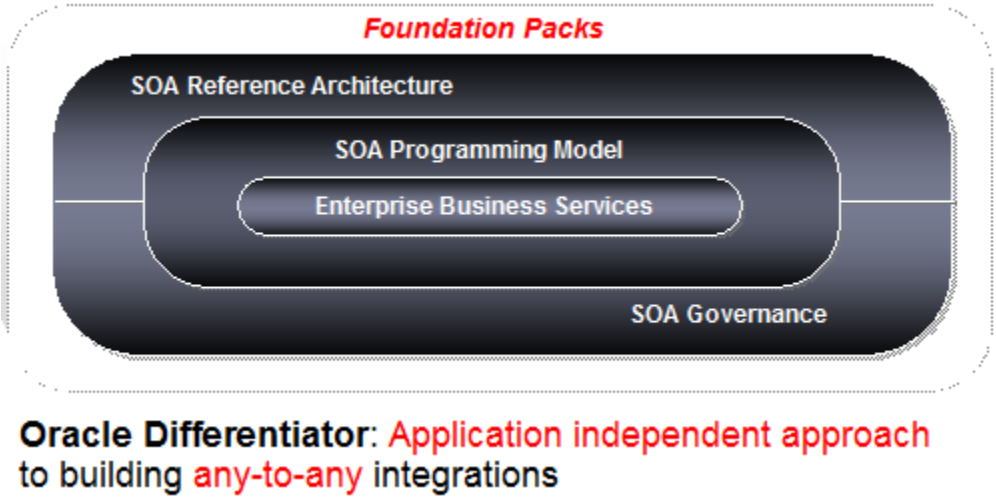
Foundation Pack's pre-built set of Enterprise Business Objects and Services, integration management infrastructure, and proven methodology significantly lowers cost of ownership and provides a faster time to value for new composite business processes.



Reducing the total cost of ownership

Proven Methodology to Mitigate Risks

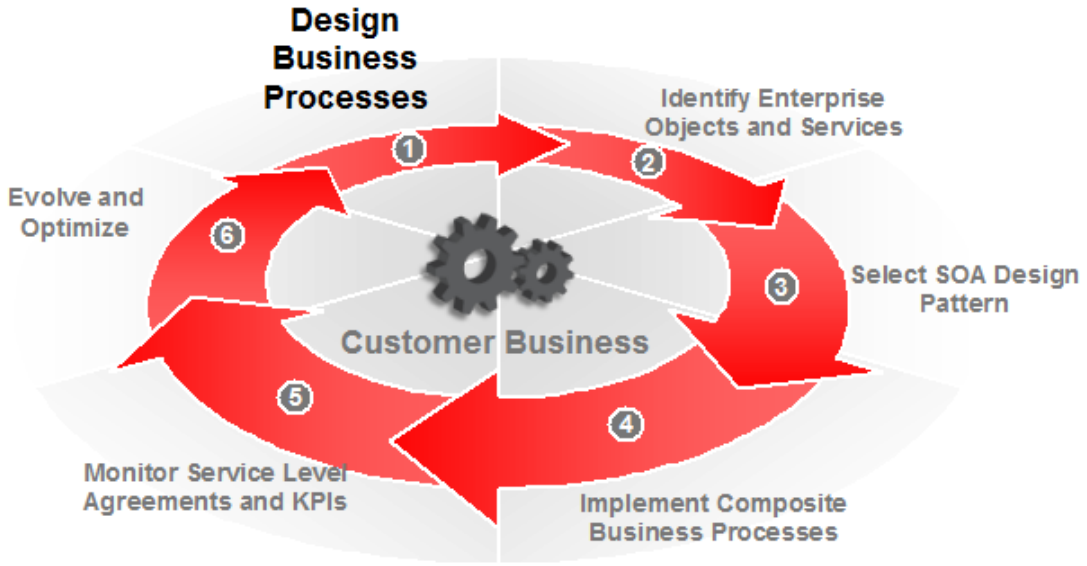
The Foundation Pack delivers the same methodology that Oracle uses to deliver its process integration packs, providing customers with a set of proven best practices and design principles to build their mission-critical business processes with reduced risk and cost.



Oracle's proven methodology

Accelerate Adoption and Migration to Enterprise SOA

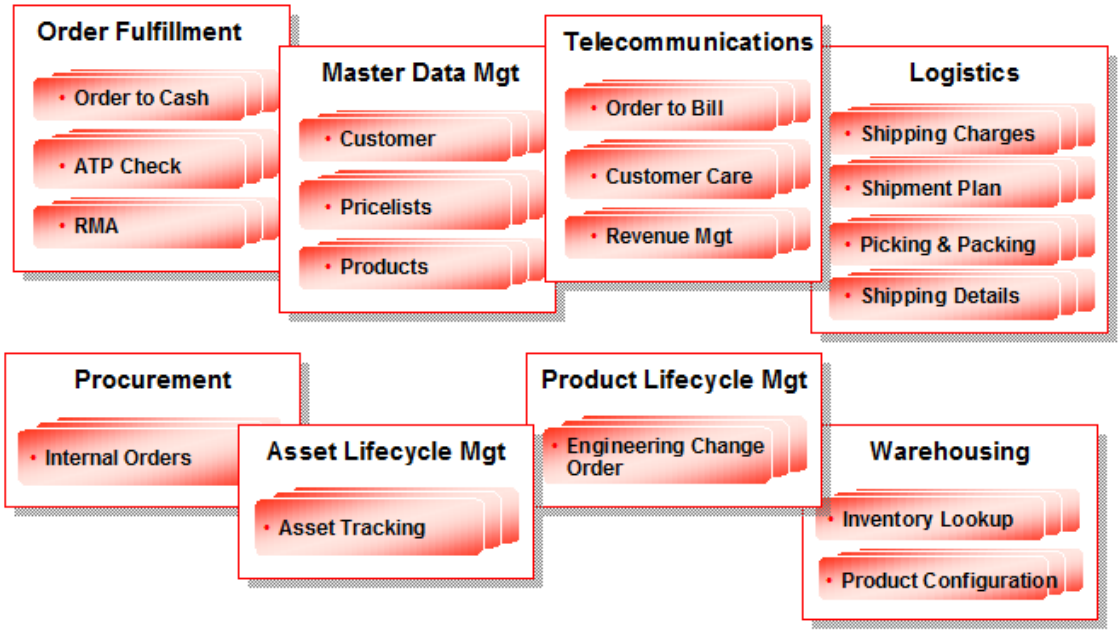
While everyone understands and agrees with the benefits of SOA, many customers do not have the datapoints or in-house skills and expertise needed to implement SOA, for example, creating reusable web services. Foundation Pack contains exactly what organizations need for SOA based integrations:



A SOA approach

AIA Foundation Pack delivers a proven service-oriented integration approach to implement agile and adaptable business processes and a robust, proven, and reliable reference architecture for SOA based integrations.

AIA Foundation Pack delivers a ready to use set of common data models for frequently required business entities and their related services.



Library of enterprise business services

AIA Foundation Pack delivers a rich set of additional infrastructure components to enable implementation, operation, maintenance, and governance of SOA.

Product Enhancements for Oracle Application Integration Architecture Foundation Pack 2.1

This section discusses new enhancements for the Oracle Application Integration Architecture Foundation Pack 2.1. These enhancements fall into four main areas:

- AIA Programming Model
- Business Service Repository
- Composite Application Validation System
- Enterprise Object Library: Enterprise Business Services, Enterprise Business Objects, Enterprise Business Messages

AIA Programming Model Changes

AIA delivers the following features and enhancements in Foundation Pack 2.1:

- ESB routing rules extensibility
- Application Business Connector Services extensibility
- BPEL Process Extensibility
- Transformations
- Reusability
- Bulk processing
- AIA design pattern samples
- AIA security

ESB Routing Rules Extensibility

AIA Foundation Pack 2.1 delivers the ability to protect the additions / modifications done to delivered routing rules by the customers.

Application Business Connector Services Extensibility

AIA Foundation Pack 2.1 allows customers to augment the ABCS functionality in a non-intrusive manner. Custom code behaviors can be executed at predefined points in an Application Business Connector Service (ABCS). This will:

- Allow customers to make custom alterations to ABM / EBM prior to returning responses to source application
- Allow customers to augment the delivered data enrichments
- Enable customers to make custom validations prior to invocation of Enterprise Business Services

- Allow the ABCS to be more durable
- Mitigate the need for changing the ABCS

The architecture provides recommendations on how the Application Business Service Connectors as well as the orchestration processes can be designed to allow customers to introduce the extensions to augment the functionality.

Each of the BPEL processes can have its own set of extensibility points based on the functional needs. You can implement the interface that will be defined for each of the extensibility points to augment the behavior. Application Business Connector Services are recommended to have a minimum of four extensibility points, in the case of request / response pattern, to allow customers to inject additional behavior. In the case of fire-and-forget pattern, the services are expected to have a minimum of two extensibility points. The orchestration processes might decide not to have any extensibility points.

Extension points are provided to the delivered BPEL processes to enable extensibility.

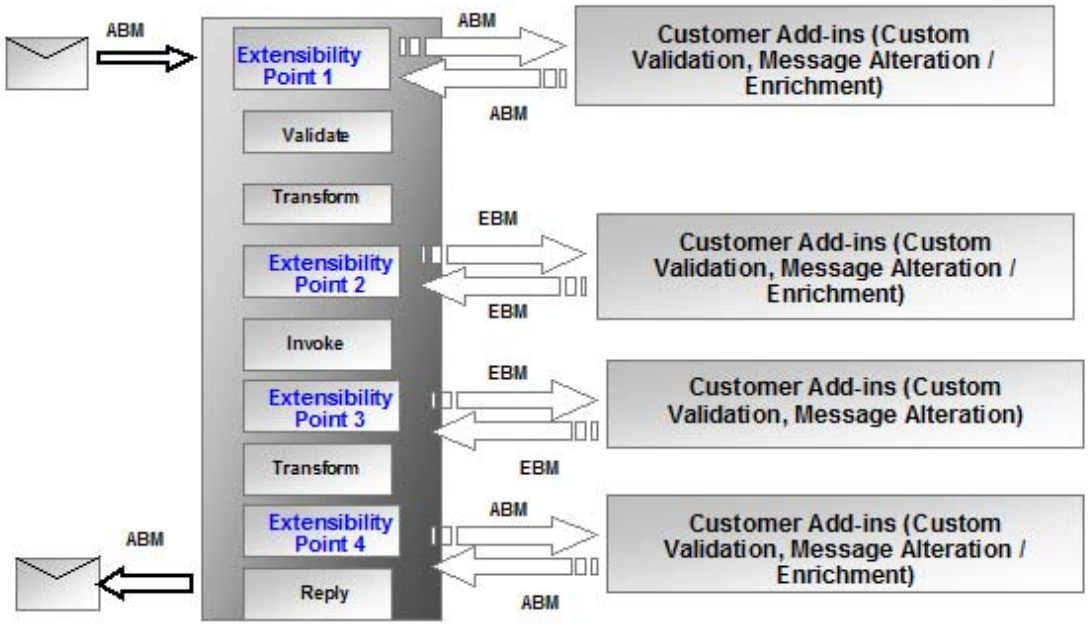
- For an ABCS, named extension points will be provided out of the box.
- For other BPEL processes, namely EBF, extension points will be based on the business requirements.

Customers only define the custom behavior - ABCS Architecture dictates the execution.

Extensions are upgrade safe and define a structured programming model. ABCS Architecture allows custom code (behaviors) to be executed at pre-defined points in the flow. The custom code augments the standard functionality available in a typical ABCS.

Making ABCS Extensible

This diagram illustrates the extensibility methodology adopted by the ABCS:



Extension-aware ABCS

Requestor ABCS can potentially have the following extensibility points

- Prior to the transformation of ABM to EBM
- Prior to the invocation of Enterprise Business Service
- Prior to the transformation of EBM to ABM
- Prior to the invocation of callback service or response -return

Provider ABCS can potentially have the following extensibility points

- Prior to the transformation of EBM to ABM
- Prior to the invocation of Application Service
- Prior to the transformation of ABM to EBM
- Prior to the invocation of callback EBS or return of response message

In case of fire-and-forget pattern, only the first two extensibility points will be made available. ABCS author can decide to introduce additional extensibility points however; caution needs to be exercised when introducing additional Extensibility Points.

ABCS Extension Enabling – Process Overview

1. Identify the extension points for the ABCS.
2. Define the Service Contract (abstract WSDL) for the ABCS specific extension service.

   Leverage the WSDL templates provided by the AIA Architecture team.

3. Add configuration properties (one for each extension point) to the AIAConfigurationProperties.xml that allow customers to specify whether they have implemented the interface to inject additional behavior or not.

```
<!-- CreateCustomerSiebelReqABCSImpl -->
   <ServiceConfiguration
serviceName="{http://xmlns.oracle.com/CreateCustomerSiebelReqABCSImp
l}CreateCustomerSiebelReqABCSImpl">
         <Property
name="ABCSExtension.PreProcessABM">false</Property>
         <Property
name="ABCSExtension.PreProcessEBM">true</Property>
         <Property
name="ABCSExtension.PostProcessEBM">false</Property>
         <Property
name="ABCSExtension.PostProcessABM">false</Property>

   </ServiceConfiguration>
```

4. At each of the extension points, ABCS needs to be conditionally programmed to invoke the extension specific operations.
5. Using abstract WSDL, create a concrete WSDL with bindings pointing to a dummy servlet (provided by FP) to facilitate QA testing.

6. Configure *WSDLRuntimeLocation,* partner-link property, to point to the concrete WSDL.
7. During Implementations, customers will create their own Concrete WSDL for their service and have it placed in *WSDLRuntimeLocation.*

BPEL Process Extensibility

BPEL Process Extensibility allows customers to augment a set of processing steps present in a delivered BPEL process. This is implemented through pre-defined extensibility points with well-defined interfaces included in BPEL processes during development at Oracle.

Transformations

AIA Foundation Pack 2.1 provides refinements to AIA 2.0 extension methodology. The main goal is to improve the developer's productivity and performance through templates.

Reusability

AIA Foundation Pack 2.1 provides architectural recommendations and guidelines for designing AIA Application artifacts to achieve the following goals:

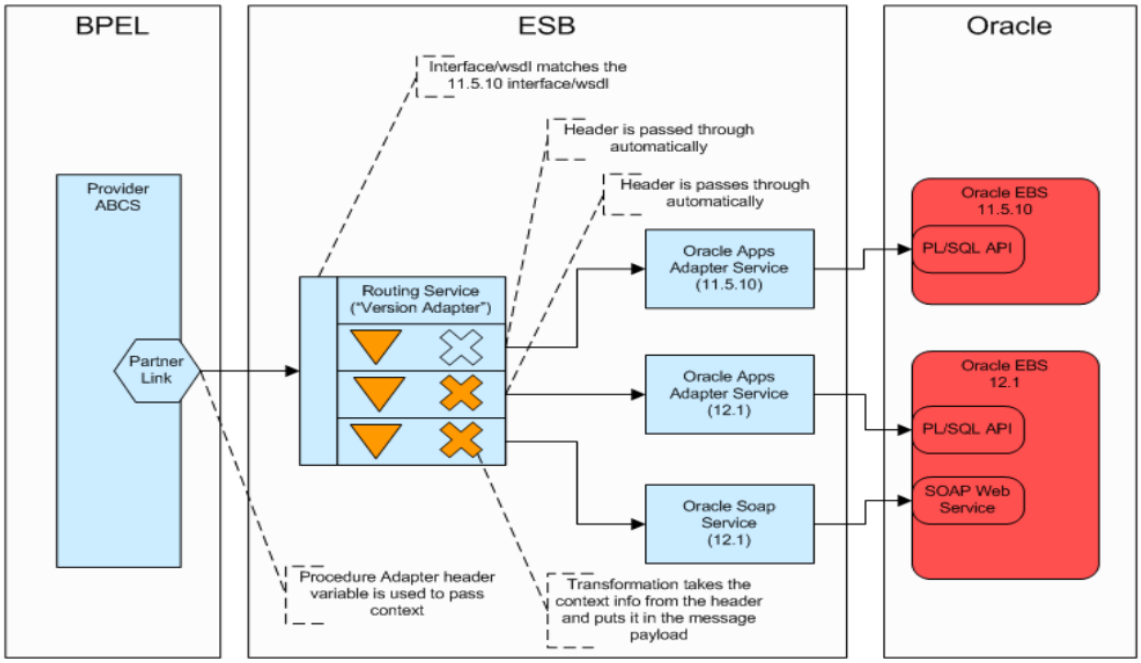
- Faster delivery of Process Integration Packs (PIPs)
  - New PIPs or enhancements to an existing PIP
    - On newer versions of participating applications
    - On new participating applications
    - Various Industries
- Increasing the durability (prolonged longevity) of AIA artifacts of the PIPs as well as the application artifacts
  - Increasing the ability of application artifacts to support multiple PIPs
  - Eliminating / Minimizing the impacts of evolution of an artifact on others
- Leveraging ABCS across PIPs, application versions
- Leveraging core services across verticals
- Providing methodology on reusability of various artifacts
- Providing methodologies to participating applications on how to build durable services

Designing an ABCS for Reusability

To design reusable ABCS, follow these guidelines:

- ABCS should be decoupled from the target system. For example,
  - Siebel Requestor ABCS should make no assumptions about the target system
  - Your Requestor ABCS could invoke a provider ABCS that was not built by you
  - Your Requestor ABCS could be replaced by a requestor ABCS not built by you

- ABCS should not be PIP specific
  - ABCS should not make any design decisions based on the PIP for which it is currently built
  - For example, EBM Population should not be based on target system or a PIP
- ABCS should not be bound to a specific application version
  - An ABCS can be used to interact with multiple application versions
- ABCS should be transport neutral - ABCS should be communicating inbound as well as outbound only via SOAP and ESB bindings
- ABCS should be programmed to interact with multiple instances of a single application
  - A single SiebelUpdateCustomerProviderABCS should be able to support multiple Siebel instances at a customer site



Facilitating ABCS support against multiple application versions

Bulk Processing Using Oracle Data Integrator

Bulk processing is processing of a batch of discrete records between participating applications. AIA is standardized on Oracle Data Integrator (ODI) as the tool of choice for bulk data integrations. Bulk processing does not involve usage of the Enterprise Business Objects, Enterprise Business Services, or a loosely coupled architecture.

AIA delivers a Knowledge Module for handling ESB cross-references and the update/delete of source records at post processing. Bulk processing is integrated to the Foundation Pack error-handling framework.

Bulk Data Integrations

ESB Cross Referencing

- All AIA scenarios that require a cross referencing function are eligible.
- Like any other ODI knowledge module (KM), the ODI ESB KM must be imported to implement cross-referencing.
- Populating the ESB Cross reference table is transparent to users and optimized for usage in ODI

Update / Delete of Source Records

- After successful processing of the data from source to target application, the source data can be updated, for one or more columns, or purged.
- Optional feature and depends on the business requirement. Example is Update of processing status column after a successful transfer and so forth.

AIA Design Pattern Samples

AIA Foundation Pack 2.1 provides samples to explain the constructs as well as best programming practices. These are easy to deploy and run with no dependency on participating applications and get developers quickly up to speed on the key constructs.

Samples Shipped with Foundation Pack 2.1

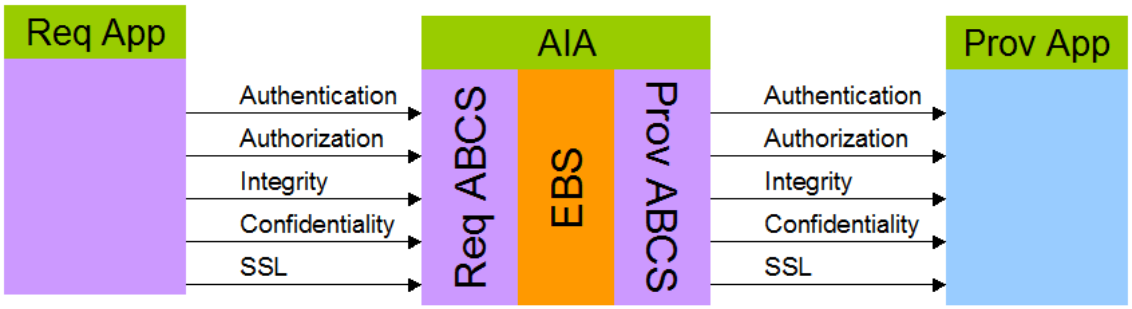
The samples shipped with Foundation Pack 2.1 are:

- Base Sample
- JMS Sample
- Routing Rules
- Synchronous Request/Response
- Extensibility
- EBF
- Asynchronous Request/Response
- WS Authentication

AIA Security

AIA Security provides security for both connecting to AIA as well as propagating security information from application to application. It uses Oracle Web Service Manager (WS) policies to configure security declaratively and XACML standards to propagate security information between applications.

This diagram illustrates the high-level security architecture:

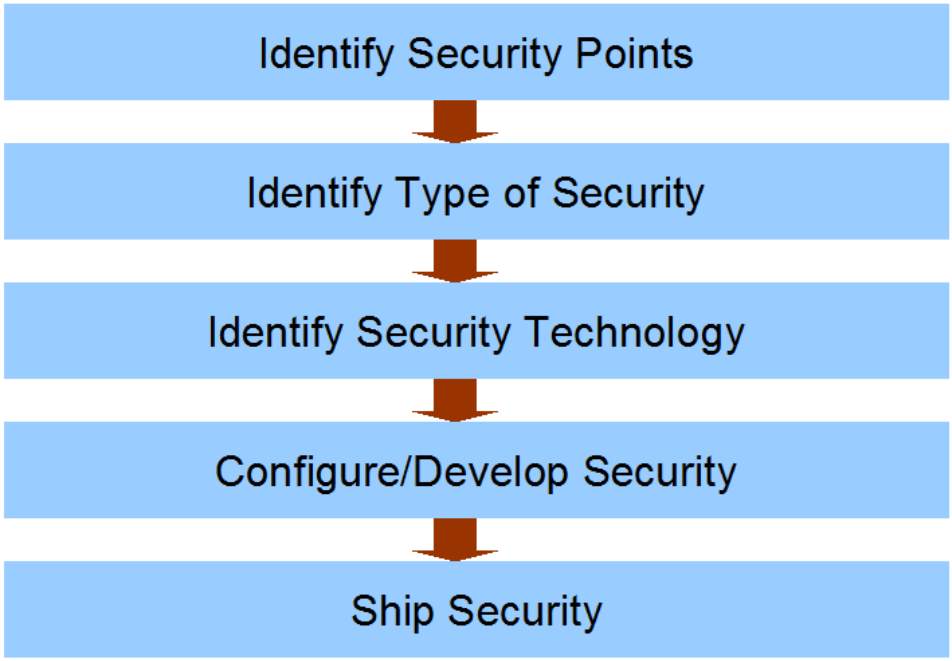


High-level security architecture

ABCS is not responsible for authentication or for generating credentials for the authentication request.

Enabling Security in AIA

This diagram illustrates the steps for enabling security in AIA:



Enabling security in AIA

Application to AIA Security Architecture

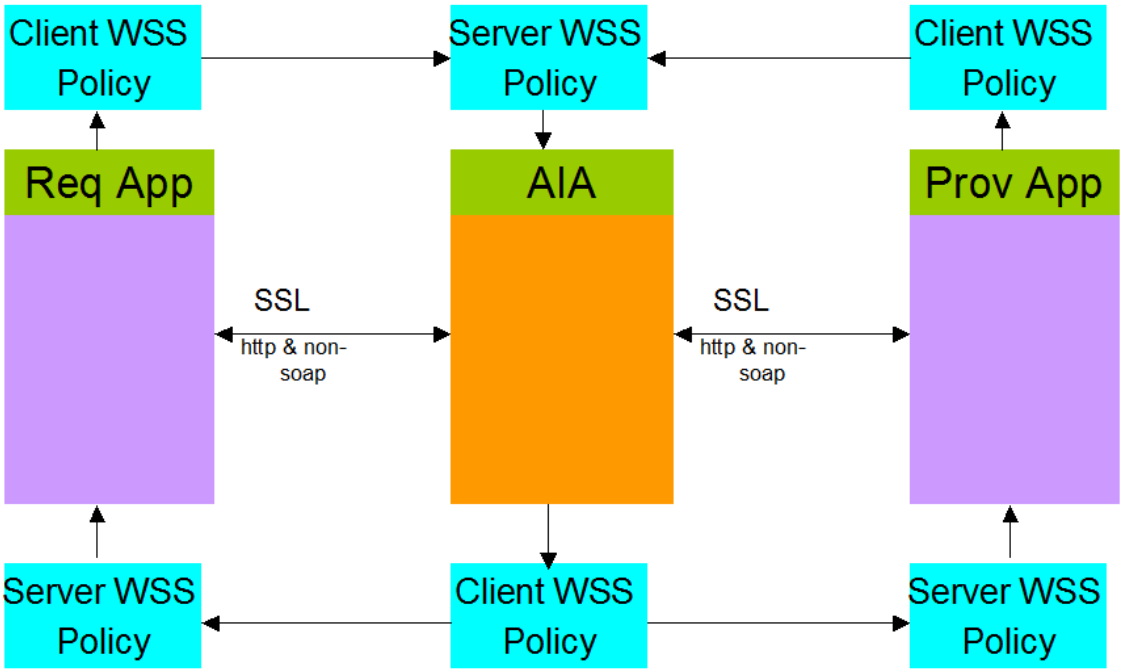
WS Security defines a standard set of SOAP extensions that enable applications to exchange secure messages. The extensions provide authentication, integrity, and confidentiality. WS Security leverages existing standard technologies such as x.509, XML signature and XML encryption and provides a mechanism for associating security tokens with message content. It is extensible to support multiple security token formats and can define different formats for different parts of the message.

SSL securely communicates between any two applications. It secures the channel as opposed to the message and is used for authentication, confidentiality and integrity.

WS-Policy provides a framework for specifying and discovering the capabilities, requirements, and general characteristics of a Web service. The properties are defined as policies.

Security Assertion Markup Language (SAML) is a framework for exchange of security-related information assertions. These assertions about authentication and authorization are expressed as XML documents. This is used for identity federation and provides technology to allow a business to securely interact with users originating from its vendors, suppliers, and customers.

This diagram illustrates the application to AIA security architecture:

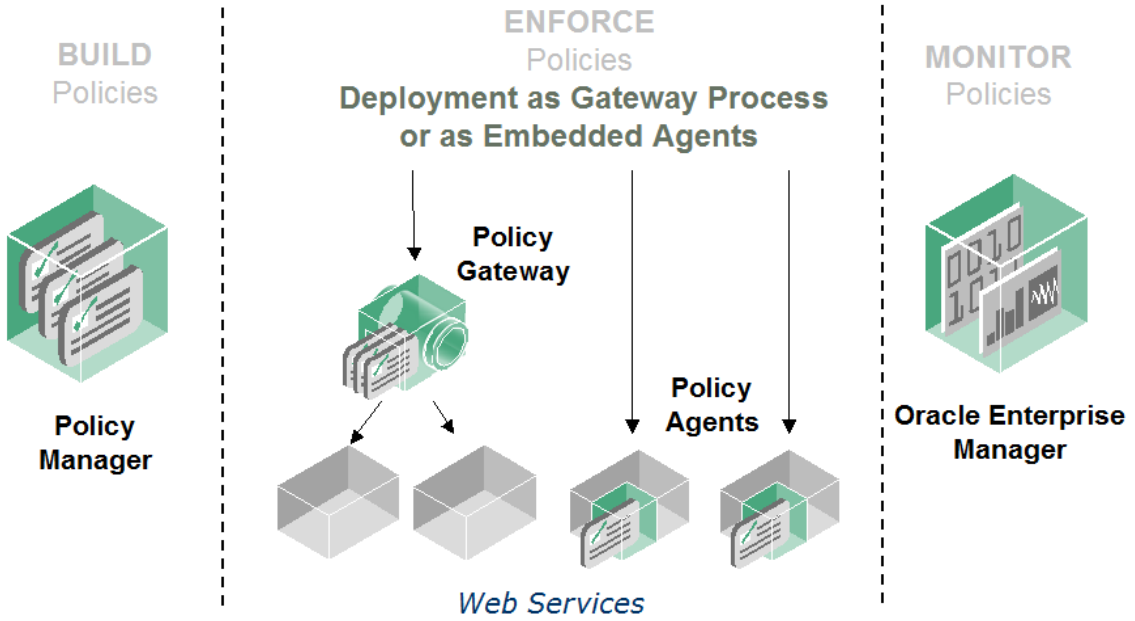


Application to AIA security architecture

Oracle Web Service Manager

Oracle Web Service Manager (OWSM) is a standalone platform for securing and managing access to web services. OWSM can be used by a developer, deployer, or security administrator, however it does not require developers to modify applications or services (no programmatic security necessary). OWSM executes security policies in real time and monitors all access-control events (graphical reports). It provides tools for defining and monitoring service-level agreements (SLA) and leverages backend identity management infrastructures such as Oracle Access Manager for authentication and authorization.

This diagram illustrates OWSM security:

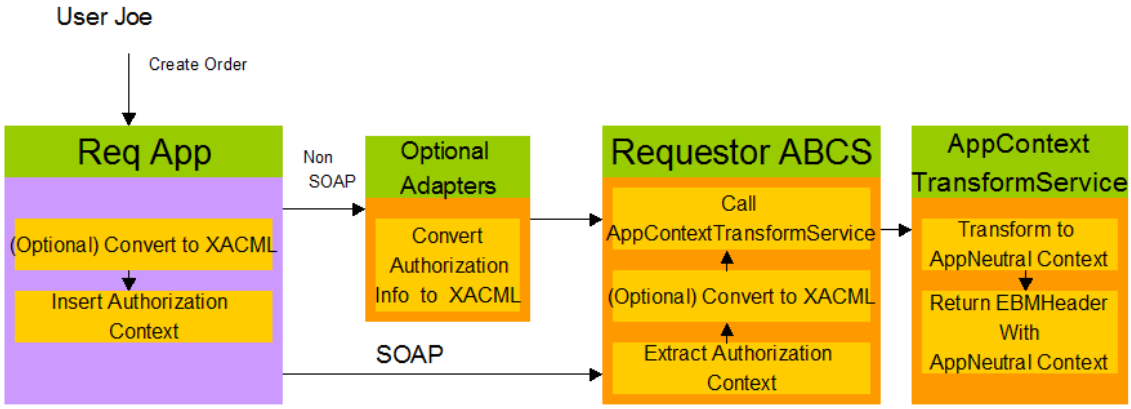


OWSM security

Requestor AppContext Architecture

The preferred approach is for the requestor application to send authorization information in XACML format. The requestor application sends the information in a SOAP header or business message and the requestor ABCS formats the information into the XACML structure. For non-soap participating applications, an adapter will receive the message in a custom way and format the information into XACML.

This diagram illustrates the Requestor AppContext flow:

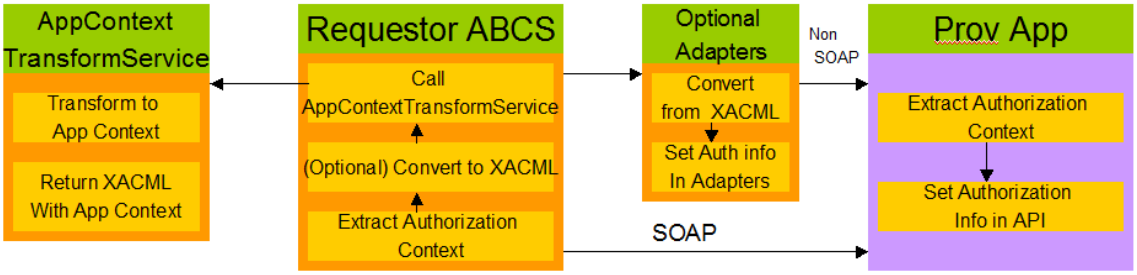


Requestor AppContext flow

Provider AppContext Architecture

The preferred approach is for the Provider ABCS to send app context in XACML format in a SOAP header or business message. For non-SOAP participating applications, an adapter will receive the message XACML from the provider ABCS and set the context in custom headers or a custom way and call the application.

This diagram illustrates the Provider AppContext flow:



Provider AppContext flow

AppContext Mapping Service

The AppContext Mapping Service is implemented as an external service per participating application. AIA recommends using BPEL with co-location to implement this service. ABCS should call this service using dynamic partner link. The AppContext Mapping Service is responsible for populating additional values needed for standard or application context including:

- Request TransformToAppContext (EBMHeader)
- Request TransformToAppNeutralContext(Request)

AIA provides the WSDL for the service.

Business Service Repository

The Business Service Repository (BSR) is a Service-Oriented Architecture (SOA) repository. In support of SOA governance, the BSR provides visibility into the entire SOA portfolio across AIA's SOA integration layer. Specifically, BSR contains the following as its first-class contents:

- **Services**: Enterprise Business Services and Application Business Connector Services
- **Objects**: Enterprise Business Objects and Enterprise Business Messages
- **Integration Scenarios**: Cross-application message flows that traverse through multiple hops of AIA services

BSR captures the assets' metadata, relationships, and runtime invocation dependencies in context of business process logics. It mirrors AIA's SOA architecture and reflects the end-to-end process implementation across heterogeneous applications.

As a SOA repository, BSR complements and augments the Universal Description, Discovery, and Integration (UDDI) functionality provided by Oracle Service Registry (OSR). It consistently synchronizes contents between the repository and the registry. By combining the repository and the registry, BSR offers comprehensive visibility coverage to AIA's SOA landscape throughout the SOA lifecycle: from abstract contents during design-time to concrete assets at runtime.

The primary features of BSR include:

- Mechanisms to publish and remove contents into the BSR and the UDDI registry in bulk or single modes.
- A user interface to search and to browse individual SOA assets in AIA's ecosystems.

- A user interface to locate integration scenarios, and thus to discover and to learn the invocation dependencies across the SOA assets in context of business process logics.
- A user interface to annotate the SOA portfolios via flex-fields.

BSR covers needs for a wide spectrum of users throughout SOA lifecycle, including functional and business analysts, architects, developers, system integrators, and system administrators.

As part of the AIA Foundation Pack 2.1, the BSR delivers the following features and enhancements:

- BPA Suite Integration
- Capturing services (WSDLs) as first class objects in BSR
- Capturing EBO (XSDs) as first class objects in BSR
- Fine-grained Search Capabilities
- Service-based Functional Impact Analysis
- Flex-fields
- Integration Scenario Enhancements

BPA Suite Integration

Read-only horizontal business process models are delivered as part of the AIA Foundation Pack 2.1. They are viewable through the BPA Publisher and present a decomposed, top-down navigation path from high-level process models all the way to low-level, fine-grained, and detailed process activities. As part of the Foundation Pack 2.1, SOA portfolios stored in BSR are linked to the BPA models in a context-sensitive manner, This BSR and BPA integration provides a unified visibility spanning across from functional process models all the way to actual SOA assets in production environments. In so doing, this feature bridges gaps between the business analyst user communities to the operational IT user groups, and renders actual executables to business users in an intuitive, straightforward, and context-sensitive manner.

This diagram illustrates L3 in the BPA Publisher. A drop-down list is provided on respective models or activities that links users to entries in the BSR, including EBOs, EBS, and integration scenarios.

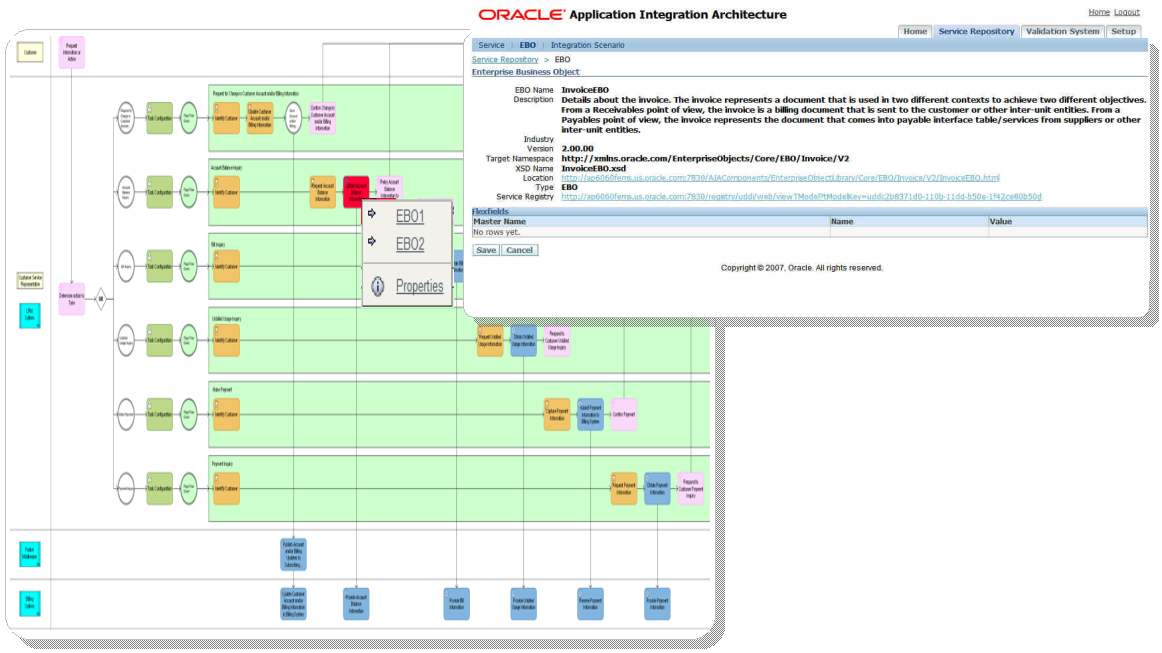


Illustration of L3 in the BPA Publisher

Capturing services (WSDLs) as First Class Objects in BSR

Services, including Enterprise Business Services and Application Business Connector Services, are of critical importance to AIA process integration. Their abstract definitions are captured and surfaced through BSR UI as first-class contents. Service entries are searchable and viewable. Their internal structures and metadata, including pre-defined annotations and user-defined flex-fields, are rendered on BSR UI. Further, corresponding service entries in UDDI are also linked in a context-sensitive manner depending on the exact SOA lifecycle stage.

- Services are published into BSR via the AIA installer or command-line script
- Services can be removed in single or bulk manners via command-line script

While the BSR UI only renders the abstract view, it does provide linkages to corresponding entries in UDDI/OSR. Depending on lifecycle stage, this may result in one-to-map correspondences between the service definition in repository/BSR, and the services in the registry/UDDI.

The BSR offers the following feature enhancements”

- Abstract Interface Definitions

  Mirroring abstract WSDL. It captures:

  - Signatures: Messages/EBMs
  - Patterns: MEP & Interface/portType
  - Operations
- Complements the UDDI

  Hyperlinks to UDDI assets

  - Design-time:

    Entries in the **shipped** taxonomy

  - Runtime:

    Entries in the **deployed** taxonomy

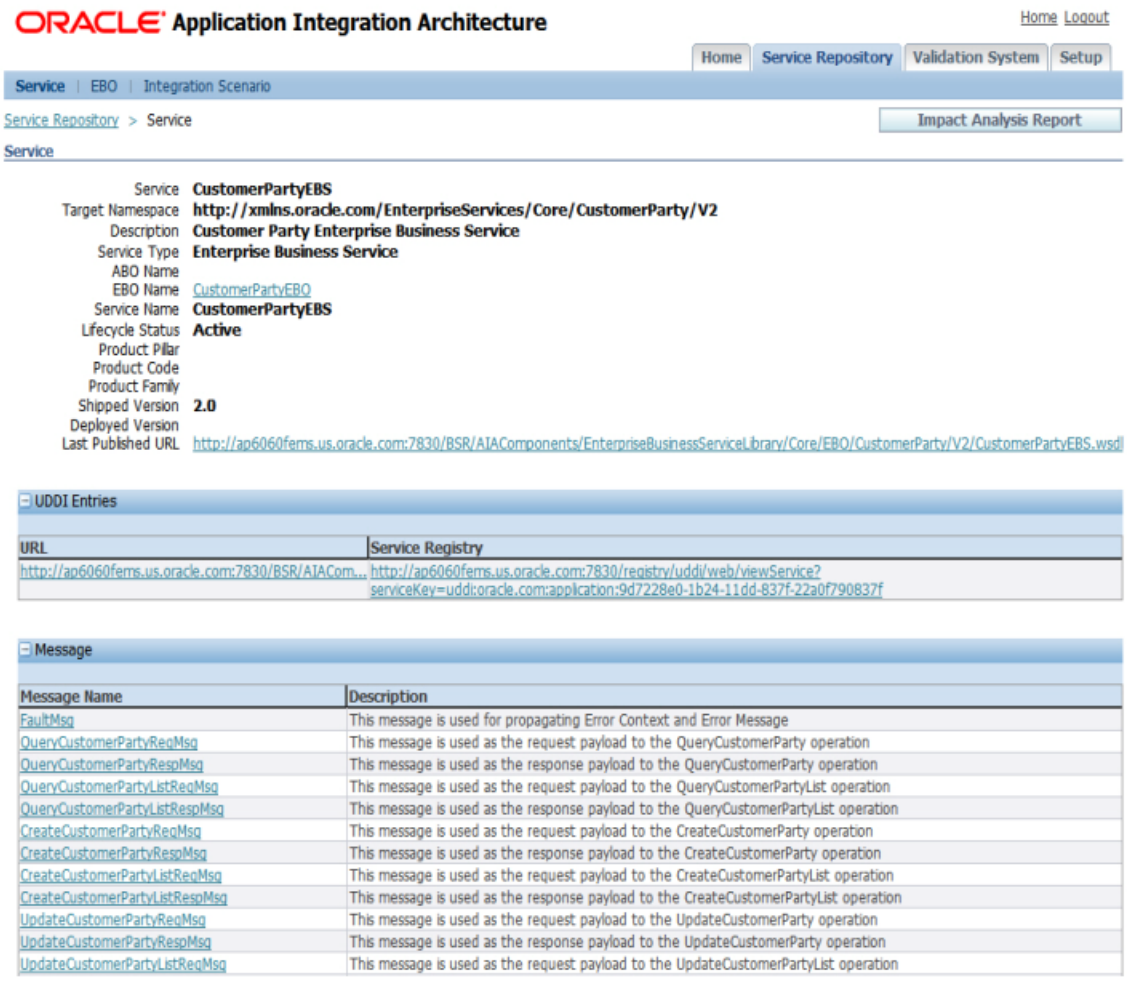
- Linkage to EBO

  Associate services to their main objects

- Impact Analysis

  Answering 'what-if' questions related to service interface definition changes

This page illustrates a BSR service entry:



BSR rendering of a service

Example

You might want to locate all services implementing the 'Query' verb that are provided from the 'Siebel' pillar. Further, this service uses the 'CustomerPartyEBO'. The screenshots provide you the search result. This illustrates how BSR would promote visibility across the AIA ecosystem regardless of who may have originally developed the services, and it also promotes reuse and reduces duplicated development efforts.

Sample query: assessing all available services, which

- Relate to Siebel
- Implement Query verb
- Manipulate CustomerParty EBO

Resulting filters:

- Product Family: Siebel
- Operation: Query%
- EBO Name: CustomerParty%

Result of search

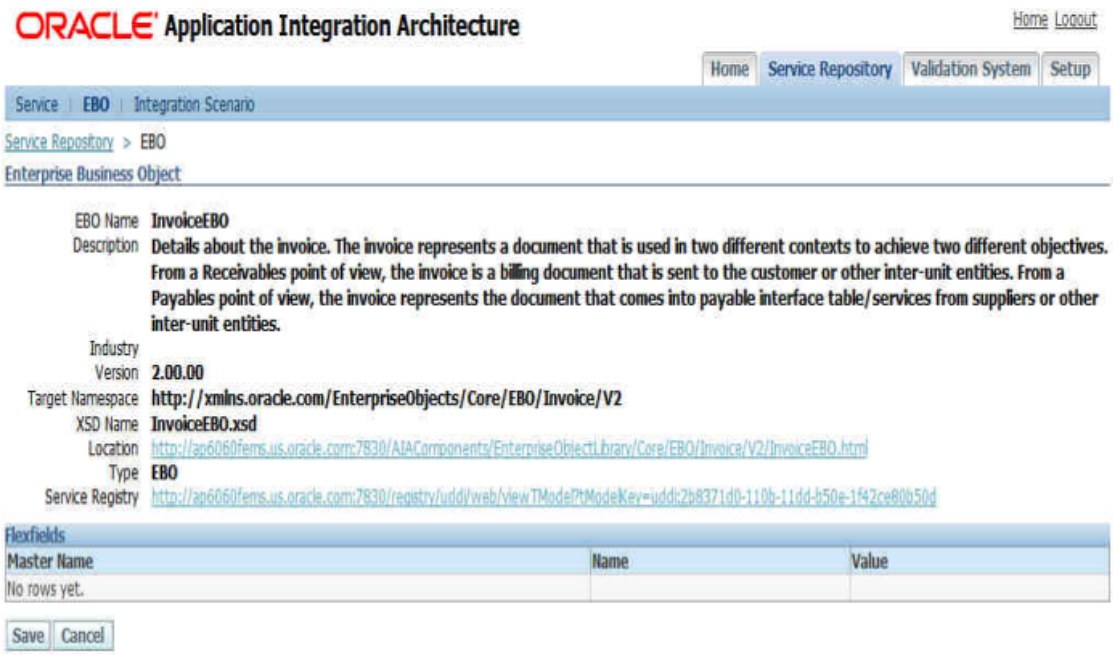
Benefits

- Promote service reuse
- Reduce duplicated service development efforts

Capturing EBO (XSDs) as First Class Objects in BSR

Enterprise Business Objects are at the heart of the AIA canonical layer that constitutes the best-in-class definition of business semantics across Oracle applications and relevant industry standards. BSR stores EBOs, EBMs, and underlying XSD component models. Specifically, EBOs and EBMs are surfaced on BSR UI as first-class contents. The UI provides further hyperlink to its XSD definition either on the file system or at an URL location. Respective UDDI/OSR entries are also accessible via the BSR UI.

From the BSR repository EBO/EBM entries, we link to their corresponding entries in the UDDI. It is always one-to-one correspondence between the repository and registry entries.



EBO page

Links to the functional documentation provides the object's internal nesting structure.

Invoice

EBO Type	Enterprise Business Object
CCTS Type	Aggregate Business Information Entity
CCTS Dictionary Entry Name	Invoice. Details
Description	Details about the invoice. The invoice represents a document that is used in two different contexts to achieve two different objectives. From a Receivables point of view, the invoice is a billing document that is sent to the customer or other inter-unit entities. From a Payables point of view, the invoice represents the document that comes into payable interface table/services from suppliers or other inter-unit entities.
Usage Rule	
Extends	
Attributes	Description TotalAmount ExtendedAmount PriorBalanceAmount AdjustedIndicator TypeCode GroupCode BillingFrequencyCode ConsolidationIndicator InvoiceDateTime

Child Components	Name	CCTS Dictionary Entry Name / Description
	Identification	Invoice. Identification. Identification Invoice Identification.
	PaymentTerm	Invoice. Payment Term. Payment Term Details about the payment terms for the invoice.
	Note	Invoice. Note. Note Note about the invoice
	Status	Invoice. Status. Status Details about the the status of the invoice
	EffectiveTimePeriod	Invoice. Effective Time Period. Effective Time Period For a recurring invoice, identifies the start and end date for recurring billing

EBO details

Fine-grained Search Capabilities

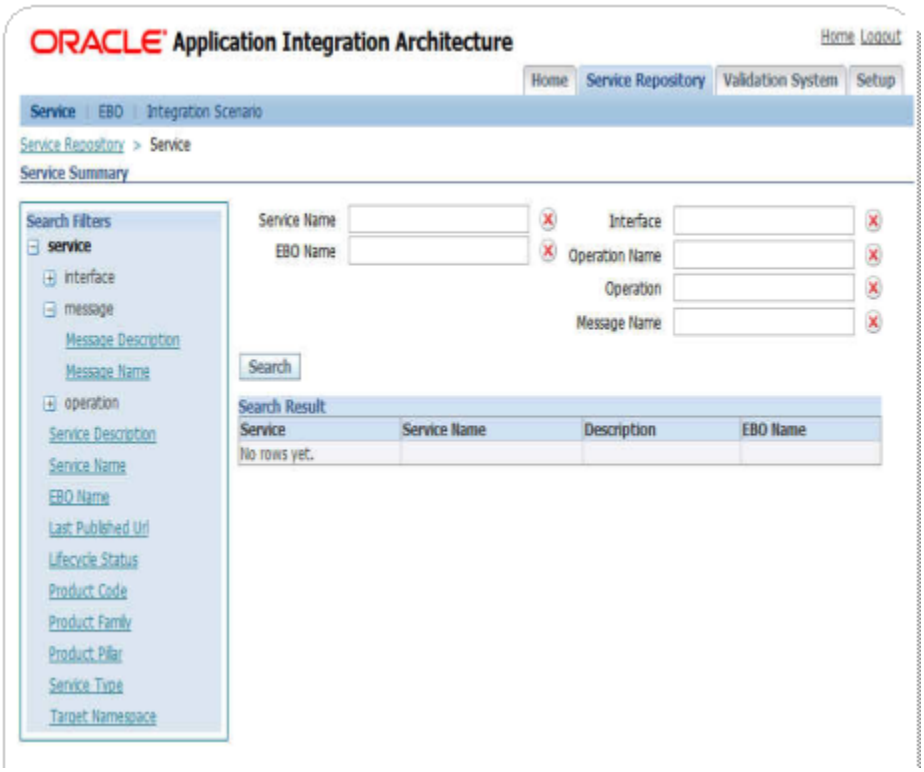
All first-class SOA assets in BSR are searchable. While simple, default search filters are provided on the assets, users may choose additional search filters to fine-tune their search results. The search filters apply to different types of SOA assets as well as their respective internal structure levels. Search filters can be derived from the AIA pre-defined annotations in source artifacts or base on the user-defined flex-fields.

No special user setup is needed.

The default search filters are based on the pre-defined annotations in the source codes (WSDLs and XSDs). These filters and annotations are more relevant to developers. Customer-defined search filters are based on flex-fields. These filters are more relevant to functional end-users

Search Filters

Using the BSR search functionality, all first-class SOA assets are accessible. There are default search filters provided as shown in the tree of the BSR screen shots. These default search filters derive from the annotations in the source WSDLs, XSDs, and integrationScenarios. Further, as you look at these artifacts, they are of different structural levels, for example, abstract WSDL/services contain root/service, message, interface/portType, and operation levels. There are pre-defined annotations on each of these levels. As a result, the tree displays different search filters pertaining to the respective levels. Finally, search filters, whether based on default annotations or user-defined flex-fields, are translatable.



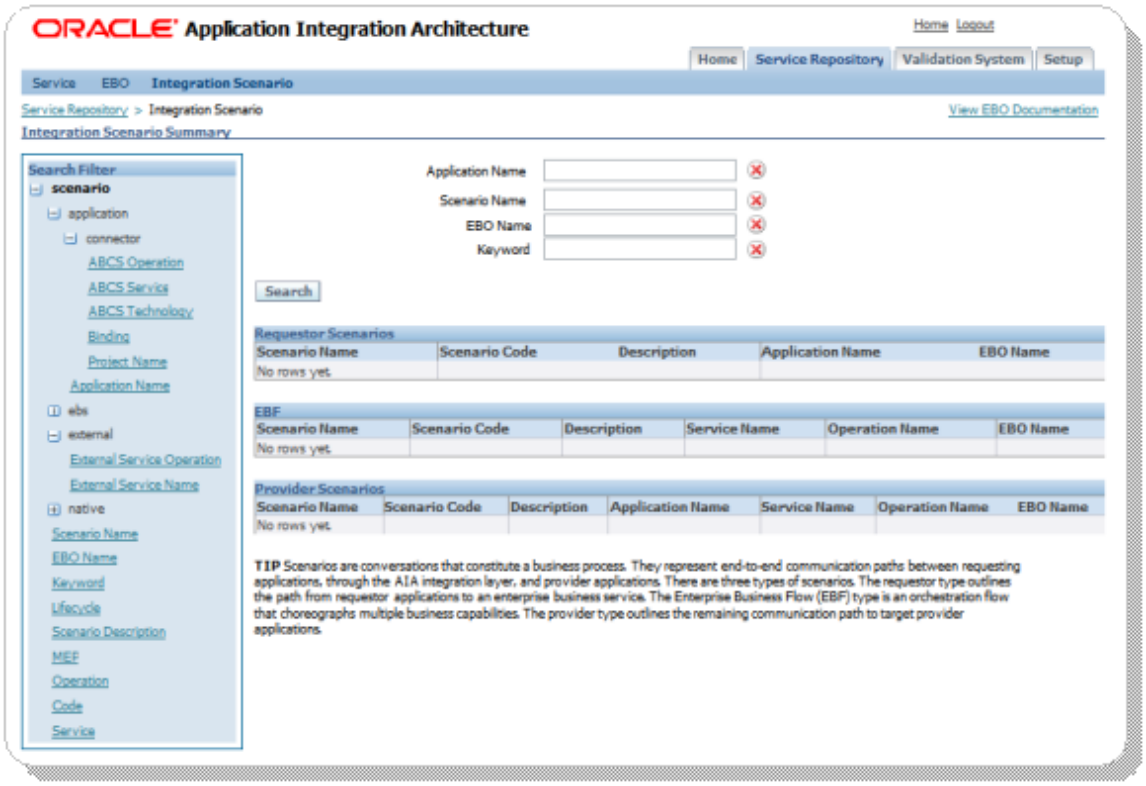
BSR page showing search filters

Assets are searchable at multiple levels. For example, services are searchable at the:

- Service level
- Interface level
- Operation level
- Message level

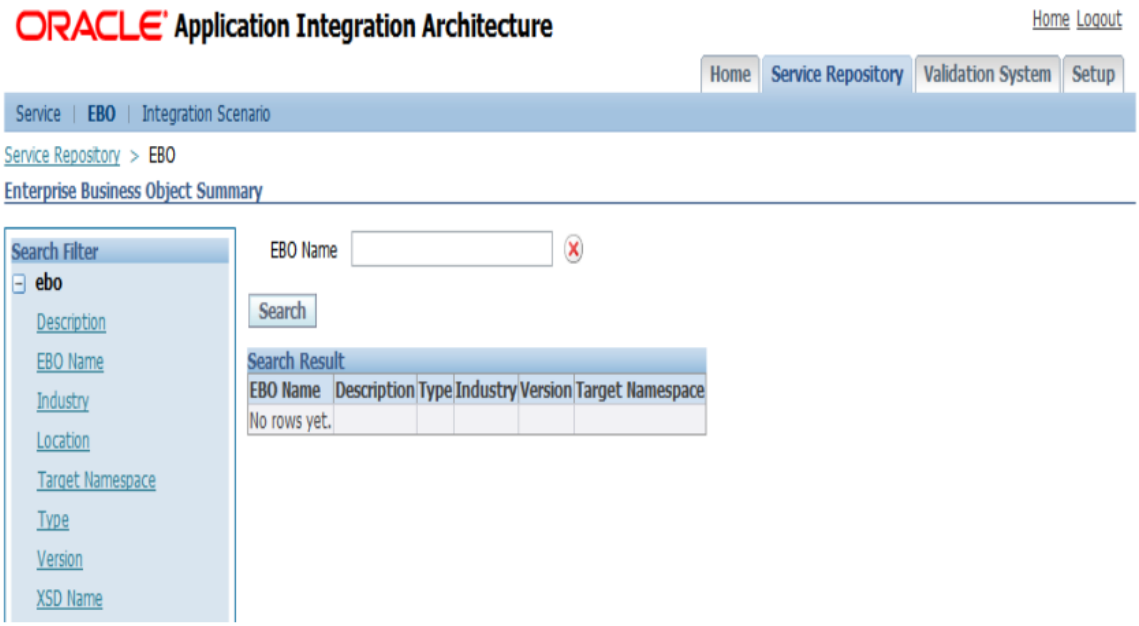
The integration scenario search searches files on multiple levels, including scenarios, connectors, EBS, external services, and native services.

This screen illustrates the integration scenario search:



BSR integration scenario search

The EBO/EBM search filters only on the top-most level:



BSR EBO search

Service-based Functional Impact Analysis

Users may assess the potential impact of service interface changes to AIA integration layer through the BSR UI. In the context of integration scenarios, multiple services and objects interact with one another. They collectively support a given process logic and constitute a multi-layered, multi-hop invocation relationship at runtime. As such, this relationship forms the basis for BSR's impact analysis feature centering around service interface changes.

You can perform the functional impact analysis with respect to AIA-style process integration. It is not based on physical artifact dependencies (e.g., import/include relationship among WSDLs and XSDs source artifacts or BPEL partnerLinks) with visibility to:

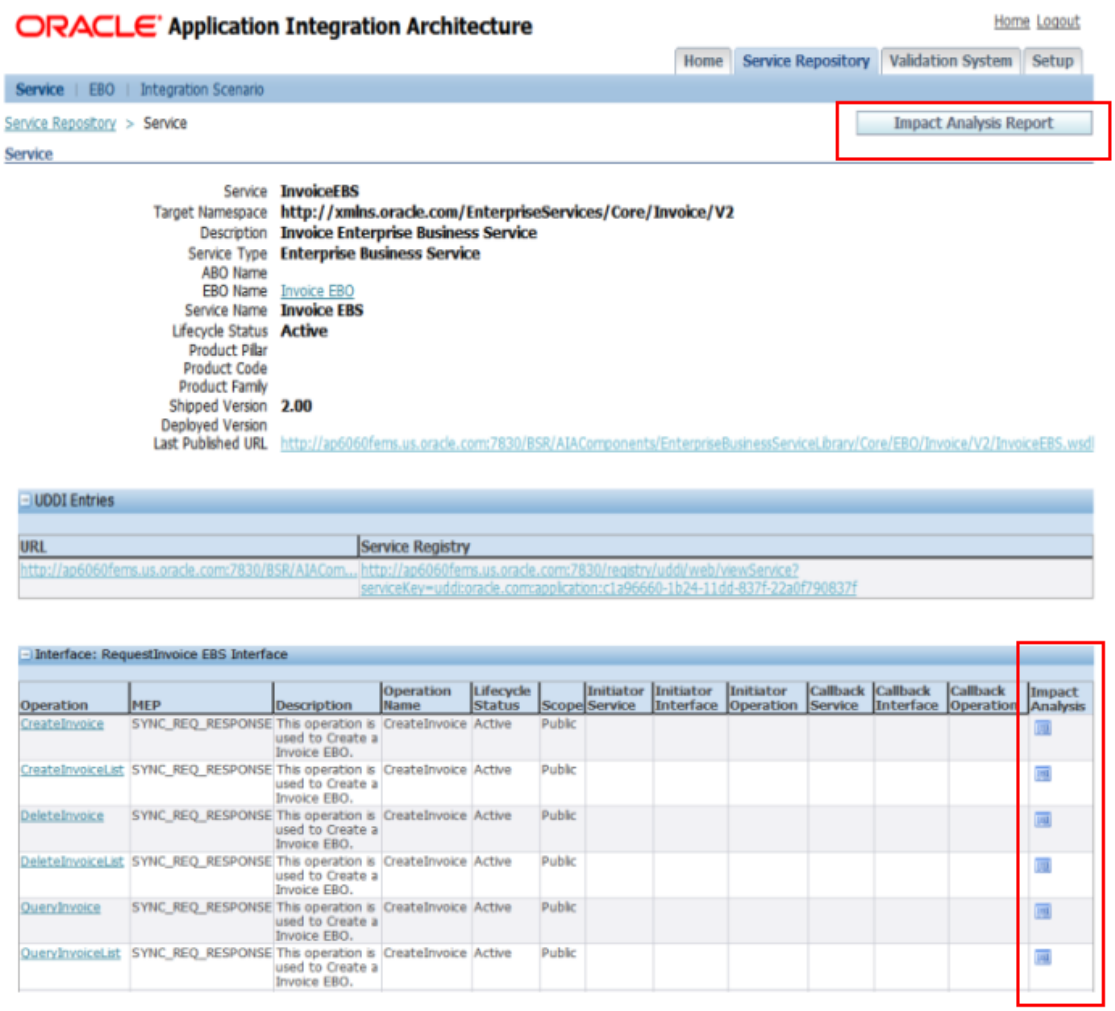
- Multi-hops: Impacts beyond your immediate neighboring services
- Multi-layers: Impacts to services, integration scenarios, and PIPs

Impact Analysis includes:

- Multiple types
  - Service based: What if I changed a service interface, what would be the impact to the AIA ecosystems, including other services, integration scenarios, and PIPs?
  - Service and operation based: If I change an operation on a given service, what are the impacts to the AIA ecosystems?
- Impact analysis reports
  - UI based reports
- Complements to other impact analyses

  What if objects were changed?

  What is the impact on internal implementation such as transformation?



Service page illustrating Impact Analysis buttons

Flex-Fields

It is often the case that functional and operational users would like to define additional attributes and descriptions on SOA assets to aid their use and management. In the AIA Foundation Pack 2.1, BSR allows end users to declare flex-fields and associate them to different types of SOA assets. The user-defined flex-fields are translatable and can serve as additional search filters during content access. Flex-fields complement the pre-defined AIA annotations because the annotations are embedded directly in the source artifacts during development, whereas flex-fields are declared in non-intrusive manner through BSR UI post deployment.

Flex-fields offer the following features:

- Multiple types

  Plain text or single-select drop down

- Role-based actions

  Read-write or read-only

- Search filters

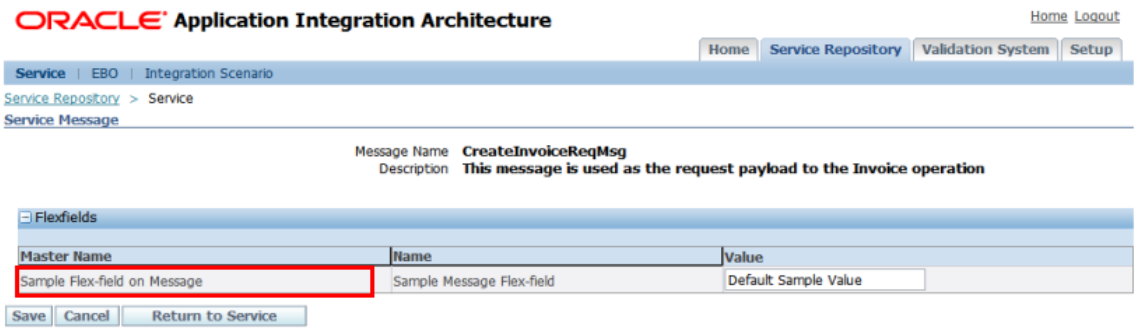
  Flex-fields are automatically made available as search filters on the asset types they are associated with.

- MLS ready

  Flex-fields are translatable, however the Master Name must always be in English. The search filters based on the flex-fields are also translatable.

Flex-fields can be associated to multiple levels for a given asset. For example, services can have flex-fields on the:

- Service level
- Interface level
- Operation level
- Message level



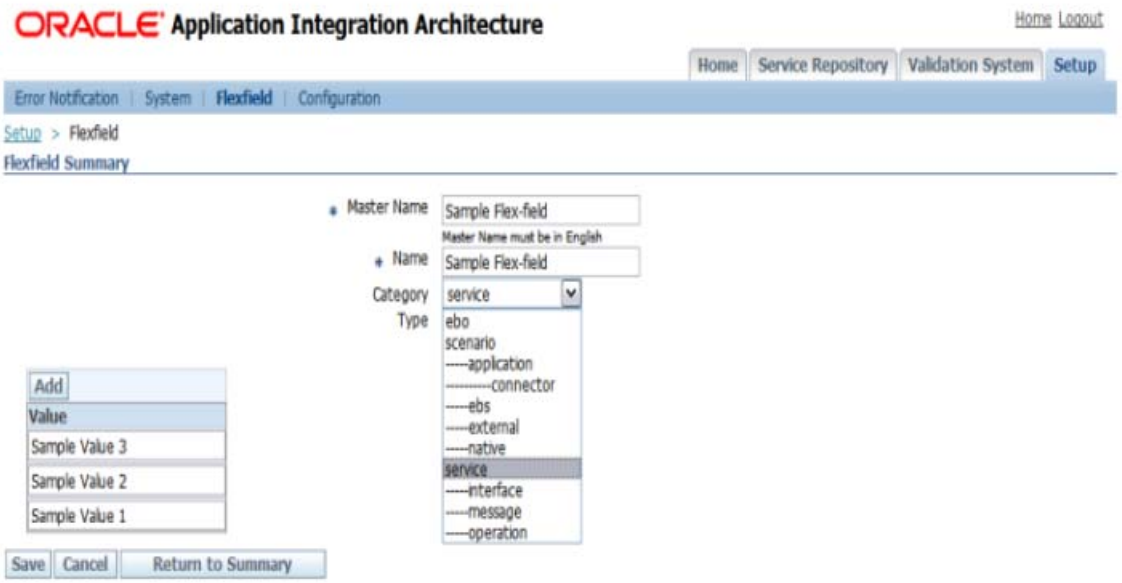
Example of flex-field on the message level

Using Flex-fields

To use flex-fields:

1. Declare

   Associate with a given category, including an asset type and level.

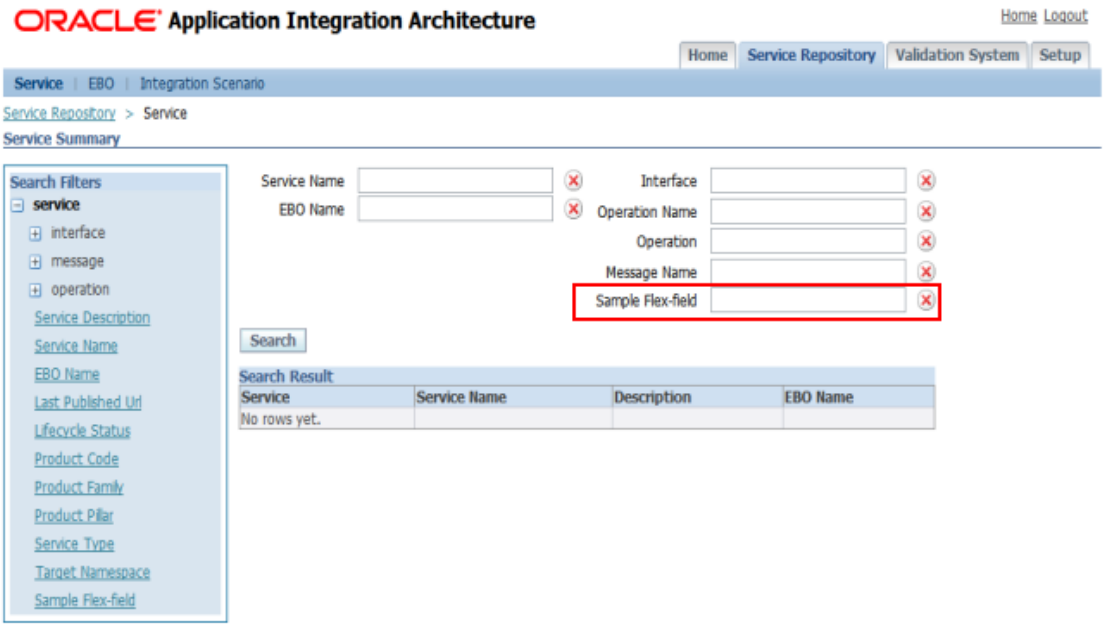


Flexfield Setup page

Flex-fields can associate with different asset types and on different levels as shown in the drop-down list.

Default values can be entered for both plain-text and LOV types of flex-fields

2. Enter or modify the value for a flex-field instance on a given SOA asset.
3. Search based on the flex-field instance value.



Example of a flex-field used as a search filter

Integration Scenario Enhancements

Integration scenarios represent cross-application message flows traversing through the AIA integration layer. They model the process topology based on AIA framework. There are three types of integration scenarios: requestor type, provider type, and enterprise business flow (EBF) type. Prior to Foundation Pack 2.1, only the requestor and provider types were captured in BSR. In this release, BSR is enhanced to capture the EBF type and to explicitly model complex orchestration of business capabilities.

Integration scenario enhancements include:

1. UI Rendering of EBF
    - End-to-end flows
        - Starting from: Requestor scenarios
        - Passing through: EBF scenarios
        - Connecting to: Provider scenarios
2. Linkages to individual assets
    - Hyperlink to applications
    - Hyperlink to EBO detail pages
        - EBO: 1st class SOA interface
    - Hyperlink to service detail pages
        - EBS: 1st class SOA interface
        - ABCS: 1st class SOA interface
        - External Services: not 1st class
        - Native Services: not 1st class
3. Removal of integration scenarios
    - Command-line script
        - Bulk mode
        - Single Mode

Composite Application Validation System

The Composite Application Validation System (CAVS) enables you to test integration web services without the presence of deployed participating applications and their web services. Using a framework that includes initiators to simulate calls to web services and simulators to simulate responses from web services, the CAVS provides a system that can test integrations without the need to set up deployments of all participating applications and web services that are involved in the integration.

At a high level, the CAVS:

- Provides a validation tool for business process integrations.

- Allows black box testing of individual services and integration flows.
- Enables the creation of response simulators, which facilitate the simulation of unavailable participating applications and services.
- Allows users to author and reuse test cases to perform functional testing throughout development, deployment, and production phases.

Oracle AIA Foundation Pack 2.1 Composite Application Validation System delivers the following enhancements:

- Improved routing to test simulators
- Routing setup and configuration UI
- Support for testing asynchronous delayed response interaction pattern
- Scheduling tests and generating reports
- User-friendly way to create request/response messages
- Data-driven testing
- Export and import of test definitions

Improved Routing to Test Simulators

When AIA artifacts route to a simulator, they can control the specific Simulator Definition ID to execute. Hence, routing to a simulator from an AIA service can be controlled at a fine grained level, that is, every invocation from the service can be routed to the simulator or not. This removes the complexity of Simulator Definition creation and improves test performance.

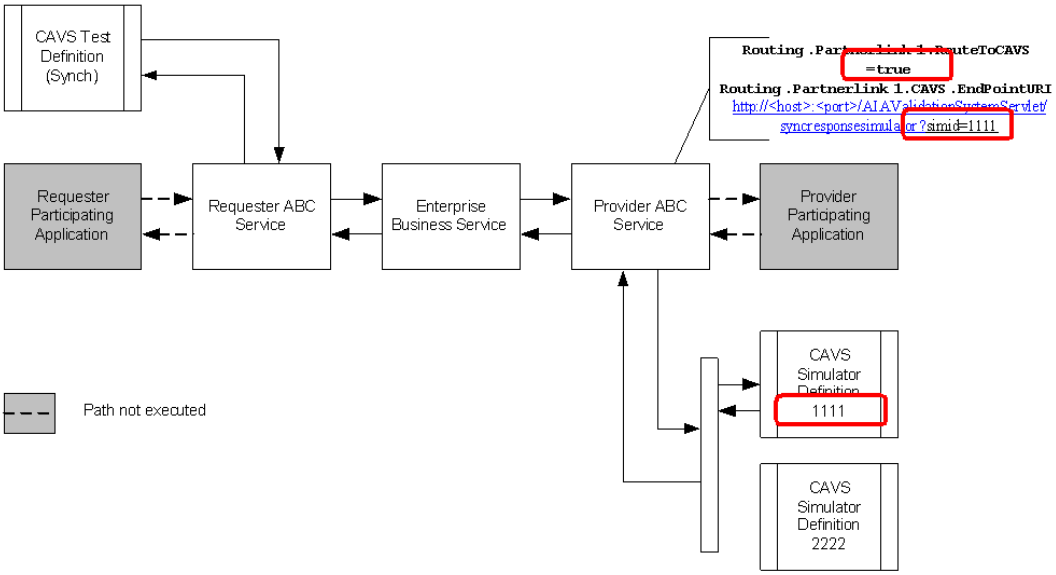
Setup

- The following values are set in the AIAConfigurationProperties.xml file:
  - *Routing.PartnerLinkName.[Operation].RouteToCAVS* when set to true will route to Simulator.

    Note: Operation tag is optional
  - *Routing.PartnerLinkName.[Operation].CAVS.EndpointURI* will hold the URL of the Simulator to be invoked(based on whether it is of type Synchronous, Notify or Async two way)
- Setting the Simulator Definition ID
  - Simulator ID is appended to the EndPointURI. For example,

    http://<host>:<port>/AIAValidationSystemServlet/syncresponsesimulator?simid=1111
- CAVS provides UI to set the above values

This diagram illustrates the routing to a specific simulator:



Routing to a specific simulator

The CAVS Test Definition takes care of the invocation of Requester ABCS, instead of the real requesting application.

On the provider-side, the ABCS will have a set of properties (including the RouteToCAVS and CAVS.EndPointURI properties) defined for it in the AIAConfigurationProperties.xml file available under AIA_HOME/config directory of the AIA installation. This can be modified using the UI to have RouteToCAVS as True. This will make the invocation reach the Simulator Definitions instead of the real service

Users can also use the UI to configure the CAVS.EndPointURI values. In the example, if 1111 is set in the properties, then the CAVS system will directly choose the 1111 Simulator Definition.

If the user chooses to not configure the value, then the system defaults to the existing behavior of searching all the simulators to find a match based on the key values selected in the Simulator Definition.

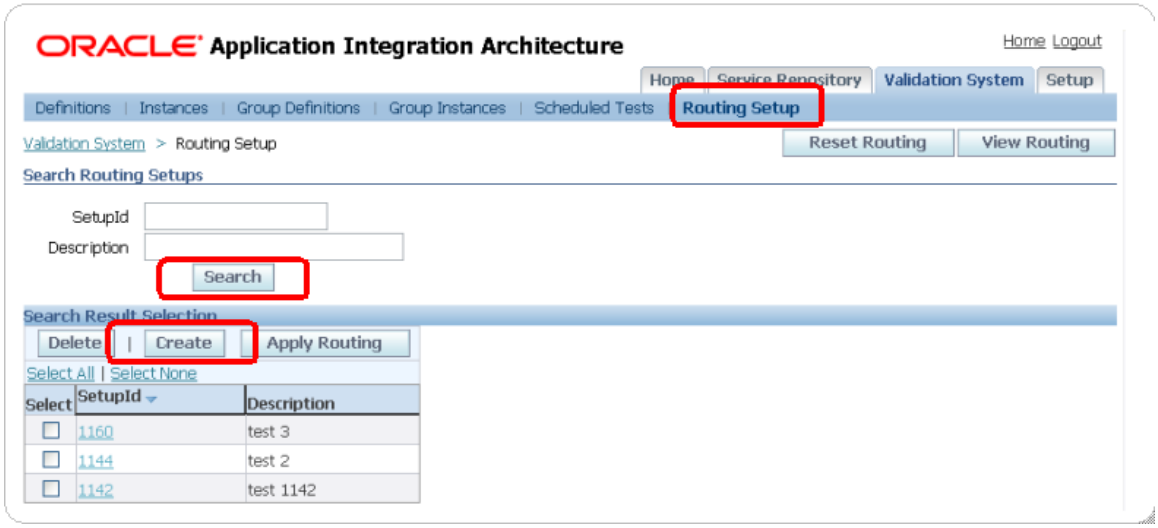
Routing Setup and Configuration UI

A user interface is provided to directly configure the Simulator Routing. This gives the ability to create several design-time instances of Routing Setups and the ability to activate Routing Setups whenever required. Pre-defined Routing Setups can also be associated with a Test Definition, which gets activated automatically during test execution. This replaces modifying the AIA Configuration Properties file manually, which is complex and error prone. The New UI masks all complexities and exposes only CAVS related properties. It also removes the need for manual intervention between test executions (test groups or scheduled tests) to change routing. Routings required for a Test Execution now become a part of the Test Definition. This is a one-time activity, which provides for improved automation.

Creating Routing Setup

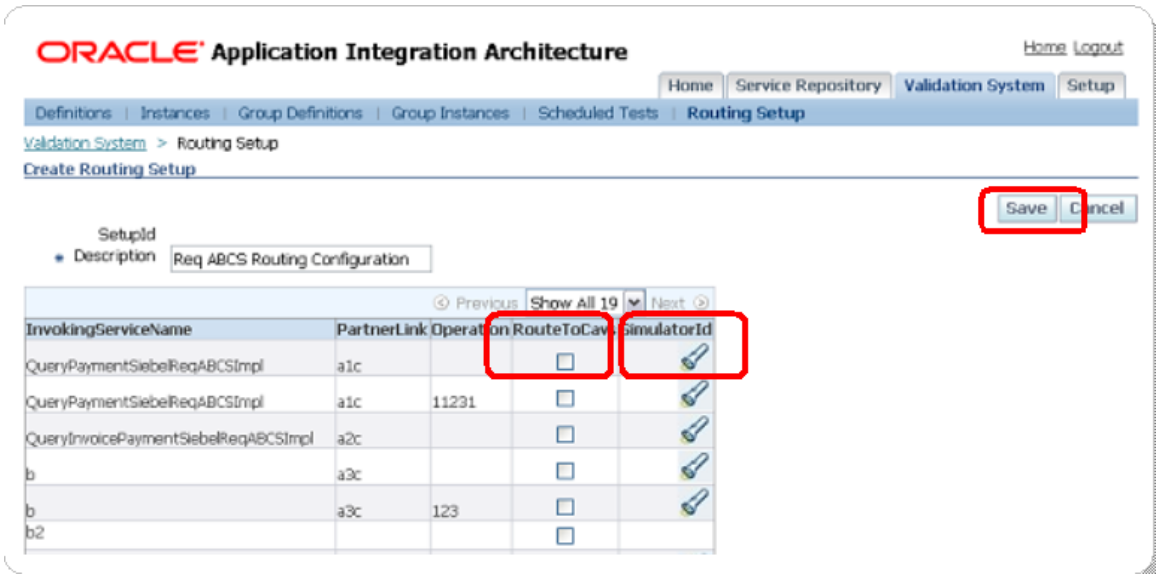
To create the routing setup:

1. On the Routing Setup page, click Create.



Routing Setup page

2. Check or clear Route to CAVS.
3. Select Simulator ID from a list of existing values.
4. Click Save to create a new Routing Setup ID.



Create Routing Setup page

Setups can be searched and modified at any time.

You can activate the required routing either manually or during test execution.

Activating the Routing Manually

At any time, choose the required routing and click Apply Routing, The current routing can be viewed using the View Routing button. All routings can be set to False using the Reset Routing button.

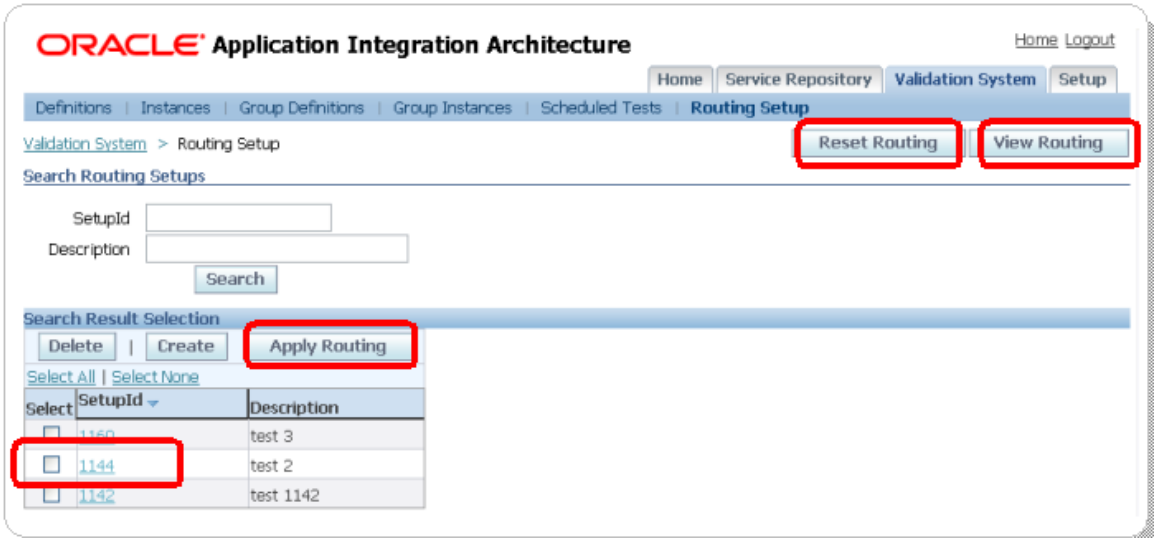


Illustration of manual routing

Activating During Test Execution

In the Test Definition, associate any existing Routing Setup ID. This will be applied automatically when executing that Test Definition. Any manually activated Routings will be overwritten.

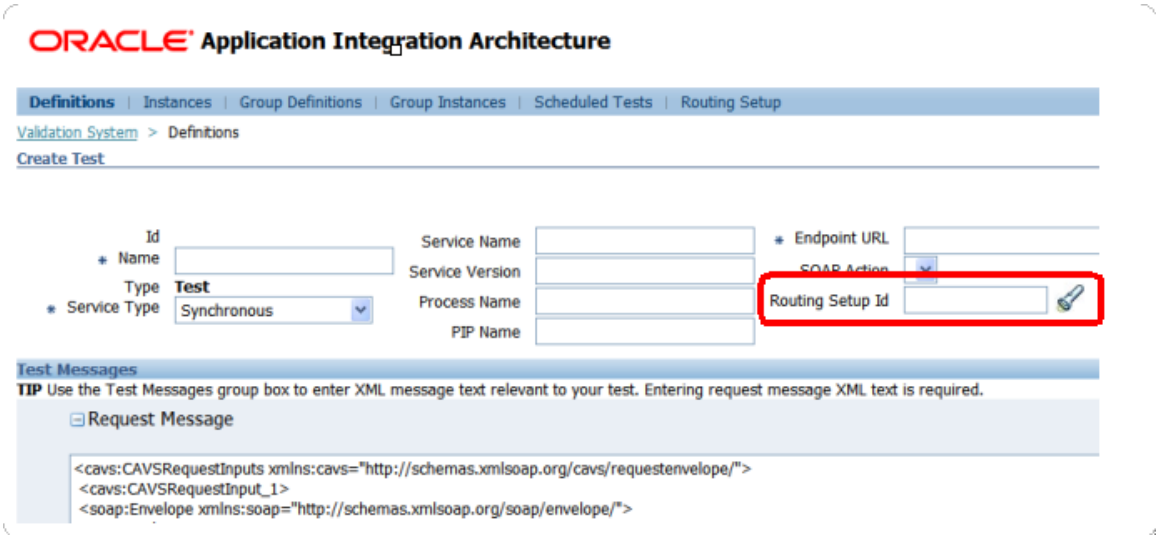
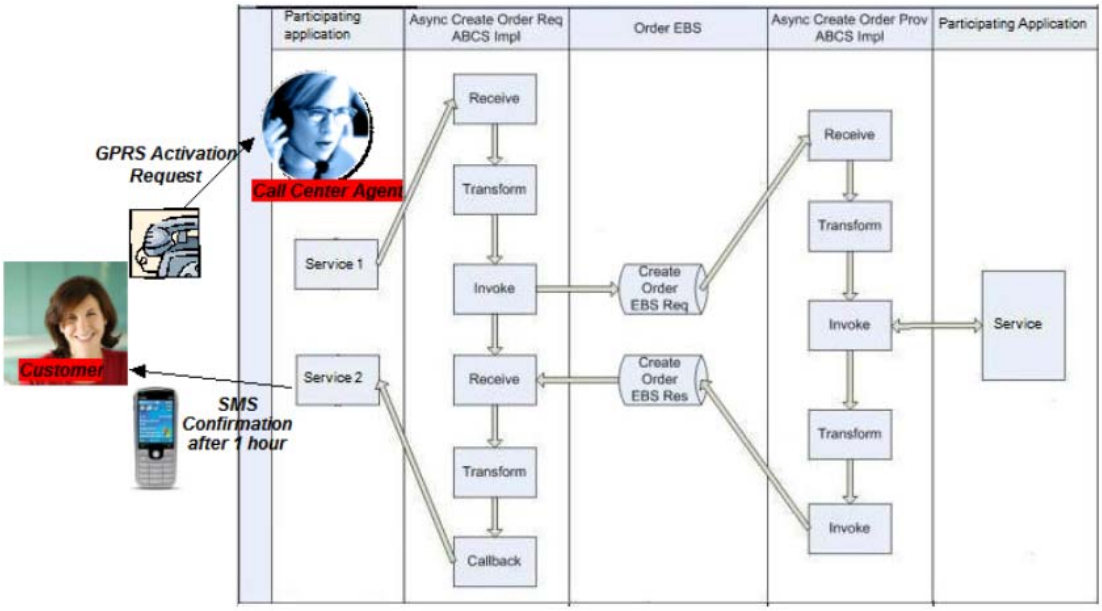


Illustration of routing activation during test execution

Support for Testing Asynchronous Delayed Response Interaction Pattern

Test Definitions utilize CAVS built-in correlation to identify and validate the delayed response against a predefined message. This now includes the ability to specify a timeout value after which the test can be deemed failed if it does not receive the response. Simulators mimic asynchronous services by allowing users to specify the URL to be optionally invoked for asynchronous delayed-response, after a specified time. Hence, it is possible to test AIA flows where the interaction pattern is asynchronous delayed response.

This diagram illustrates the asynchronous delayed-response use case:



Testing AIA flows with an asynchronous delayed-response pattern

The customer calls the call center and submits a request for GPRS activation. The customer does not wait on the call until the activation request is complete. Instead the customer hangs up the phone and the Call center agent is also ready to take another call. However Service 2 is invoked later to send back an SMS confirmation.

To test such a flow, you would want to verify the Requester ABCS or the provider ABCS in isolation or together.

In this flow, both the RequesterABCS and ProviderABCS behave in an asynchronous request-response manner because in both cases the invoking service doesn't expect a response and the ABCS' actually call back service/operation different from the invoking service. These invocations need to be correlated.

The difference between the Requester and Provider ABCS behavior is that ProviderABCS provides an inbuilt correlation but from a RequesterABCS perspective there is no inbuilt correlation because there are two different services out of the control of AIA.

CAVS would need to address both cases. It is important that CAVS provide a way to correlate messages in and out of the RequesterABCS via the Test Definitions, so that the CAVS user could send a request from Service1 and compare the actual payload received at Service2 versus the expected payload although Service 1 and Service 2 are technically unrelated.

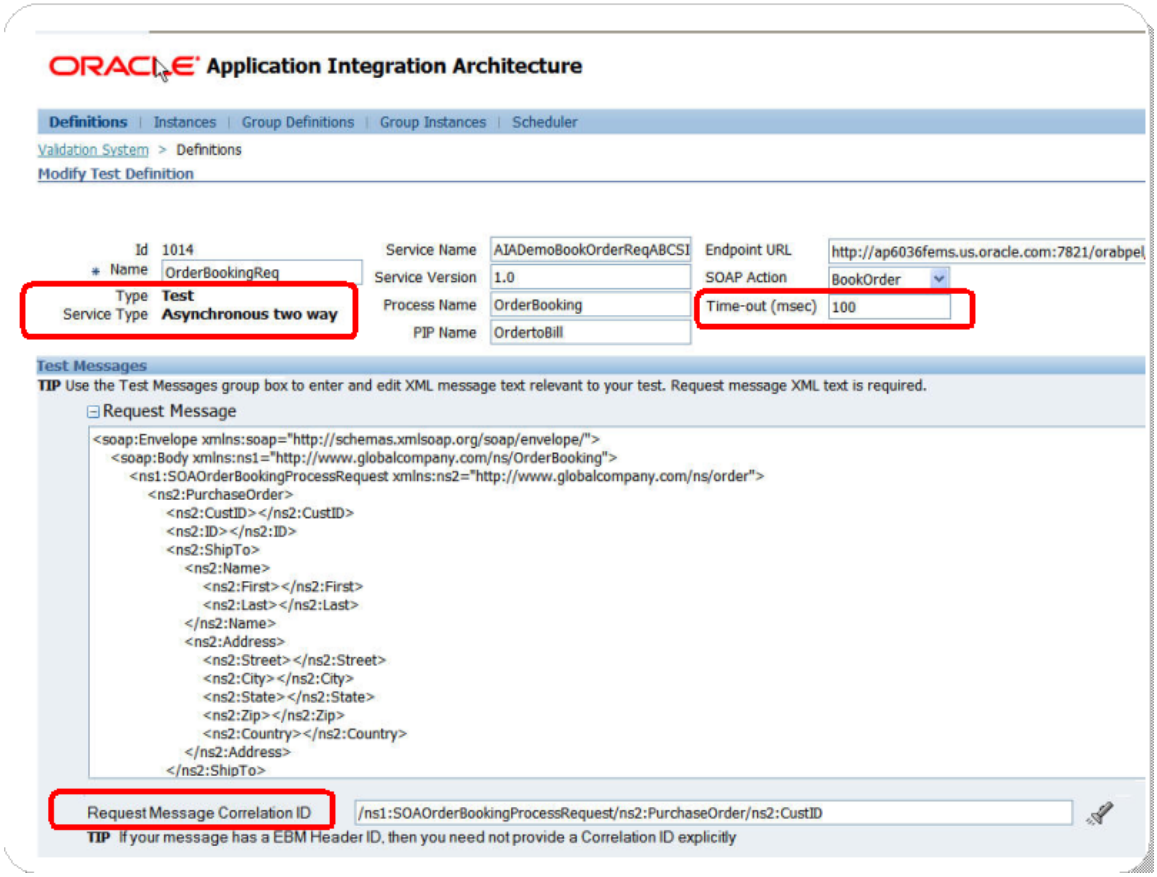
What is also required is the ability to create Simulators of type "asynchronous two-way" that can simulate the Provider ABCS or the provider application.

The setup involves:

- Creating the test definition
- Creating the simulator definition

Creating the Test Definition

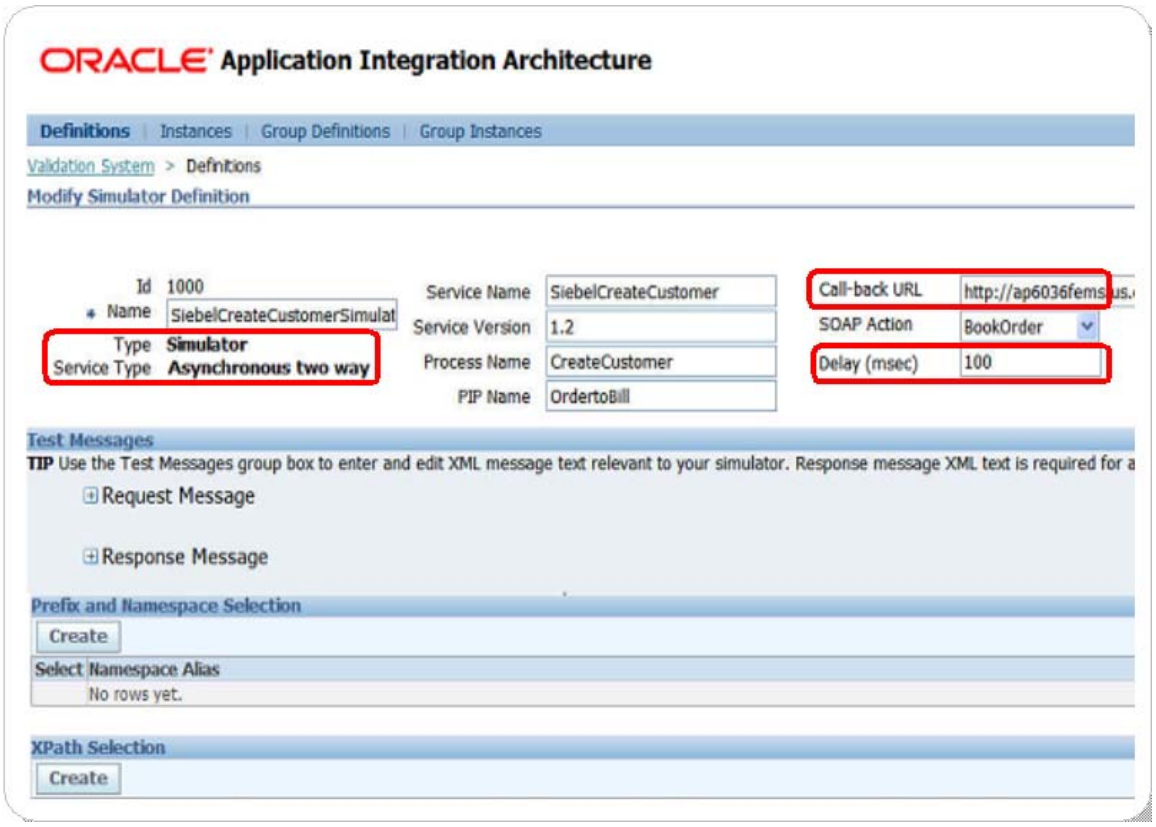
1. Create a Test Definition of type Asynchronous two way
2. Provide the request and response message
3. Choose Correlation ID
4. Set the Time-out value.
   a. Set if you want to validate against execution time
   b. Otherwise set value to -1



Test Definition page

Creating the Simulator Definition

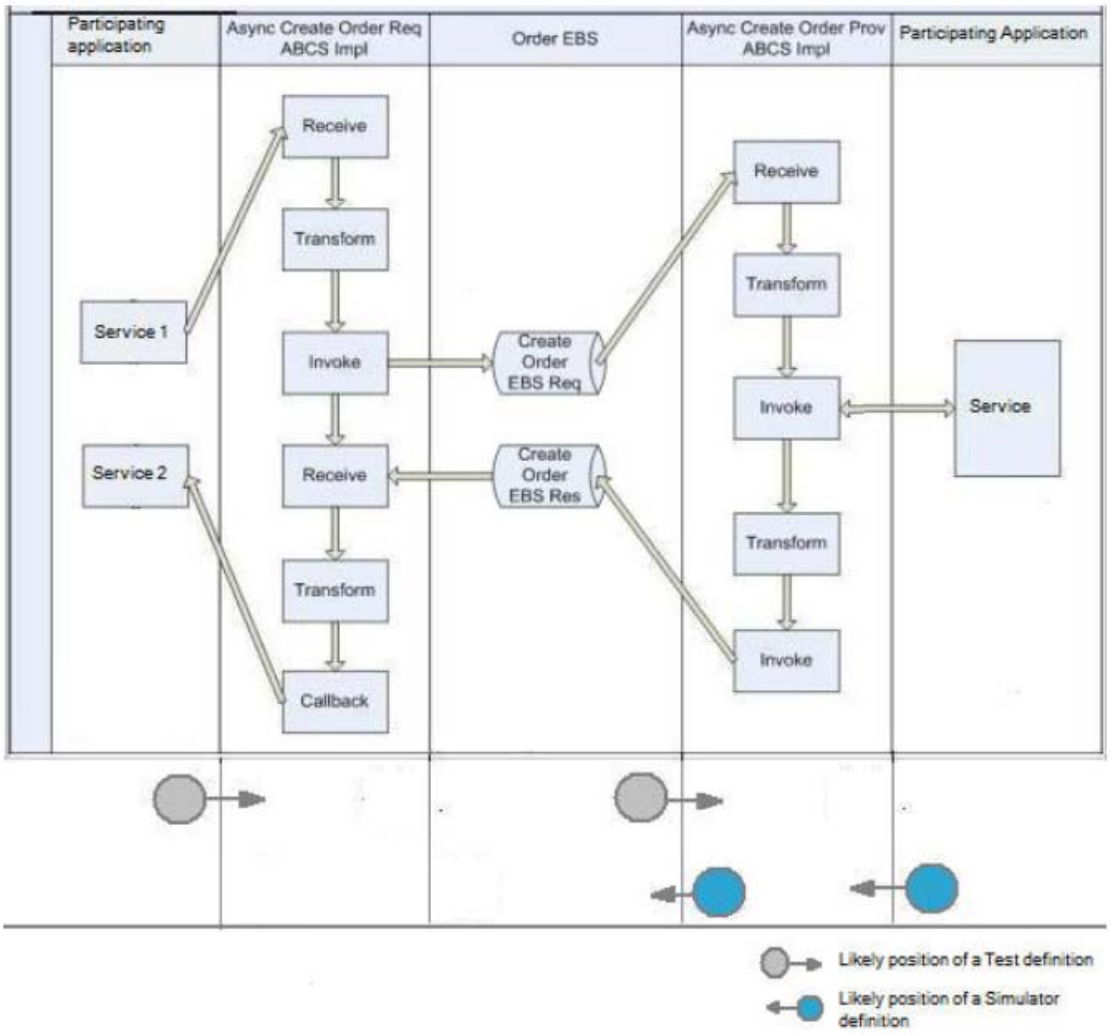
1. Create a Simulator Definition of type Asynchronous two way
2. Provide the request and response message
3. Provide the URL where the response invocation must occur
4. Optionally, set a delay after which the response should be sent



Simulator Definition page

Testing the Asynchronous Service

The diagram shows the possible points where the Test Definition and Simulator Definition can be placed. Based on the decision, the routing configurations are first updated. On execution of the Async two-way Test Definition, a Test instance is created, which moves to 'Delayed' status. A Simulator of Type 'Async two way' or a real service would respond after a certain time and the response is compared with the actual response in the Test Definition that is in Delayed status. Based on the result, success or failure will be reported



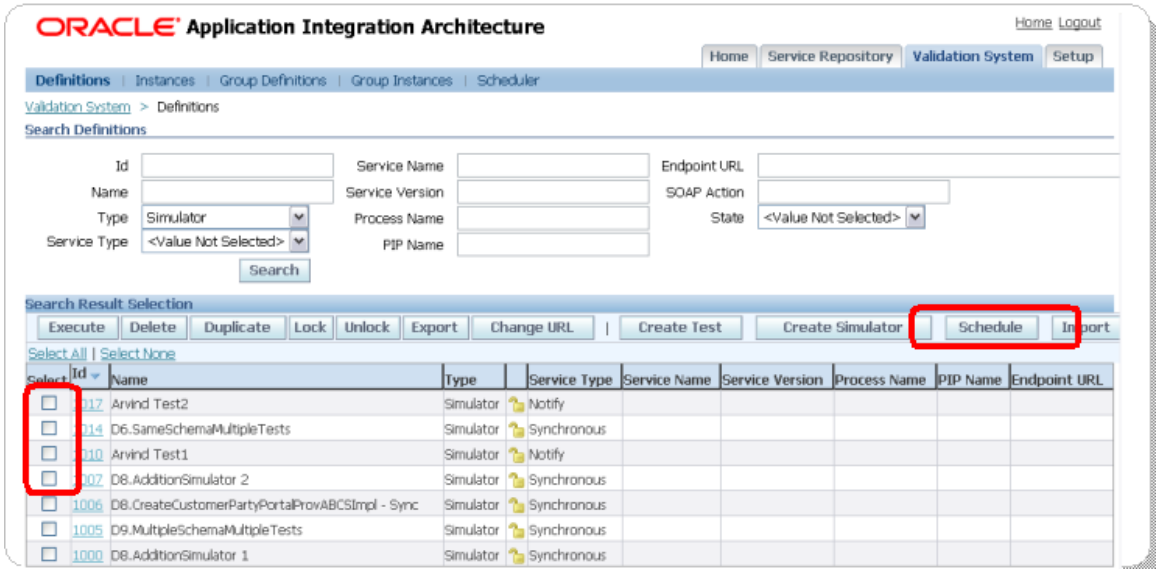
Testing the asynchronous service

Scheduling Tests and Generating Reports

Users can schedule executions of Test Definitions or Test Group Definitions at pre-defined time intervals. They also have the ability to view/search scheduled tests and view their execution statuses and results and to generate reports when test groups are executed using the scheduler feature. This helps execute regression testing at pre-defined intervals and test build validity in development cycles.

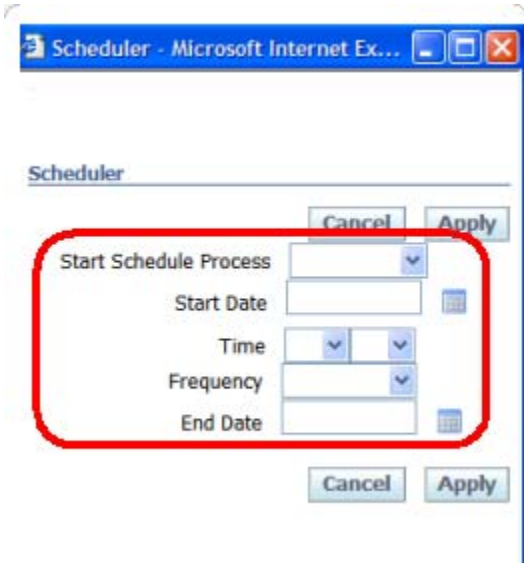
Scheduling Tests

Select the Test definitions or Test group definitions that you want to schedule and click the Schedule button.



Definitions page

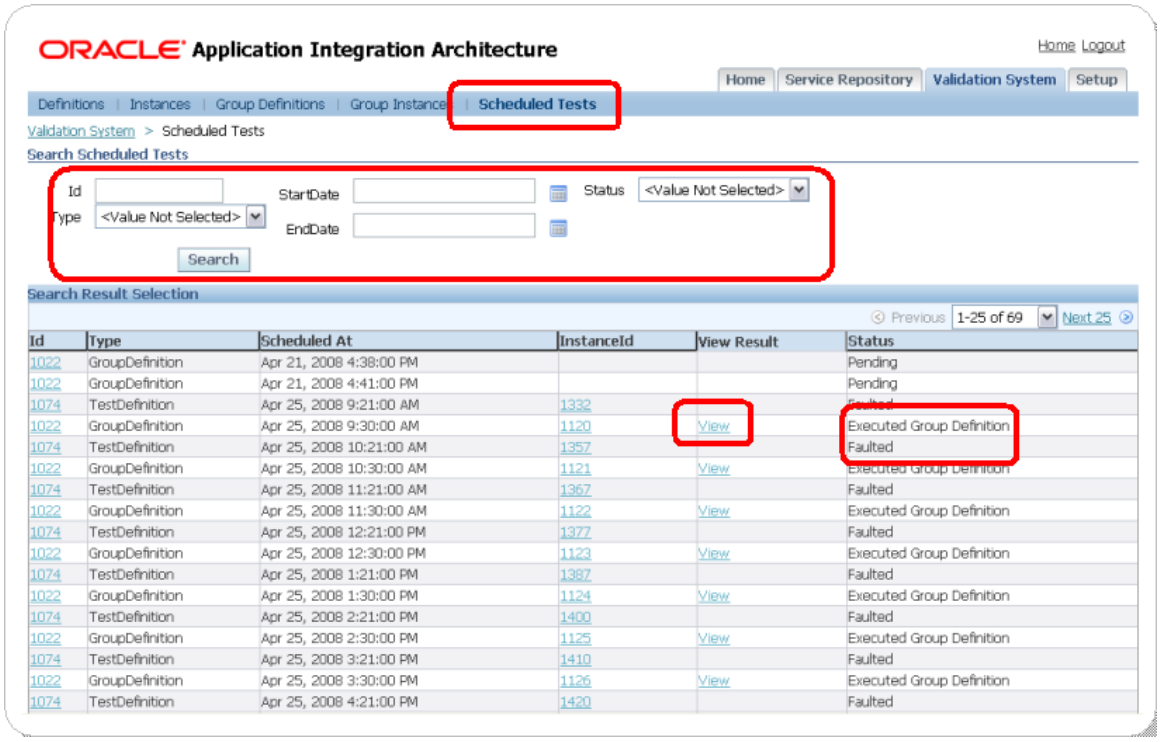
A pop-up appears where you can provide the time interval. Choose to start scheduling immediately or later in the ‘Start Schedule Process’ dropdown. If “Choose to start later” is chosen, the start data and time must be entered. You can choose the frequency to be ‘only once’ or ‘daily’, ‘weekly’ etc. If you do not choose ‘only once’, provide an end date.



Scheduler pop-up

Viewing Scheduled Tests

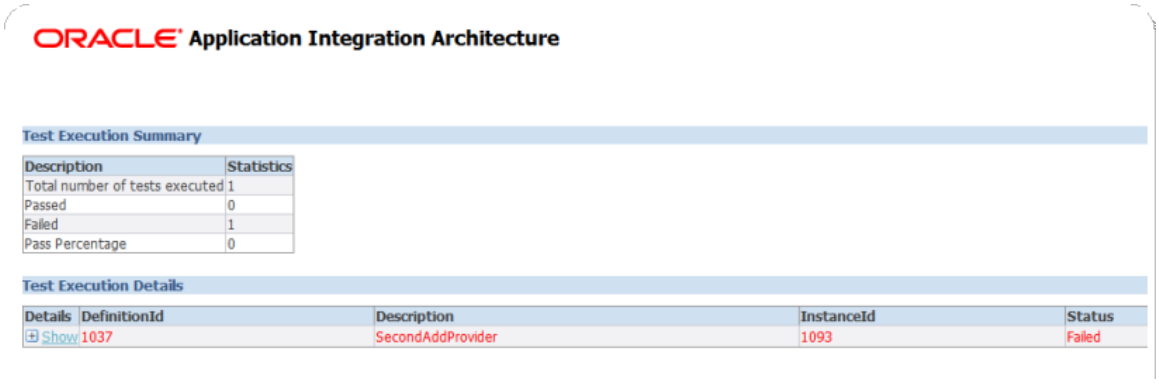
Click on Scheduled Tests tab. You can search already scheduled tests based on several criteria. The Status of each test execution is displayed; “Pending” indicates the test is waiting to be executed. The other statuses indicate the success or failure of the test execution



Scheduled Tests page

Viewing Reports or Test Results

The system provides summary statistics of a test group execution. This applies only to test groups and not to test instances. Users can drill down on each test within the group to understand the reasons behind the failure.



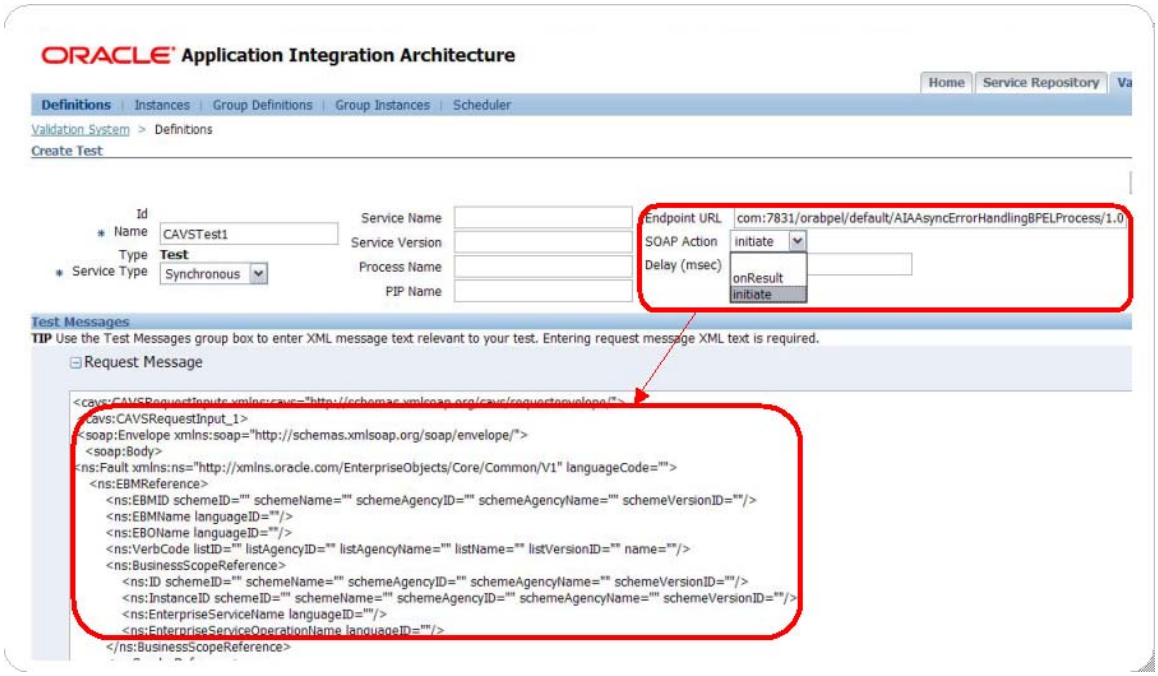
Test Results page

User-Friendly Way to Create Request/Response Messages

It is now possible to auto-generate request and response stubs based on the Endpoint URL and the SOAP Action selected. This includes the ability to directly map fields from request message to the response message in case of Simulators and the ability to use SYSDATE instead of hard-coding date values (extensible later to include more constants) in case of Simulators. This removes the requirement for users to write complex SOAP messages.

Auto Generation of Message Skeleton

This screen illustrates the auto generation of the message skeleton:



Definitions page

Auto generation applies only to BPEL processes running on the same server as CAVS.

If CAVS is unable to auto-generate the complete message skeleton, it will provide the user with the required SOAP envelope structure and then instruct the user to copy/paste their payload at a specific location within the structure.

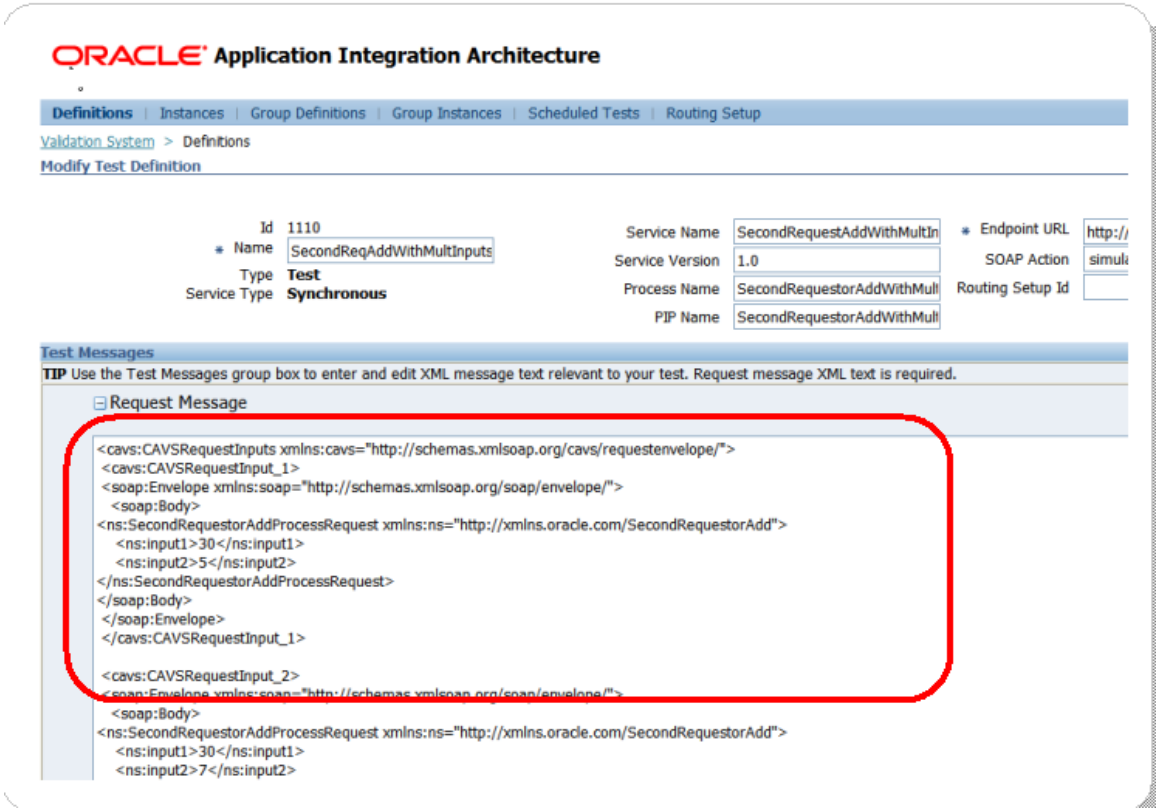
Data-Driven Testing

This enables the creation of Test and Simulator Definitions that can hold multiple data sets for a service. On execution of the Test Definition all the data sets execute sequentially. When calling a Simulator Definition that holds multiple datasets, CAVS can select the required data set. This includes the ability to create a Simulator Definition that can hold all simulations required by a single chatty conversation service. It also gives you the ability to see all correlated simulations in one Simulator definition rather than being distributed across Simulator Definitions. Test Definitions and Simulator Definitions are reusable because the cross-references created by the test case executions are automatically rolled back.

Creating Multiple Sets of Data

You can include multiple sets of data by copy/pasting the envelope indicated by `<cavs:CAVSRequestInput_xx/>`

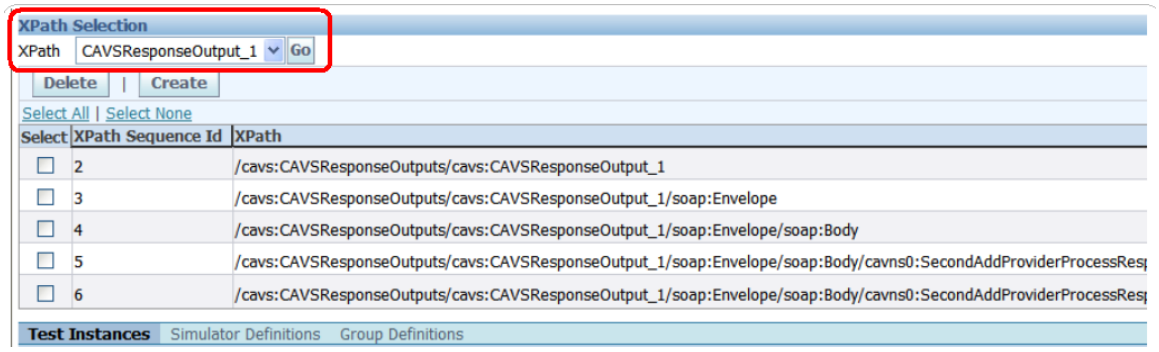
For every set in the request there must be corresponding set in the response `<cavs:CAVSResponseOutput_xx>`



Creating multiple sets of data

Working with XPATH for Multiple Sets

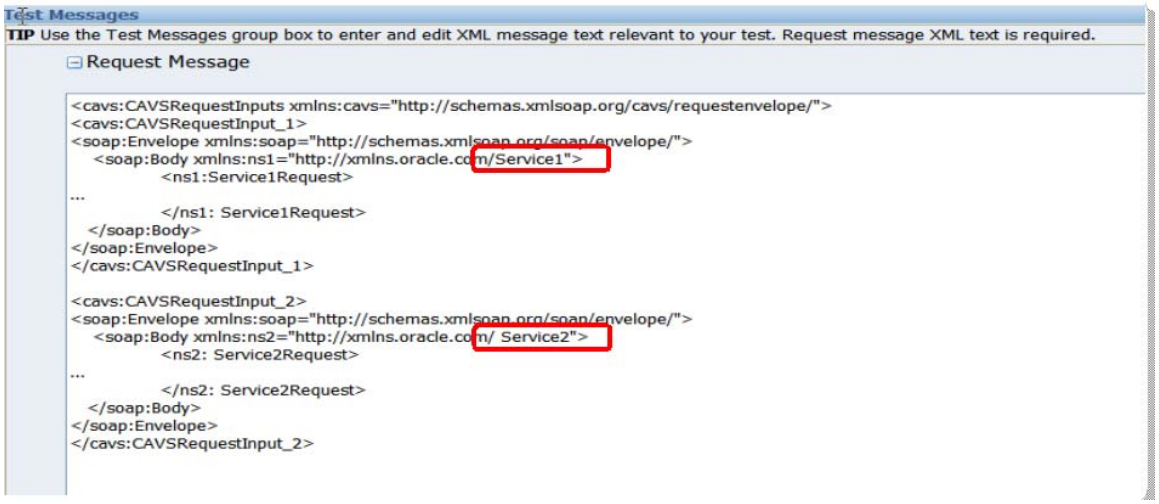
Generate XPATH will generate XPATH for all sets of data. Use the XPATH selection drop-down to select the specific data of interest.



Selecting specific data

Working with Chatty Conversation Simulators

A simulator can hold sets of data belonging to different services. In a chatty conversation scenario, the user configures the simulator routing to always point to one Simulator Definition, which in-turn simulates all services participating in the chatty conversation



Executing Data Driven Definitions

All sets of data in a Test Definition are executed sequentially creating one test instance for each data set. Each request set has a response set that is used for comparing the results. When calling a simulator that has multiple sets of data, one of the sets get executed based on the matching with incoming request. If the Simulator is to send a response, then the appropriate response set will be sent back as the response to the calling service.

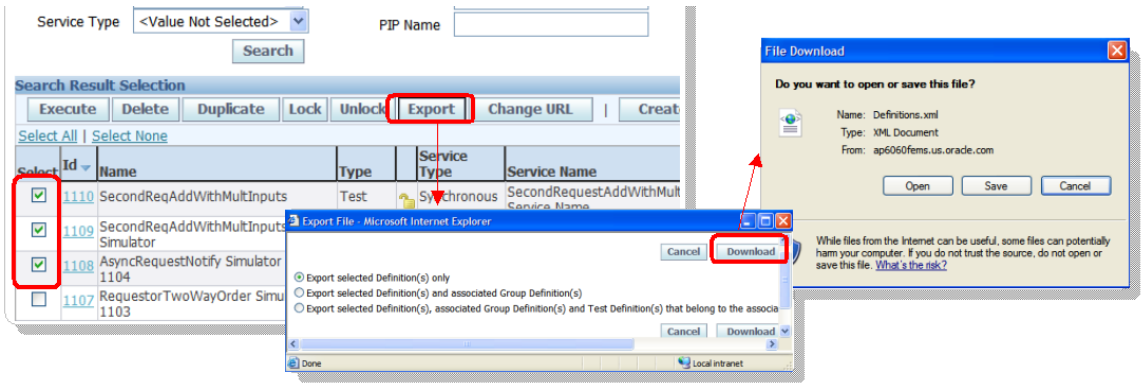
Export and Import of Test Definitions

It is now possible to migrate test/simulator definitions and test group definitions across various instances of AIA. Export allows you to export Test Definitions, Simulator Definitions, and associated Group Definitions all at once into one xml file. This file is stored in the local system and can be imported later. The import will re-generate the ID fields based on the target instance's sequences. This allows you to move definitions from one instance of AIA to another and helps in debugging because the same tests can be run on different instances. With every new development build, the existing build may be refreshed, but the definitions can be exported and reused, thereby, saving a lot of effort. This feature does not aim to operate between different versions of AIA. Instead the focus is on migrating within different instances of the same version of AIA.

Exporting Test and Test Group Definitions

Choose the Test/Simulator Definitions and click Export. The Export options pop-up appears. Users can choose to also export associated Test groups (and optionally Test Definitions included in such groups but not selected explicitly). There is no separate Export button in Test group definition page. Data will be exported in a standard XML format adhering to a standard template. This xml file will be stored in the local OS.

This diagram illustrates the export process:

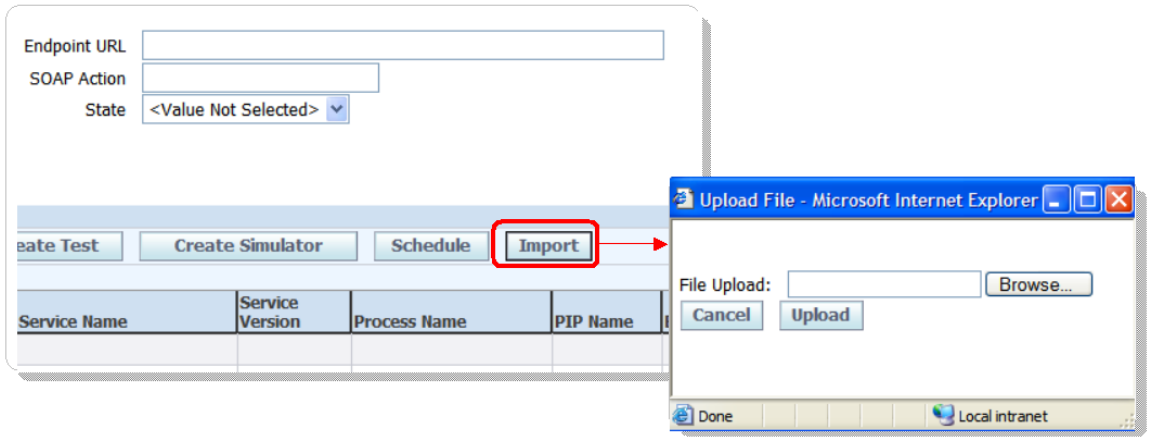


Exporting test and test group definitions

Importing Test and Test Group Definitions

Click Import to import the Definitions file that you earlier exported to the Local OS. Before importing you can also add more data to the xml file, as long as it adheres to the schema (template). During the import, the keys (like Test ID) change based on the target environment's sequence IDs.

This diagram illustrates the import process:



Importing test and test group definitions

Enterprise Object Library: Enterprise Business Services, Enterprise Business Objects and Enterprise Business Messages

An Enterprise Business Object (EBO) is a standard business data object definition used in the canonical data model. It contains components that satisfy the requirements of business objects from best of breed application data models and is application independent. EBOs are used by AIA to enable integration of many applications rather than point-to-point integrations, which become inefficient if there are more than two participating applications.

An Enterprise business message (EBM) acts as the container for the EBOs for an enterprise business service operation.

An Enterprise business service (EBS) provides the web service implementation of the EBO. An EBS is application-independent and performs the functions required by the generic business service. The primary purposes of an EBS are as follows:

- To receive the request from the calling application.
- To identify the implementation as well as the deployment that is responsible for providing the requested service.
- To delegate the requested task to the right implementation.
- To receive the response and return the enterprise business message (EBM) to the calling application.

Enterprise Business Services, Objects and Messages

Oracle AIA Foundation Pack 2.1 Enterprise Object Library (EOL) delivers the following new or updated Enterprise Business Services, objects and messages:

EOL Object	Details
Bill of Lading (New)	A bill of lading (also referred to as a BOL or B/L) is a document issued by a carrier, for example, a ship's master or by a company's shipping department, acknowledging that specified goods have been received on board as cargo for conveyance to a named place for delivery to a designated consignee. In case of loss, damage, or delay, the bill of lading is the basis for filing freight claims.
Demand Forecast (New)	Statement estimating the expected demand for a particular good, component, or service from a customer, region or the entire market over a specified future period. For example, a monthly forecast of 10 units each month for an item from a customer.
Freight Terms (New)	An agreement indicating who (the buyer or the seller) pays the freight costs of an order and when they are to be paid, for example, at the destination or the shipping point.
Shipment Request (New)	A Shipment Request is an instruction, typically from an outsourcer to a logistics provider to pick, pack, and ship on behalf of the outsourcer. The Shipment Request contains all required information for the logistics provider to perform the outbound activities. Shipment Requests typically contain information such as customer, ship to address, carrier, ship date, expected receipt date, items, and quantity. It may also contain valued-added service instructions such as packing or labeling. If a customer has requested a specific grade of material or lot/serial number, this information would be communicated in the Shipment Request. Pricing and billing information may be excluded from the Shipment Request.
Inventory Reservation (New)	An allotment of supply, such as on hand inventory or purchase order to a demand such as sales order, manufacturing order.
Pick List (New)	Pick List is a document that lists material to be retrieved ("picked") from various locations in a warehouse in order to fill a production order or a sales order, or a shipping order.
Accounting Entry (New)	An accounting entry consists of multiple accounting lines that will be used by the GL application to generate journals that then can be posted to

	Ledger.
Credit Memo (New)	A Credit Memo (short for credit memorandum) is a commercial document issued by a seller to a buyer, indicating the products, quantities, and agreed prices for products or services that the seller provided the buyer with, but the buyer returned or did not receive.
Currency Exchange (New)	Defines currency conversion rates between any two currencies used to convert foreign currency amounts. For example, due to the weakening dollar, one US dollar = .73 Euros. 1 Euro = 1.63 Japanese Yen.
Debit Memo (New)	A Debit Memo (short for debit memorandum) is a statement that is generated when a Party fails to pay or short-pays an invoice (or the amount due is not fully invoiced), identifying the balance amount due and any late fees owed. In function, debit memos are identical to invoices.
Chart of Accounts (New)	A systematic listing of all accounts used by a company. The Chart of Accounts represents a single GAAP recorded within a shared Chart of Accounts Structure, the Chart of Accounts then controlling the list of values assigned to each segment of a Chart of Accounts Structure.
PaymentTerm (New)	Payment Term represents the arrangement between a debtor and creditor for when a creditor should be paid for goods or services supplied by the creditor to the debtor.
Payable Invoice (New)	Payable invoice is a commercial document issued by a supplier indicating the products, quantities, and agreed prices for products or services that the supplier has already provided to the buyer.
Promotion (New)	A rebate is a type of sales promotion marketers' use primarily as incentives or supplements to product sales. Rebates are offered either by the retailer or the manufacturer of the particular item.
Sales Quote (New)	A commercial statement detailing a set of products and services to be purchased in a single transaction by one party from another for a defined price
Bank Account (New)	A bank account is a financial account with a banking institution recording the financial transactions between the customer(s) and the bank and the resulting financial position of the customer with the bank. Bank accounts may have a positive or credit balance.
BusinessCalendar(New)	A business calendar allows you to specify which days of the week are business days for calculating average balances. This calendar ensures that journals created in General Ledger and Subledger Accounting are only posted on valid business days.
Fund Transfer (New)	Cash Transfers, including bank account transfers, between two internal bank accounts and adhoc payments that transfer fund from internal bank account to external bank account in ad-hoc bases. For example, transfer from bank account 12345 to 1234.
Fund Transfer Schedule (New)	Used for recurring transfers. The fund transfer schedule lists the rules for sending these recurring transfers. For example, you might set up a transfer to another account for $100.
Shipment Advice (New)	Shipment Advice is a document sent from the Logistics Service Provider system advising the status of a shipment sent to the Customer of a

	Logistics outsourcer.
Check (New)	A negotiable instrument instructing a financial institution to pay a specific amount of a specific currency from a specific demand account held in the payers name with that institution.
Customer Party (Update)	An individual or organization to which the Enterprise intends to sell products or services. For example, ABC Computer and John Smith are customer parties of the Enterprise Dell.
Supplier Party (Update)	An individual or company from which the deploying company intends to procure products or services. For example, ABC Computer and John Smith are suppliers of the deploying company.
Advance Shipment Notice (Update)	A notice (ASN) sent by the vendor to the customer indicating what merchandise has been shipped. It enables the receiver to identify a package's contents electronically without having to open it. For example, when the delivery gets shipped, an ASN will be sent to the customer, describing the delivery with its containers and lines.