Oracle® Application Integration Architecture - Foundation Pack 2.5: Core Infrastructure Components Guide

Release 2.5

Part No. E15764-01

October 2009

Oracle Application Integration Architecture - Foundation Pack 2.5: Core Infrastructure Components Guide

Part No. E15764-01

Copyright © 2007, 2009, 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

Copyright © 2009, Oracle. All rights reserved.

Copyright © 2009, Oracle. All rights reserved.

Preface

The *Oracle Application Integration Architecture – Foundation Pack: Core Infrastructure Components Guide* provides information about how to:

- Work with the Business Service Repository (BSR).
- Work with the Composite Application Validation System (CAVS).
- Set up and use error handling and logging.
- Work with the diagnostics framework.
- Work with Oracle AIA developer tools.

Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide

See the *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide* for information about how to:

- Create an integration scenario.
- Define business service patterns.
- Design and develop EBSs.
- Design and develop enterprise business flows (EBFs).
- Design and construct ABC services.
- Work with message transformation, enrichment, and configuration.
- Develop custom XPath functions.
- Design and construct JMS Adapter services.
- Work with EBM headers.
- Work with message routing.
- Work with transactions.
- Develop Oracle AIA services to work with the CAVS.
- Configure Oracle AIA processes to be eligible for error handling and logging.
- Extend EBOs.
- Work with the Event Aggregation programming model.
- Work with the Publish-and-Subscribe programming model.

In addition, this guide provides Oracle AIA naming standards.

The *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide* is a companion volume to the following guides and resources discussed in this preface:

- The *Oracle Application Integration Architecture – Foundation Pack: Concepts and Technologies Guide*
- The *Oracle Application Integration Architecture – Foundation Pack: Core Infrastructure Components Guide*
- Process integration pack (PIP) implementation guides
- Additional resources

Oracle Application Integration Architecture – Foundation Pack: Concepts and Technologies Guide

See the *Oracle Application Integration Architecture - Foundation Pack: Concepts and Technologies Guide* for definitions of fundamental Oracle AIA concepts and information about:

- Oracle AIA
- Enterprise business objects (EBOs) and enterprise business messages (EBMs)
- Enterprise business services (EBSs)
- Application business connector (ABC) services
- Interaction patterns
- Extensibility
- Versioning
- Business processes
- Batch processing
- Infrastructure services
- Security

Oracle AIA PIP Implementation Guides

A PIP is a pre-built set of integrated orchestration flows, application integration logic, and extensible EBOs and EBSs required to manage the state and execution of a defined set of activities or tasks between specific Oracle applications associated with a given process.

A PIP provides everything you need to deploy a selected integrated business process area. The PIP product offering is suited to those customers seeking to rapidly implement a discreet business process.

Copyright © 2009, Oracle. All rights reserved.

Implementation guides are available for each PIP.

Additional Resources

The following resources are also available:

Resource	Location
Oracle Application Integration Architecture - Installation and Upgrade Guide	Metalink: https://metalink.oracle.com/
Known issues, workarounds, and most current list of patches	Metalink: https://metalink.oracle.com/
Release Notes	Oracle Technology Network: http://www.oracle.com/technology/
Documentation updates	Metalink: https://metalink.oracle.com/

Part 1: Using the BSR

Chapter 1: Understanding the BSR

This chapter discusses:

- The Business Service Repository (BSR).
- The BSR architecture.
- Supported BSR object types.
- Available annotations for supported BSR object types.

The BSR

The BSR is a service-oriented architecture (SOA) repository. It stores and provides information about the objects, messages, and services that compose the integration scenarios in your Oracle Application Integration Architecture (AIA) ecosystem.

These integration scenarios are conversations that constitute a business process. They represent end-to-end communication paths between requestor applications, through the Oracle AIA integration layer, and provider applications. The Oracle AIA Process Integrations (PIs) delivered in Oracle AIA process integration packs (PIPs) are composed of integration scenarios.

For example, the Agent Assisted Customer Care PIP includes a number of integration points. One of these integration points is the Account Balance PI, which includes two integration scenarios:

- Query Account Balance Summary
- Query Account Balance Details

An integration scenario can be further divided into three types:

- Requestor type.

  Outlines the path from requestor applications to the Oracle AIA canonical layer, such as an enterprise business service (EBS).

- Enterprise business flow (EBF) type.

  Represents the orchestration flow that choreographs multiple business capabilities, each of which is represented as an EBS or EBF. The EBS contains routing rules that mediate runtime messages to appropriate provider applications.

- Provider type.

  Outlines the remaining communication path to target provider applications.

As a SOA repository, the BSR augments Oracle Service Registry (OSR) Universal Description, Discovery and Integration (UDDI) functionality. By combining a repository and registry, the BSR and OSR provide comprehensive coverage of artifacts, from static and abstract assets to run-time and concrete services that are used throughout the SOA life cycle, from design-time through run-time.

The registry and repository contents are in sync with Oracle SOA Suite design- and run-time resources. For example, Oracle AIA design-time deliverables, consisting of the interfaces of EBSs, application business connector (ABC) services, and so forth, can be published into the BSR without having to be deployed and up and running beforehand.

Once the services are deployed into the SOA engine and are subsequently published into the OSR, the BSR is also capable of providing links to respective OSR runtime entries. These OSR runtime service entries contain concrete WSDLs, such as the actual endpoint URLs at which these services may be running.

An OSR implementation is not required for installation of Oracle AIA and the BSR. You can run the Oracle AIA Installer without having an OSR implementation in place. At a later time, you can add the OSR to your implementation and run an Oracle AIA command-line script to populate Oracle AIA contents into the OSR. Running this command-line script will maintain data consistency between the BSR and OSR, including links between the two.

For more information about running these command-line scripts, see Performing the Initial Load of Design-Time Content to the OSR.

A service repository serves the following activities in an organization: service architecture, design, and development. It provides the ability to attach unstructured documentation to the assets it contains and provides knowledge management and a human interface for discovering assets. It can be used both at design- and run-time.

A service registry has traditionally served a different user than a service repository. A service registry stores metadata related to a particular asset you may have interest in, without actually containing the asset. A service registry is UDDI-based and is essentially an online directory of running services that enables service providers to advertise their offerings and allows service consumers to find services that match their criteria.

The BSR's primary objectives are to provide the following:

- A mechanism for publishing and removing content into the BSR.

  The BSR includes scripts that enable you to publish single or multiple WSDL, integration scenario XML, and XSD files into the BSR and the UDDI registry.

For more information, see Chapter 2: Publishing and Unpublishing Content in the BSR and OSR.

- A user interface (UI) for discovering and learning about the SOA portfolio in your Oracle AIA ecosystem.

  The BSR presents the components of your integrations in a centrally managed and searchable repository, enabling it to become the system of record for your business services and their topologies.

Copyright © 2009, Oracle. All rights reserved.

For more information, see Chapter 3: Using the BSR UI to View Integration Scenarios.
For more information, see Chapter 4: Adding and Editing Draft Integration Scenarios.

- Oracle AIA setup UI.

  The BSR provides UI pages used to support and administer various components and activities in your Oracle AIA ecosystem.

  - Application Registry page.

    This UI enables you to manage information about the applications participating in your Oracle AIA ecosystem.

For more information, see Chapter 5: Managing the Oracle AIA Application Registry.

  - Error Notifications page.

    This UI enables you to manage mappings between actor and FYI roles and their participating applications for use during Oracle AIA error notifications.

For more information, see Setting Up Error Notifications for Oracle AIA Processes.

  - Configurations page.

    This UI enables you to load any updates to the Oracle AIA configuration properties file without having to reboot the server.

For more information, see Chapter 7: Loading Oracle AIA Configuration File Updates.

  - Flex-Field page.

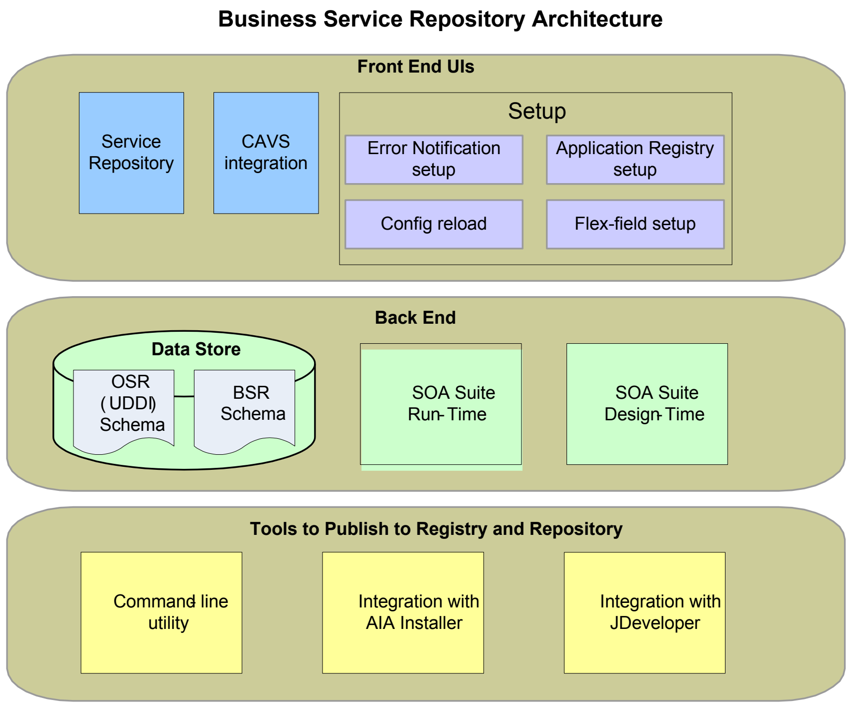
    This UI enables privileged users to declare and attach flex fields, user-defined annotations, and taxonomies, to SOA assets in the repository. The flex fields in turn become search filters.

Potential BSR users are active across the span of the SOA artifact life cycle and include functional and business analysts, architects, developers, system integrators, and system administrators. For example, a business analyst working on requirements for building out a particular business process can use the BSR integration scenario UI to find out which services are available in a particular integration area and then determine which additional services may need to be built. System administrators can use the BSR setup UI to manage the Oracle AIA ecosystem's application registry.

In support of SOA governance, the BSR provides visibility into the individual services (enterprise business services and application business connector services) and objects (enterprise business objects and enterprise business messages) in your Oracle AIA ecosystem. The BSR also provides the relationships between these services and objects and their metadata. Most importantly, through integration scenarios, the BSR showcases the context in which the artifacts are tied together to support end-to-end process integrations. Impact analysis can performed based on these complex relationships

The BSR Architecture

This diagram illustrates a high-level view of the BSR architecture.



BSR architecture

The BSR front end includes the Service Repository page, which provides a UI for discovering and learning about the SOA portfolio in your Oracle AIA ecosystem. Through an integration with the Composite Application Validation System (CAVS), the BSR front end also provides a UI for defining, running, and analyzing CAVS tests. Use the BSR-provided setup pages to define and maintain error notification mappings and the application registry for your Oracle AIA ecosystem.

For more information, see Working with the CAVS.

The BSR provides a few options for publishing content into the registry and repository. The BSR is integrated with the Oracle AIA Installer, which performs the initial load of design-time data into the BSR as a part of the installation process. The Oracle AIA Installer also performs the initial load of the deployed PIP run-time data into the OSR.

We also provide publication scripts that can be used at design- or run-time to publish individual or batch pieces of content, allowing you to keep registry and repository content up to date. You may also choose to publish individual XSD and WSDL files to the registry only, using the OSR UI.

For more information, see Chapter 2: Publishing and Unpublishing Content in the BSR and OSR.

Copyright © 2009, Oracle. All rights reserved.

Note: Oracle AIA content publishing follows UDDI v3 specifications. The OSR is UDDI v3-compliant. Oracle AIA artifacts should be portable across any UDDI v3-compliant registries. BSR security is provided by the Oracle Application Development Framework JAZN user store.

Object Types Supported in the BSR

The following object types can be published into the BSR and registry.

Note: Successful publication of these objects is dependent on them passing required annotations for which validations have been built into the publication utility logic.

For more information, see Annotations Available for Supported Object Types.

- XSD

  The XSD files published into the registry represent enterprise business objects (EBOs) and enterprise business messages (EBMs) in your Oracle AIA ecosystem.

- WSDL

  The WSDL files published into the registry represent EBSs and ABC services in your Oracle AIA ecosystem.

- Integration scenario XML

  The integration scenario XML files published into the BSR represent integration scenarios in your Oracle AIA ecosystem.

Annotations Available for Supported Object Types

When developed according to Oracle AIA standards, these object types contain sections that are meant to hold annotations about the object. These annotations serve as the source of truth about the object, providing metadata that is made available using the BSR and registry to help educate and inform users about the object.

Note: Before you use the BSR publication scripts to update BSR and registry content, ensure that you have properly annotated the WSDL, XSD, and integration scenario XML files that you want to publish according to the requirements referred to in this section. If you have not made all required annotations, you may not see expected results in the BSR and registry.

XSD Annotations

Annotations available for EBOs include:

```
   <xsd:annotation>
      <xsd:documentation>
         <svcdoc:EBO>
            <svcdoc:Description>Description of the EBO. There is a 255-
   characters limit.
            </svcdoc:Description>
            <svcdoc:Type>EBO</svcdoc:Type>
            <svcdoc:Industry>Industry, such as Telco
            </svcdoc:Industry>
         </svcdoc:EBO>
      </xsd:documentation>
   </xsd:annotation>
```

Annotations available for EBMs are as follows:

```
<xsd:annotation>
      <xsd:documentation>
            <svcdoc:EBO>
            <svcdoc:Description>
            Description of the EBM. There is a 255-characters limit.
            </svcdoc:Description>
            <svcdoc:Type>EBM</svcdoc:Type>
            <svcdoc:Industry>Industry, such as Telco</svcdoc:Industry>
            <svcdoc:EBOName>
            Name of its associated EBO, such as
            InvoiceEBO</svcdoc:EBOName>
            </svcdoc:EBO>
      </xsd:documentation>
   </xsd:annotation>
```

WSDL Annotations

Use the EBS and ABC service WSDL templates and samples provided in <AIA Home>/samples/AIASamples/AIAServices-TemplateSampleWSDLs.zip as a guide to available annotations.

Integration Scenario XML Annotations

Use the integration scenario XML templates provided in <AIA Home>/Infrastructure/BSR/BSRSamples/IntegrationScenarios/IntegrationScenario.xml as a guide to available annotations.

Flex Fields

Complementary to the annotations embedded in the source artifacts, BSR introduces flex field capabilities, which allow end users, rather than developers, to declare and associate user-defined annotations, taxonomies, and tags, on BSR artifacts. The user-defined flex fields subsequently become search filters.

Copyright © 2009, Oracle. All rights reserved.

Chapter 2: Publishing and Unpublishing Content in the BSR and OSR

This chapter provides an overview of publishing and unpublishing content in the Business Service Repository (BSR) and Oracle Service Registry (OSR) and discusses how to:

- Run the BSR with UDDI technology.
- Perform the initial load of design-time content to the OSR.
- Update BSR and OSR content using publication scripts.
- Create taxonomies.
- Remove BSR and OSR content using publication scripts.
- Describe the output log

Understanding Publication of Content into the BSR and OSR

The Oracle Application Integration Architecture (AIA) Installer performs the initial load of design-time content into the BSR. This includes Foundation Pack Enterprise Object Library content, as well as process integration pack (PIP) content.

The Oracle AIA Installer also performs the initial load of deployed run-time PIP content into the Oracle Service Registry (OSR). Run-time Foundation Pack contents, such as EBSs, are deployed as a part of PIP activation and are pushed to the OSR as part of PIP deployment. This content is accessible in the deployed taxonomy tree.

A command-line script is provided to perform the initial load of design-time Foundation Pack Enterprise Object Library and PIP content into the OSR. This content is accessible in the shipped taxonomy tree.

Subsequent updates to BSR and registry content can be performed in several ways:

- You can publish service-oriented architecture (SOA) artifacts individually or in bulk using BSR publication scripts.
- You can publish individual XSD and WSDL files into the registry using the Oracle Service Registry (OSR) user interface (UI).

Note: Oracle AIA content publishing follows UDDI v3 specifications. The OSR is UDDI v3-compliant. Oracle AIA artifacts should be portable across any UDDI v3-compliant registries.

Running the BSR with UDDI Technology

When you install the BSR using the Oracle AIA Installer, you are given the option to use UDDI technology with the BSR.

For more information about enabling this option using the Oracle AIA Installer, see *Oracle Application Integration Architecture – Foundation Pack: Install and Upgrade Guide.*

If at any point after installation, you need to change your selection to enable or disable use of UDDI technology, you can accomplish this by manually editing the bsrConfig.properties file located in AIA_HOME/Infrastructure/BSR/conf, where AIA_HOME refers to the physical path of the Oracle AIA installation.

- To run without the UDDI, leave the “registry.url=” setting blank.
- To run with the UDDI, set the “registry.url=” setting to the UDDI registry URL. For example, “registry.url=http://adc6005.oracle.com:7777/registry”.

Performing the Initial Load of Design-Time Content to the OSR

The Oracle AIA Installer performs the initial load of Foundation Pack Enterprise Object Library and PIP design-time content into the BSR and run-time content for deployed PIPs into the OSR.

The initial load of design-time Foundation Pack Enterprise Object Library and PIP content into the OSR is accomplished by running the following command-line script anytime after the Oracle AIA installation is complete.

To perform the initial load of design-time content to the OSR:

1. Access a command line utility.
2. Change the directory to AIA_HOME/Infrastructure/install/scripts.
3. Use the appropriate command for your operating system:
   - For Linux: Run ***ant —noconfig -f FPBSRPublish.xml PublishBSRArtifactsToOSR***
   - For Windows: Run ***ant -f FPBSRPublish.xml PublishBSRArtifactsToOSR***

Note: The script will run for three to four hours. Do not bind the application server or install another PIP until the script completes.

4. Design-time content will be accessible using the shipped taxonomy tree.

Copyright © 2009, Oracle. All rights reserved.

Updating BSR and Registry Content Using Publication Scripts

While the initial load of content into the BSR and registry is performed by the Oracle AIA installer tool, you may need to update content in the BSR and registry post-installation. You can accomplish these content updates using the following scripts:

- bsrPublish.sh

  Publishes an individual WSDL or XSD file to the repository.

- bsrIntegScenPublish.sh

  Publishes an individual integration scenario XML file to the repository.

- bsrPublishShippedWSDL.sh

  Publishes a batch of WSDL and associated XSD files to the repository.

- bsrPublishShippedIntegScen.sh

  Publishes a batch of integration scenario XML files to the repository.

Once you have published updated content, you should be able to view services, objects, messages, and integration scenarios in the BSR. Additionally, you can also use the OSR to view updates to enterprise business services (EBSs), application business connector (ABC) services, and enterprise business objects (EBOs) in the context of their taxonomies.

When creating new taxonomy trees to categorize your own data (services, objects, and so forth), rather than altering Oracle AIA-delivered taxonomies, we recommend that you take an upgrade-safe approach and create your own taxonomies in the OSR.

For more information about using the OSR to create taxonomies, see *Oracle Fusion Middleware Service Registry Product Documentation,* "Taxonomy Management."

If you decide to alter Oracle AIA-delivered taxonomies, this will impact upgrade.

For more information about the Oracle AIA upgrade process, see the *Oracle Application Integration Architecture - Installation and Upgrade Guide.*

Note: Before you use the BSR publication scripts to update BSR and registry content, ensure that you have properly annotated the WSDL, XSD, and integration scenario XML files that you want to publish. If you have not made all required annotations, you may not see expected results in the BSR and registry.

In addition, before you publish an integration scenario, ensure that all services used in the integration scenario have already been published. If you attempt to publish an integration scenario for which a service has not been published, you will receive an error.

For more information, see Annotations Available for Supported Object Types.

Publishing Individual XSD and WSDL Files:

To publish an individual XSD or WSDL file to the registry

1. Open the command line utility that you want to use to run the UNIX shell script file.
2. Enter ./bsrPublish.sh <filepath or URL> < WSDL or XSD>

Note: During publication, WSDLs are identified as either design-time or run-time. A design-time WSDL is a WSDL that does not have any services or bindings defined. bsrPublish will inject a dummy service and binding to the design time WSDL. If you republish the design time WSDL you need to remove the dummy service and binding from the WSDL or restore the original WSDL.

Publishing Individual Integration Scenario XML Files:

To publish an individual integration scenario XML file to the BSR

1. Open the command line utility that you want to use to run the UNIX shell script file.
2. Enter ./bsrIntegScenPublish.sh <file path>

Publishing XSD and WSDL Files in Batch:

To publish a batch of XSD and WSDL files to the BSR and registry

1. Locate and make a backup copy of <aia.home>/Infrastructure/BSR/config/bsrWSDLLoad.xml.
2. Edit <aia.home>/Infrastructure/BSR/config/bsrWSDLLoad.xml to include all of the file URLs that you would like to publish to the BSR.
3. Open the command line utility that you want to use to run the UNIX shell script file.
4. Enter ./bsrPublishShippedWSDL.sh.

Note: During publication, WSDLs are identified as either design-time or run-time. A design-time WSDL is a WSDL that does not have any services or bindings defined. bsrPublishShippedWSD will inject a dummy service and binding to the design time WSDL. If you republish the design time WSDL you need to remove the dummy service and binding from the WSDL or restore the original WSDL.

Copyright © 2009, Oracle. All rights reserved.

Publishing Integration Scenario XML Files in Batch:

To publish a batch of integration scenario XML files to the BSR

1. Open the command line utility that you want to use to run the UNIX shell script file.
2. Enter ./bsrPublishShippedIntegScen.sh.

Creating Taxonomies

When creating new taxonomy trees to categorize your own data (services, objects, and so forth), rather than altering Oracle AIA-delivered taxonomies, we recommend that you take an upgrade-safe approach and create your own taxonomies in the OSR.

For more information about using the OSR to create taxonomies, see *Oracle Fusion Middleware Service Registry Product Documentation,* "Taxonomy Management."

If you decide to alter Oracle AIA-delivered taxonomies, this will impact upgrade.

For more information about the Oracle AIA upgrade process, see the *Oracle Application Integration Architecture - Installation and Upgrade Guide.*

Removing BSR and Registry Content Using Publication Scripts

You may need to remove content in the BSR and registry post-installation. You can accomplish this by using the following scripts:

- For Unix, the file is bsrUnpublish.sh
- For Windows, the file is bsrUnpublish.bat.
- For bulk removal:

  ```
  bsrUnpublish.sh -r -l <filelist xml>
  ```

- For a single file removal:

  ```
  bsrUnpublish.sh -r -u <wsdl or xsd or scenario file> -t
  <wsdl|xsd|scenario>
  ```

  where:

  ```
  -r remove
  -u filename or url
  -l filelist xml
  -t file type
  ```

Examples

To remove a list of files you need to compose an input xml file:

An example of input file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<BsrFileList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNameSpaceSchemaLocation="http://xml.oracle.com/aia/bsr/V1.0
bsrFileList.xsd">
   <Artifact>
<url>/slot/ems1684/product/10.1.3.2/OracleAS_1/Apacth/Apache/htdocs/
AIAComponents/EnterpriseBusinessServiceLibrary/Industry/Telco/ItemEB
S.wsdl</url>
        <type>WSDL</type>
   </Artifact>
<Artifact>
:
</Artifact>
</ BsrFileList >
```

Examples of bulk removal commands:

```
./bsrUnpublish.sh -r -l bsrWsdlLoad.xml
./bsrUnpublish.sh -r -l bsrxsdLoad.xml
./bsrUnpublish.sh -r -l bsrintegrationLoad.xml
```

To remove a single file you pass the file name and file type:

Examples of single file removal:

```
./bsrUnpublish.sh -r -u
http://xxx/AIAComponents/Telco/AccountBalance.wsdl -t wsdl

./bsrUnpublish.sh -r -u http://xxx/AIAComponents/Telco/ItemEBO.xsd -
t xsd

./bsrUnpublish.sh -r -u
http://xxx/PIPS/EnterpriseObject/Integration/AIASiebelABCSImplProvide
r.xml -t scenario
```

If -r is not specified, the program will run in read-only mode. It will print all information discovered without deleting anything.

If -r is specified, and the program detects dependency, the following logic is applied:

- If the program detects an integration dependency for a WSDL service, that is, the integration scenario has a reference to that service/WSDL, it will not delete the WSDL service. It will automatically switch to read-only mode. The service cannot be deleted until all integration dependencies are removed.
- The program will delete an integration scenario even if there are dependencies between requestor and provider.
- For XSD (EBO), it will delete the entry based on the XSD name and target namespace with no check on dependency.

Copyright © 2009, Oracle. All rights reserved.

- UDDI server tModels for service and XSD will be deleted if any are found.

Describing the Output Log

The output for all scripts is based on the configuration specified in <aia.home>/Infrastructure/BSR/config/log4j.properties. By default, the output will be written to bsraccess.log in the same directory that you execute the script. You can set different log levels and change output mechanism.

For more information, see Apache Software Foundation for log4j configuration.

- Example of log4j.properties

### direct log messages to stdout ###

log4j.appender.stdout	org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target	System.out
log4j.appender.stdout.layout	org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern	=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

### file appender

log4j.appender.file	org.apache.log4j.RollingFileAppender
log4j.appender.file.maxFileSize	1000KB
log4j.appender.file.maxBackupIndex	10
log4j.appender.file.File	bsraccess.log
log4j.appender.file.threshold	info
log4j.appender.file.layout	org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern	%d{ABSOLUTE} %5p %c{1}:%L - %m%n
log4j.rootLogger	debug, file

You can change info to debug to get more detailed output

- log4j.appender.file.threshold=debug

Chapter 3: Using the BSR UI to View Integration Scenarios

This chapter provides an overview of integration scenarios in the Business Service Repository (BSR) and discusses how to view integration scenarios in the BSR user interface (UI).

Understanding Integration Scenarios in the BSR

The BSR UI provides an interactive graphical view of each integration scenario in your Oracle Application Integration Architecture (AIA) ecosystem. These integration scenarios are conversations that constitute a business process. They represent end-to-end communication paths between requestor applications, through the Oracle AIA integration layer, and provider applications.

The Oracle AIA Process Integrations (PIs) delivered in Oracle Adaptive Business Solutions Process Integration Packs (PIPs) are composed of integration scenarios. For example, the Agent Assisted Customer Care PIP includes a number of integration points. One of these integration points is the Account Balance PI, which includes two integration scenarios:

- Query Account Balance Summary
- Query Account Balance Details

As implemented, each integration scenario is captured in two or three separate integration scenario XML artifacts that represent the end-to-end flow:

- Requestor type.

  This integration scenario type outlines the path from requestor applications to an enterprise business service (EBS).

- Enterprise business flow (EBF)

  This type outlines the EBF portion of integration scenarios. It represents the orchestration of business capabilities.

- Provider type.

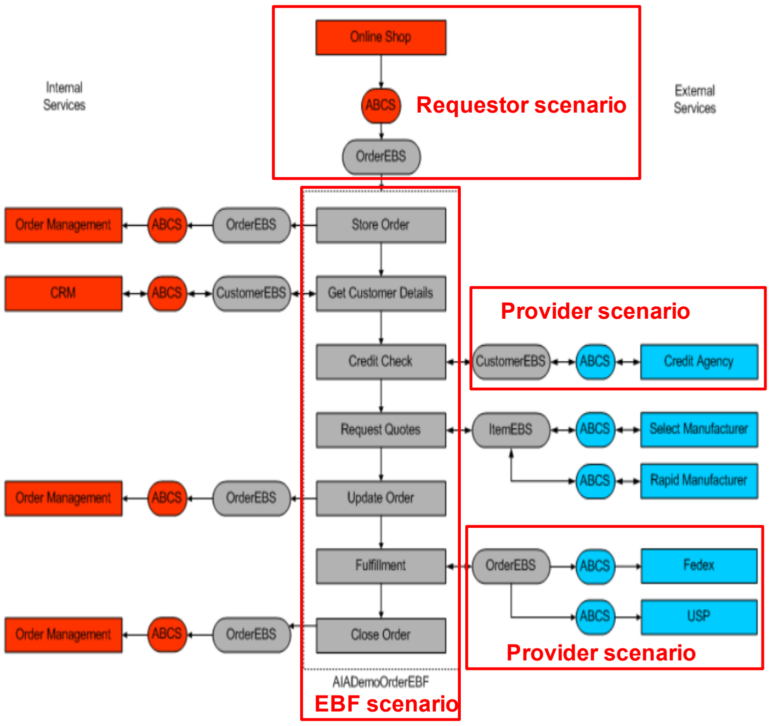
  This integration scenario type outlines the remaining half of the communication path to the target provider applications.

This diagram illustrates a fictitious example of order submission and fulfillment process. It demonstrates the three types of scenarios that could exist in the Oracle AIA ecosystem that could be captured in the BSR. For instance, an order is submitted from an online shop web UI. The order is subsequently routed by OrderEBS through an EBF. The EBF orchestrates the multiple business activities that need to take place on the order, including:

- Persisting or storing the order
- Getting customer details based on the order information

- Validating the customer's credit worthiness
- Requesting quotes
- Updating the order status after determining the manufacture
- Fulfilling the order by external shipping vendors before closing it.

The order fulfillment business process logic could be long running and traverse through multiple business capabilities, some of which may be internal to the company (such as persisting order or retrieving customer detail), while others may be external (such as credit check or quote). By using the three types of integration scenarios, we are able to represent the entire end-to-end process.



BSR scenarios

Note: Depending on the business logic involved, an integration may require multiple requestor and provider integration scenarios to convey an entire end-to-end flow.

These views into an integration scenario provide functional and business analysts, architects, developers, and system integrators with an overview of the relationships between web services and objects that make up the integration scenario. The BSR UI provides a way to discover the building blocks of an integration scenario and to see how they are related.

Viewing Integration Scenarios

This section discusses how to:

- View integration scenario summaries.
- View requestor integration scenario details.

Copyright © 2009, Oracle. All rights reserved.

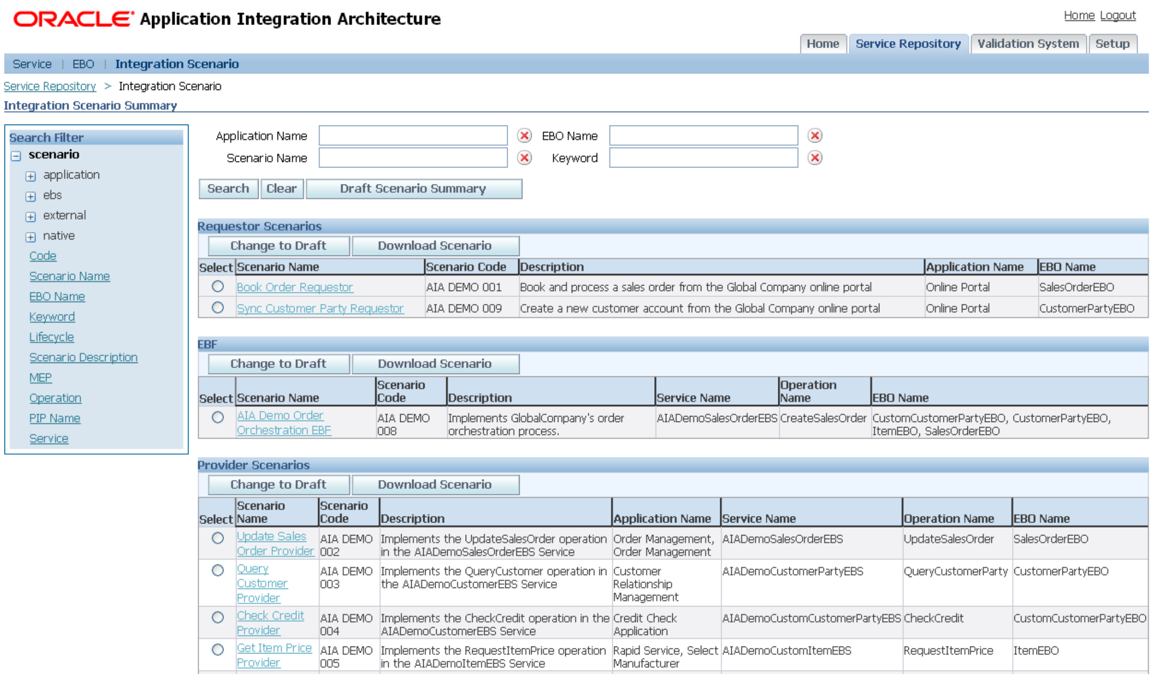
- View provider integration scenario details.
- View EBF integration scenario details.

Pages Used to View Integration Scenarios

Page Name	Navigation	Usage
Integration Scenario Summary	Access the Oracle AIA Console. Select the Service Repository tab.	Search for and view a summary of an integration scenario. Access a page that provides integration scenario details.
Integration Scenario	Click the Scenario Name link on the Integration Scenario Summary page.	View details about an integration scenario.

Viewing Integration Scenario Summaries

Access the Integration Scenario Summary page.



Requestor Scenarios

Select	Scenario Name	Scenario Code	Description	Application Name	EBO Name
○	Book Order Requestor	AIA DEMO 001	Book and process a sales order from the Global Company online portal	Online Portal	SalesOrderEBO
○	Sync Customer Party Requestor	AIA DEMO 009	Create a new customer account from the Global Company online portal	Online Portal	CustomerPartyEBO

EBF

Select	Scenario Name	Scenario Code	Description	Service Name	Operation Name	EBO Name
○	AIA Demo Order Orchestration EBF	AIA DEMO 008	Implements GlobalCompany's order orchestration process.	AIADemoSalesOrderEBS	CreateSalesOrder	CustomCustomerPartyEBO, CustomerPartyEBO, ItemEBO, SalesOrderEBO

Provider Scenarios

Select	Scenario Name	Scenario Code	Description	Application Name	Service Name	Operation Name	EBO Name
○	Update Sales Order Provider	AIA DEMO 002	Implements the UpdateSalesOrder operation in the AIADemoSalesOrderEBS Service	Order Management, Order Management	AIADemoSalesOrderEBS	UpdateSalesOrder	SalesOrderEBO
○	Query Customer Provider	AIA DEMO 003	Implements the QueryCustomer operation in the AIADemoCustomerEBS Service	Customer Relationship Management	AIADemoCustomerPartyEBS	QueryCustomerParty	CustomerPartyEBO
○	Check Credit Provider	AIA DEMO 004	Implements the CheckCredit operation in the AIADemoCustomerEBS Service	Credit Check Application	AIADemoCustomCustomerPartyEBS	CheckCredit	CustomCustomerPartyEBO
○	Get Item Price Provider	AIA DEMO 005	Implements the RequestItemPrice operation in the AIADemoItemEBS Service	Rapid Service, Select Manufacturer	AIADemoCustomItemEBS	RequestItemPrice	ItemEBO

Integration Scenario Summary page

Upon accessing the Integration Scenario Summary page, you are presented with default search filters you can use to locate specific integration scenarios. Users may optionally choose additional search filters (displayed in tree format on the left side of the page) to fine-tune their search results.

For more information about search filters, refer to the following sections.

Application Name	Enter the name of the application used in the integration scenario you are searching for.
Scenario Name	Enter the name of the integration scenario you are searching for.
EBO Name (enterprise business object name)	Enter the name of the EBO used by the integration scenario.
Keyword	Enter a single keyword that is relevant to the integration scenario you are searching for. Available keywords are derived from integration scenario annotations.
Search	Click to execute a search for integration scenarios using the search criteria you have entered on the page. To return all integration scenarios in the search results, clear all search fields and click **Search**.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page, where you can search for, create, and update draft integration scenarios. Only Administrator users have access to this page.

Requestor Scenarios, EBF, and Provider Scenarios

These search result grids display on clicking **Search**. Use them to view summaries of integration scenarios returned in a search. The search results vary by the type of scenario.

Requestor: A requestor integration scenario outlines the path from the requesting participating application to an enterprise business service (EBS).

EBF: An EBF outlines the enterprise business flow used in the integration scenario.

Provider: A provider integration scenario outlines the path from an EBS to a target providing participating application.

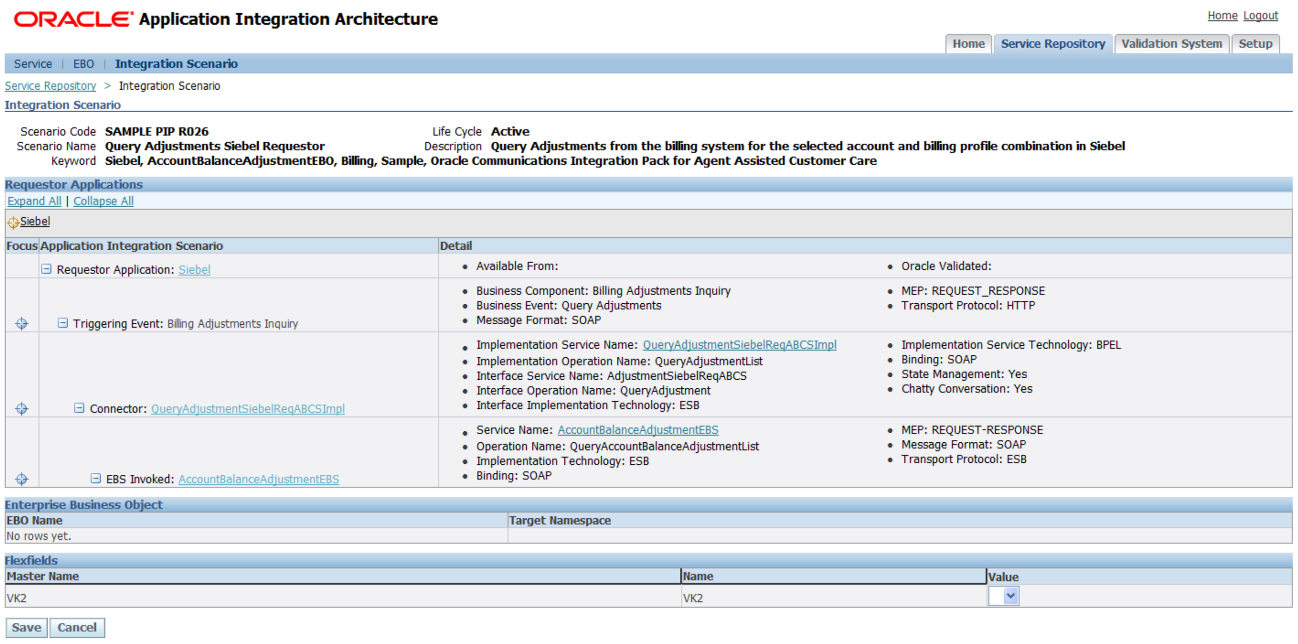
Change to Draft	Displays when integration scenarios are displayed in a search result grid. Only Administrator users have access to this page. Select the **Select** option and click **Change to Draft** to place the active integration scenario in draft status. An integration scenario in draft status is no longer accessible using the Integration Scenario Summary page, which displays only active integration scenarios. Integration scenarios in draft status are accessible using the Integration Scenario Draft Summary page, where you can work with draft integration scenarios. **For more information** about working with integration scenarios in draft status, see Chapter 4: Adding and Editing Draft Integration Scenarios.
Download Scenario	Displays when integration scenarios are displayed in a search result grid. Select the **Select** option and click **Download Scenario** to

Copyright © 2009, Oracle. All rights reserved.

	view or download the selected integration scenario. The default file name for a downloaded scenario is ***Scenario.xml.***
Scenario Name	Click to access the Integration Scenario page for the integration scenario.
Scenario Code	Displays the user-defined scenario code assigned to the integration scenario. This value is derived from the integration scenario annotation and should be unique.
Description	Displays a description of the integration scenario.
Application Name	For an integration scenario with a **Scenario Type** of ***Requestor***, displays the name of the requesting participating application. For an integration scenario with a **Scenario Type** of ***Provider***, displays the name of the providing participating application.
EBO Name	Displays the name of the EBO that is used by the integration scenario.
Service Name	Displays the name of the service used in the scenario.
Operation Name	Displays the name of the operation used in the scenario.

Viewing Requestor Integration Scenario Details

Access the Integration Scenario page for a requestor integration scenario.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Service | EBO | Integration Scenario
Service Repository > Integration Scenario

Integration Scenario

Scenario Code	SAMPLE PIP R026	Life Cycle	Active
Scenario Name	Query Adjustments Siebel Requestor	Description	Query Adjustments from the billing system for the selected account and billing profile combination in Siebel
Keyword	Siebel, AccountBalanceAdjustmentEBO, Billing, Sample, Oracle Communications Integration Pack for Agent Assisted Customer Care		

Requestor Applications

Expand All | Collapse All

Siebel

Focus	Application Integration Scenario	Detail	
	Requestor Application: Siebel	• Available From:	• Oracle Validated:
	Triggering Event: Billing Adjustments Inquiry	• Business Component: Billing Adjustments Inquiry • Business Event: Query Adjustments • Message Format: SOAP	• MEP: REQUEST_RESPONSE • Transport Protocol: HTTP
	Connector: QueryAdjustmentSiebelReqABCSImpl	• Implementation Service Name: QueryAdjustmentSiebelReqABCSImpl • Implementation Operation Name: QueryAdjustmentList • Interface Service Name: AdjustmentSiebelReqABCS • Interface Operation Name: QueryAdjustment • Interface Implementation Technology: ESB	• Implementation Service Technology: BPEL • Binding: SOAP • State Management: Yes • Chatty Conversation: Yes
	EBS Invoked: AccountBalanceAdjustmentEBS	• Service Name: AccountBalanceAdjustmentEBS • Operation Name: QueryAccountBalanceAdjustmentList • Implementation Technology: ESB • Binding: SOAP	• MEP: REQUEST-RESPONSE • Message Format: SOAP • Transport Protocol: ESB

Enterprise Business Object

EBO Name	Target Namespace
No rows yet.	

Flexfields

Master Name	Name	Value
VK2	VK2	

Integration Scenario page displaying a requestor integration scenario

Click **Expand All** to display all steps in the integration scenario. Content displayed for the integration scenario is derived from annotations provided in the integration scenario XML artifact.

When this page displays a requestor integration scenario, it outlines the path from requestor application to an application business connector (ABC) service to an EBS in the Oracle AIA integration layer. To view the remaining communication path from the EBS, to EBF, to provider connector services, and to the provider application, click the EBS Invoked link in the last step of the requestor integration scenario.

Life Cycle	Indicates the life cycle phase of the integration scenario. This value is set in the integration scenario XML. For example: ***Active*** ***Deprecated*** ***Obsolete*** ***Planned***

Requestor Applications

The **Requestor Applications** area displays for a requestor integration scenario. Use the controls on the page to click through the integration scenario communication flow starting with the requesting participating application and ending with the EBS.

Focus	Click the **Select to focus in** button ⌖ to focus on the selected step in the integration scenario communication flow. The selected step moves to the top of the table. Click the breadcrumb link for a step in the integration scenario communication flow to return focus to that step.
Application Integration Scenario	Displays the steps in the integration scenario communication flow.
Detail	Displays details about the steps in the integration scenario communication flow.

The **Requestor Application** row displays the name of the requesting participating application. The **Detail** column displays the following information about the requestor application.

Available From	Displays the name of the vendor or system integrator that built the integration scenario.
Oracle Validated	Indicates whether the integration scenario was built following a formal Oracle Application Integration Architecture methodology program.

The **Triggering Event** row displays the name of the triggering event for the requestor integration scenario. The **Detail** column displays the following information about the triggering event.

Generally, integration scenarios are triggered by events in the UI. For example, an agent may click a button in a requesting participating application. This UI event is translated into a business event, which is propagated from the requesting participating application to an ABC service and to an EBS in the Oracle AIA layer. This **Triggering Event** row is meant to convey the details of this business event.

Business Component	Displays the name of the business component associated with the triggering event.

Copyright © 2009, Oracle. All rights reserved.

Business Event	Displays the name of the business event associated with the triggering event.
Message Format	Displays the message format used by the triggering event. For example: ***SOAP*** ***XML***
MEP (message exchange pattern)	Displays the MEP used by the triggering event. For example: ***REQ UEST RESPONSE*** ***FIRE-AND-FORGET*** ***REQ UEST-DELAYEDRESPONSE***
Transport Protocol	Displays the transport protocol used by the triggering event. For example: ***HTTP*** ***JMS*** ***AQ***

The **Connector** row displays the name of the ABC implementation service used by the requestor integration flow. The **Detail** column displays the following information about the ABC implementation service.

Interface Service Name	When applicable, displays the name of the ABC interface service used by the requestor ABC implementation service.
Interface Operation Name	When applicable, displays the operation used by the ABC interface service.
Implementation Service Name	Displays the name of the requestor ABC implementation service used by the requestor integration scenario. This is presented as a link. When the ABCSImpl does exist in the BSR (such as the case for a PIP), you can go to the ABCSImpl detailed page by clicking the link. When this ABCSImpl is not captured by the BSR (Foundation Pack, for example, does not contain the ABC service), a warning message is prompted if users click the link.
Implementation Operation Name	Displays the operation used by the requestor integration scenario.
Interface Implementation Technology	When applicable, displays the technology used to implement the ABC service interface.
Implementation Service Technology	Displays the technology used to implement the requestor ABC implementation service. For example: ***BPEL*** ***ESB***

Binding	Displays the type of binding used by the requestor ABC implementation service.
State Management	Indicates whether the requestor ABC service is managing the state while having outbound interactions with the requestor application. For example: ***Yes*** ***No***
Chatty Conversation	Indicates whether the requestor ABC implementation service participates in a chatty conversation. For example: ***Yes*** ***No***

The EBS Invoked row displays the name of the EBS that is invoked by the requestor ABC implementation service. Click the EBS Invoked link to access the remaining steps in the communication path from the EBS in the Oracle AIA integration layer to optional EBF, and subsequently to the provider connector services, to the provider application.

For more information about provider scenarios, see the following section.

The **Detail** column displays the following information about the EBS.

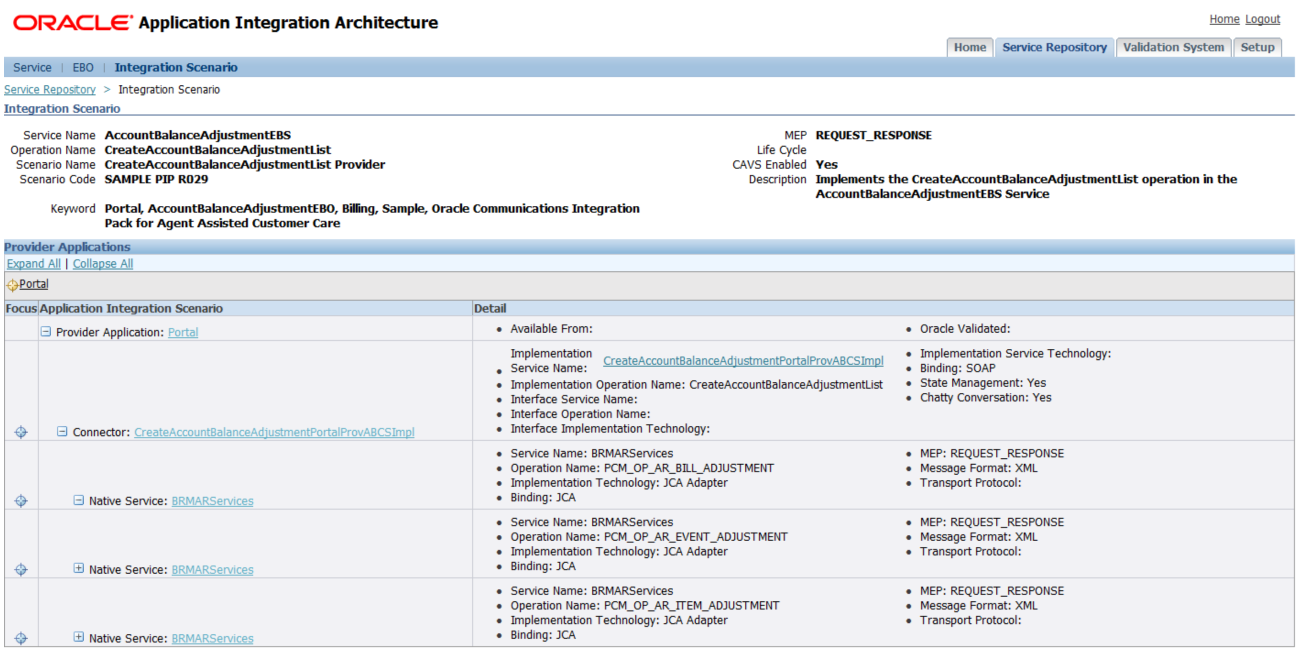
Service Name	Displays that name of the EBS invoked by the requestor ABC implementation service. This field is displayed as a link. When clicking the link, users are brought the EBS' detailed page)
Operation Name	Displays the EBS operation invoked by the requestor ABC implementation service.
Implementation Technology	Displays the technology used to implement the EBS. For example: ***ESB*** ***BPEL***
Binding	Displays the type of binding used by the EBS.
MEP (message exchange pattern)	Displays the MEP used by the EBS. For example: ***REQUEST-RESPONSE*** ***FIRE-AND-FORGET*** ***REQUEST-DELAYED RESPONSE***
Message Format	Displays the message format used by the EBS. For example: ***SOAP***

Copyright © 2009, Oracle. All rights reserved.

	XML
Transport Protocol	Displays the transport protocol used by the EBS.
	For example:
	HTTP
	JMS

Viewing Provider Integration Scenario Details

Access the Integration Scenario page for a provider integration scenario.



ORACLE Application Integration Architecture

Home Logout

Home | Service Repository | Validation System | Setup

Service | EBO | Integration Scenario

Service Repository > Integration Scenario

Integration Scenario

Service Name	AccountBalanceAdjustmentEBS	MEP	REQUEST_RESPONSE
Operation Name	CreateAccountBalanceAdjustmentList	Life Cycle	
Scenario Name	CreateAccountBalanceAdjustmentList Provider	CAVS Enabled	Yes
Scenario Code	SAMPLE PIP R029	Description	Implements the CreateAccountBalanceAdjustmentList operation in the AccountBalanceAdjustmentEBS Service
Keyword	Portal, AccountBalanceAdjustmentEBO, Billing, Sample, Oracle Communications Integration Pack for Agent Assisted Customer Care		

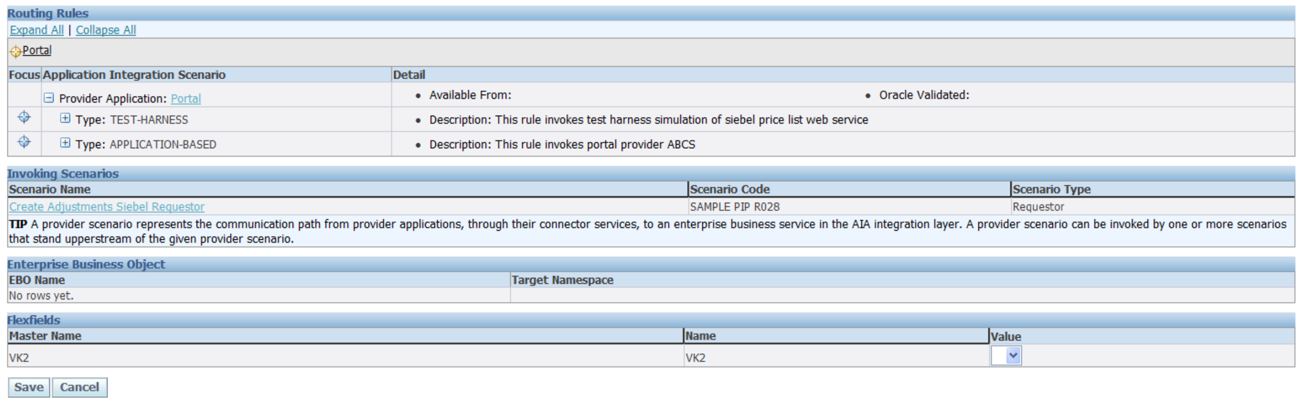
Provider Applications

Expand All | Collapse All

Portal

Focus	Application Integration Scenario	Detail	
	Provider Application: Portal	• Available From:	• Oracle Validated:
	Connector: CreateAccountBalanceAdjustmentPortalProvABCSImpl	• Implementation Service Name: CreateAccountBalanceAdjustmentPortalProvABCSImpl • Implementation Operation Name: CreateAccountBalanceAdjustmentList • Interface Service Name: • Interface Operation Name: • Interface Implementation Technology:	• Implementation Service Technology: • Binding: SOAP • State Management: Yes • Chatty Conversation: Yes
	Native Service: BRMARServices	• Service Name: BRMARServices • Operation Name: PCM_OP_AR_BILL_ADJUSTMENT • Implementation Technology: JCA Adapter • Binding: JCA	• MEP: REQUEST_RESPONSE • Message Format: XML • Transport Protocol:
	Native Service: BRMARServices	• Service Name: BRMARServices • Operation Name: PCM_OP_AR_EVENT_ADJUSTMENT • Implementation Technology: JCA Adapter • Binding: JCA	• MEP: REQUEST_RESPONSE • Message Format: XML • Transport Protocol:
	Native Service: BRMARServices	• Service Name: BRMARServices • Operation Name: PCM_OP_AR_ITEM_ADJUSTMENT • Implementation Technology: JCA Adapter • Binding: JCA	• MEP: REQUEST_RESPONSE • Message Format: XML • Transport Protocol:

Integration Scenario page displaying provider integration scenario (1 of 2)



Routing Rules

Expand All | Collapse All

Portal

Focus	Application Integration Scenario	Detail	
	Provider Application: Portal	• Available From:	• Oracle Validated:
	Type: TEST-HARNESS	• Description: This rule invokes test harness simulation of siebel price list web service	
	Type: APPLICATION-BASED	• Description: This rule invokes portal provider ABCS	

Invoking Scenarios

Scenario Name	Scenario Code	Scenario Type
Create Adjustments Siebel Requestor	SAMPLE PIP R028	Requestor

TIP A provider scenario represents the communication path from provider applications, through their connector services, to an enterprise business service in the AIA integration layer. A provider scenario can be invoked by one or more scenarios that stand upperstream of the given provider scenario.

Enterprise Business Object

EBO Name	Target Namespace
No rows yet.	

Flexfields

Master Name	Name	Value
VK2	VK2	

Save | Cancel

Integration Scenario page displaying provider integration scenario (2 of 2)

Click **Expand All** to display all steps in the integration scenario. Content displayed for the integration scenario is derived from annotations provided in the integration scenario XML artifact.

When this page displays a provider integration scenario, it can illustrate the communication flow to an EBS, EBF, external service, or provider application native services.

For example, if this page displays a provider integration scenario that starts with an EBS, it can outline the communication path from the EBS to an ABC service to an external service or provider application native services, and ultimately reach the participating edge applications.

If this page displays a provider integration scenario that starts with an EBF, it can outline the communication path from the EBF to another EBF or to an EBS.

If the provider integration scenario includes an EBS or EBF, you can click the EBS Invoked link to view the next step in the communication flow. Multiple EBS Invoked links indicate that the service has multiple provider applications.

Service Name	Provides the name of the service triggers the provider integration scenario.
Operation Name	Provides the name of the operation triggers the provider integration scenario.
Scenario Code	Displays the user-defined scenario code assigned to the integration scenario. This value is derived from the integration scenario annotation and should be unique.
MEP (message exchange pattern)	Displays the message exchange pattern used by the service. For example: ***REQUEST-RESPONSE*** ***FIRE-AND-FORGET*** ***REQUEST-DELAYEDRESPONSE***
Life Cycle	Indicates the life cycle phase of the integration scenario. This value is set in the integration scenario XML. For example: ***Active*** ***Deprecated*** ***Obsolete*** ***Planned***
CAVS Enabled	Indicates whether the integration scenario has been developed to work with testing functionality provided by the Composite Application Validation System (CAVS).

For more information, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Developing CAVS-Enabled Oracle AIA Services"

Copyright © 2009, Oracle. All rights reserved.

Provider Applications

The **Provider Applications** area displays for a provider integration scenario. Use the controls on the page to click through the provider integration scenario communication flow. If this path includes an EBS or an EBF, you can click the EBS Invoked link to view the next step in the communication flow. Multiple EBS Invoked links indicate that the service has multiple provider applications.

Focus	Click the **Select to focus in** button to focus on the selected step in the integration scenario communication flow. The selected step moves to the top of the table. Click the breadcrumb link for a step in the integration scenario communication flow to return focus to that step.
Application Integration Scenario	Displays the steps in the integration scenario communication flow.
Detail	Displays details about the steps in the integration scenario communication flow.

The **Provider Application** row displays the name of the provider participating application. The **Detail** column displays the following information about the provider application.

Available From	Displays the name of the vendor or system integrator that built the integration scenario.
Oracle Validated	Indicates whether the integration scenario was built following a formal Oracle Application Integration Architecture methodology program.

The **Connector** row displays the name of the connector service used by the provider integration flow. The **Detail** column displays the following information about the connector service.

Interface Service Name	When applicable, displays the name of the ABC interface service used by the connector service.
Interface Operation Name	When applicable, displays the ABC interface service operation used by the connector service.
Implementation Service Name	Displays the name of the connector service used by the provider integration scenario. This is presented as a link. When the ABCSImpl does exist in the BSR (such as the case for a PIP), users can go to the ABCSImpl detailed page by clicking the link. When this ABCSImpl is not captured by the BSR (Foundation Pack, for example, does not contain ABC services), a warning message is prompted if users click the link.
Implementation Operation Name	Displays the name of the operation used by the provider integration scenario.
Interface Implementation Technology	When applicable, displays the technology used to implement the ABC interface service.
Implementation Service Technology	Displays the technology used to implement the connector service. For example:

	BPEL
	ESB
Binding	Displays the type of binding used by the connector service.
State Management	Indicates whether the provider connector service is managing the state while having outbound interactions with the provider application. For example: ***Yes*** ***No***
Chatty Conversation	Indicates whether the connector service participates in a chatty conversation. For example: ***Yes*** ***No***

The connector service can be associated with native services, external services, and EBSs. The **Detail** column displays the following information for possible Native Service, External Service, and EBS Invoked rows under the **Connector** row.

Service Name	Displays that name of the service invoked by the connector.
Operation Name	Displays the name of the service operation invoked by the connector.
Implementation Technology	Displays the technology used to implement the service.
Binding	Displays the type of binding used by the service.
MEP (message exchange pattern)	Displays the MEP used by the service. For example: ***REQUEST-RESPONSE*** ***FIRE-AND-FORGET*** ***REQUEST-DELAYED RESPONSE***
Message Format	Displays the message format used by the service. For example: ***SOAP*** ***XML***
Transport Protocol	Displays the transport protocol used by the service. For example: ***HTTP*** ***JMS***
Vendor	For external services only, displays the name of the vendor or system integrator that built the service.

Copyright © 2009, Oracle. All rights reserved.

Invoking Scenarios

The **Invoking Scenarios** grid provides information about the integration scenarios that can invoke the displayed provider integration scenario. Provider integration scenarios can be invoked by one or more integration scenarios that reside upstream from the given provider integration scenario.

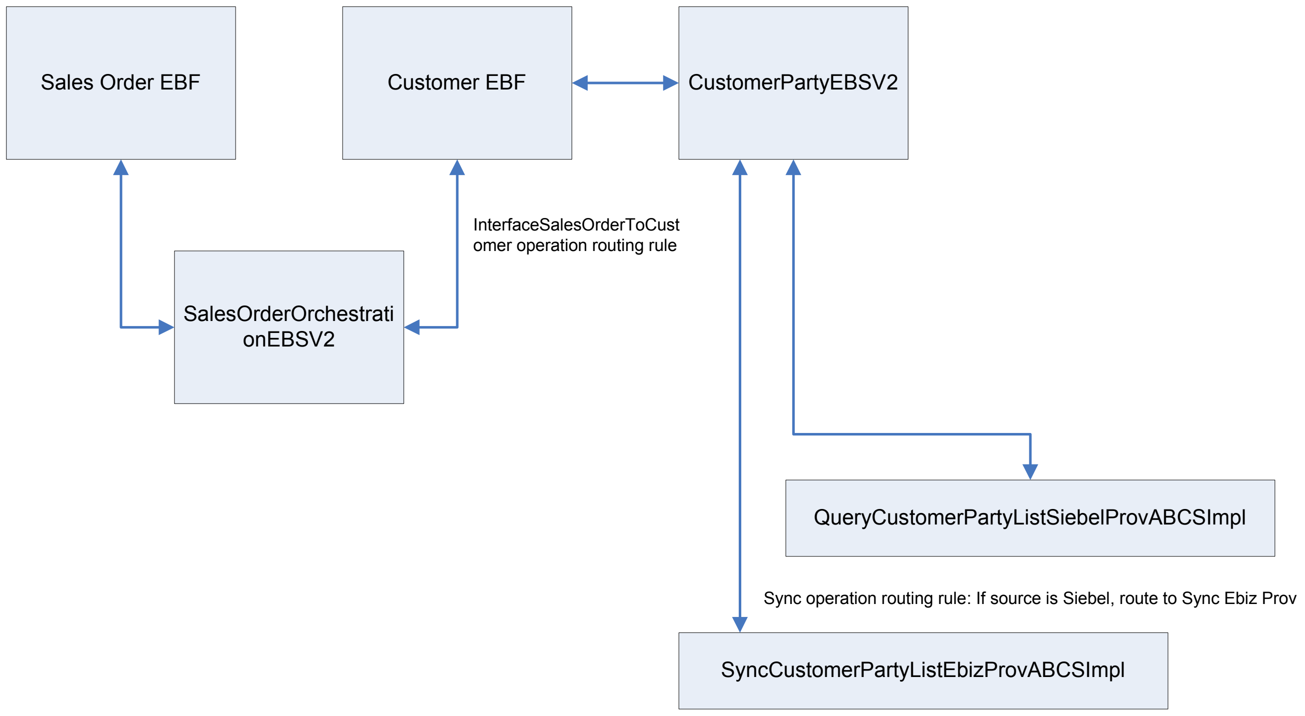
Click the **Scenario Name** link to access the requester integration scenario details on the Integration Scenario page.

For more information, see Viewing Requestor Integration Scenario Details.

Routing Rules

The provider scenarios originate from the Oracle AIA canonical layer, such as an EBS. It is often the case that the EBS performs mediation during runtime to route and delegate the real business logic to the back-end provider scenarios and providing applications. The routing rules of such delegation logic are manifested in the Routing Rules section. Ordinarily, one routing rule in this section would be the Test Harness routing rule, which dictates that the execution flow would be routed/delegated to the Composite Application Validation System (CAVS) if the test harness is enabled. Otherwise, the normal application logic takes precedence during normal execution.

Routing Rules Example


Sales Order EBF
Customer EBF
CustomerPartyEBSV2
InterfaceSalesOrderToCustomer operation routing rule
SalesOrderOrchestrationEBSV2
QueryCustomerPartyListSiebelProvABCSImpl
Sync operation routing rule: If source is Siebel, route to Sync Ebiz Prov
SyncCustomerPartyListEbizProvABCSImpl

Sales order to fulfillment scenario

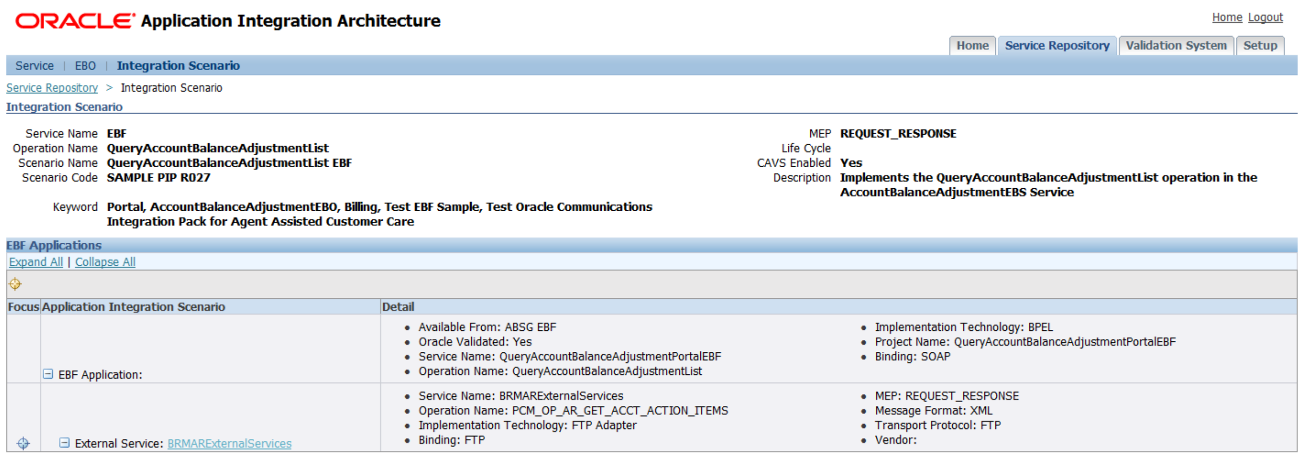
The diagram illustrates a sales order to fulfillment scenario. In this case, SalesOrderOrchestrationEBSV2 will route the request from the sales order flow to the customer flow. When a query operation is invoked in the customer flow, CustomerPartyEBSV2 will route the request to QueryCustomerPartyListSiebelProvABCSImpl. When a sync operation is invoked, CustomerPartyEBSV2 will route the request to SyncCustomerPartyListEbizProvABCSImpl.

In the BSR, the routing rules table of an EBF/provider scenario will contain the rules which route the request from the EBS to itself. For example, the routing rules of Customer EBF would have rule of SalesOrderOrchestrationEBSV2 while the routing rules of QueryCustomerPartyListSiebelProvABCSImpl would have a routing rule of query operation in CustomerPartyEBSV2.

We deliver routing rules of the sales order flow to customer flow however the user can change the routing rules to point to their own customer business flow.

Viewing EBF Details

Access the Integration Scenario page for an EBF.


ORACLE Application Integration Architecture
Home Logout
Home
Service Repository
Validation System
Setup
Service | EBO | Integration Scenario
Service Repository > Integration Scenario

Integration Scenario

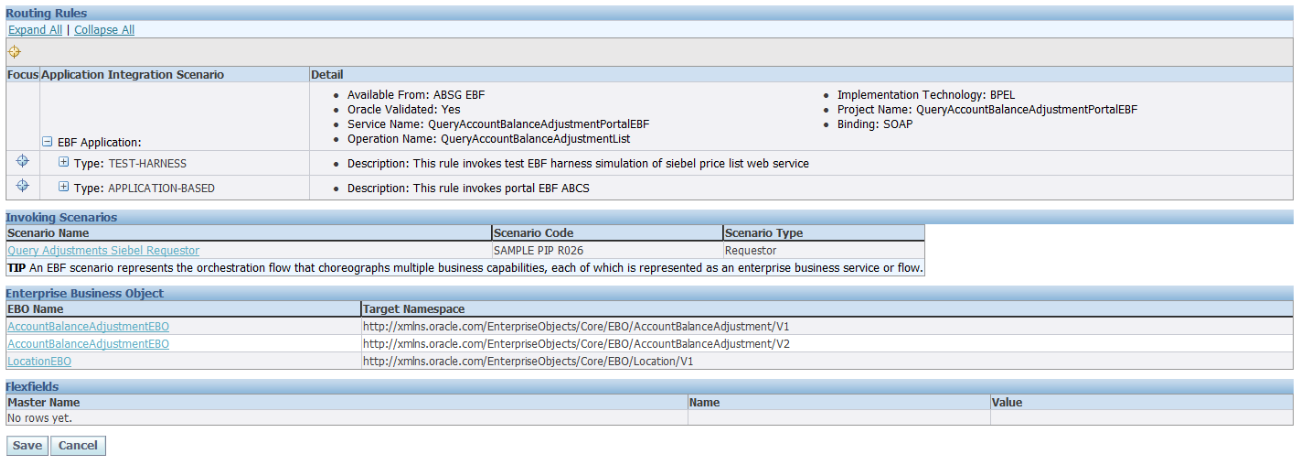
Service Name	EBF
Operation Name	QueryAccountBalanceAdjustmentList
Scenario Name	QueryAccountBalanceAdjustmentList EBF
Scenario Code	SAMPLE PIP R027
Keyword	Portal, AccountBalanceAdjustmentEBO, Billing, Test EBF Sample, Test Oracle Communications Integration Pack for Agent Assisted Customer Care
MEP	REQUEST_RESPONSE
Life Cycle	
CAVS Enabled	Yes
Description	Implements the QueryAccountBalanceAdjustmentList operation in the AccountBalanceAdjustmentEBS Service

EBF Applications

Expand All | Collapse All

Focus	Application Integration Scenario	Detail	
	EBF Application:	• Available From: ABSG EBF • Oracle Validated: Yes • Service Name: QueryAccountBalanceAdjustmentPortalEBF • Operation Name: QueryAccountBalanceAdjustmentList	• Implementation Technology: BPEL • Project Name: QueryAccountBalanceAdjustmentPortalEBF • Binding: SOAP
	External Service: BRMARExternalServices	• Service Name: BRMARExternalServices • Operation Name: PCM_OP_AR_GET_ACCT_ACTION_ITEMS • Implementation Technology: FTP Adapter • Binding: FTP	• MEP: REQUEST_RESPONSE • Message Format: XML • Transport Protocol: FTP • Vendor:

Integration Scenario page displaying EBF (1 of 2)



Routing Rules

Expand All | Collapse All

Focus	Application Integration Scenario	Detail	
	EBF Application:	• Available From: ABSG EBF • Oracle Validated: Yes • Service Name: QueryAccountBalanceAdjustmentPortalEBF • Operation Name: QueryAccountBalanceAdjustmentList	• Implementation Technology: BPEL • Project Name: QueryAccountBalanceAdjustmentPortalEBF • Binding: SOAP
	Type: TEST-HARNESS	• Description: This rule invokes test EBF harness simulation of siebel price list web service	
	Type: APPLICATION-BASED	• Description: This rule invokes portal EBF ABCS	

Invoking Scenarios

Scenario Name	Scenario Code	Scenario Type
Query Adjustments Siebel Requestor	SAMPLE PIP R026	Requestor

TIP An EBF scenario represents the orchestration flow that choreographs multiple business capabilities, each of which is represented as an enterprise business service or flow.

Enterprise Business Object

EBO Name	Target Namespace
AccountBalanceAdjustmentEBO	http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/AccountBalanceAdjustment/V1
AccountBalanceAdjustmentEBO	http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/AccountBalanceAdjustment/V2
LocationEBO	http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/Location/V1

Flexfields

Master Name	Name	Value
No rows yet.		

Integration Scenario page displaying EBF (2 of 2)

Service Name	Provides the name of the service that triggers the EBF.

Copyright © 2009, Oracle. All rights reserved.

Operation Name	Provides the name of the operation triggers the EBF.
Scenario Code	Displays the user-defined scenario code assigned to the EBF. This value is derived from the annotation and should be unique.
MEP (message exchange pattern)	Displays the message exchange pattern used by the EBF. For example: ***REQUEST-RESPONSE*** ***FIRE-AND-FORGET*** ***REQUEST-DELAYED RESPONSE***
Life Cycle	Indicates the life cycle phase of the EBF. This value is set in the XML. For example: ***Active*** ***Deprecated*** ***Obsolete*** ***Planned***
CAVS Enabled	Indicates whether the EBF has been developed to work with testing functionality provided by the Composite Application Validation System (CAVS). **For more information**, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Developing CAVS-Enabled Oracle AIA Services "
PIP Name	Displays the name of the PIP utilizing the EBF.

EBF Applications

Use the controls on the page to click through the EBF.

Available From	Displays the name of the vendor or system integrator that built the EBF.
Oracle Validated	Indicates whether the EBF was built following a formal Oracle AIA methodology program.
Service Name	Displays that name of the service that implements/represents the EBF.
Operation Name	Displays the name of the service operation that initiates/implements the EBF's logic.
Implementation Technology	Displays the technology used to implement the EBF such as Java, custom, or BPEL.
Binding	Displays the type of binding used by the EBF.

The EBF can be associated with native services, external services, and EBSs. The **Detail** column displays the following information for possible Native Service, External Service, and EBS Invoked rows under the EBF row.

Service Name	Displays that name of the (native, external, or EBS) service invoked by the EBF.
Operation Name	Displays the name of the service operation invoked by the EBF.
Implementation Technology	Displays the technology used to implement the service.
Binding	Displays the type of binding used by the service.
MEP (message exchange pattern)	Displays the MEP used by the service. For example: ***REQUEST-RESPONSE*** ***FIRE-AND-FORGET*** ***REQUEST-DELAYED RESPONSE***
Message Format	Displays the message format used by the service. For example: ***SOAP*** ***XML***
Transport Protocol	Displays the transport protocol used by the service. For example: ***HTTP*** ***JMS***
Vendor	For external services only, displays the name of the vendor or system integrator that built the service.

Invoking Scenarios

The grid displays the scenario that invokes the EBF with a link to the scenario and code and type details.

EBO

The grid displays the EBOs used in the EBF with a link to the EBO details page and the target namespace details.

Flexfields

The grid displays any flexfields that are related to the EBF.

Copyright © 2009, Oracle. All rights reserved.

Chapter 4: Adding and Editing Draft Integration Scenarios

This chapter provides an overview of adding and editing draft integration scenarios and discusses how to:

- Declare a process integration pack (PIP) in the Business Service Repository (BSR).
- Search for draft integration scenarios.
- View draft integration scenario synopses.
- Add and edit draft requestor integration scenarios.
- Add and edit draft provider integration scenarios.
- Add and edit draft enterprise business flow (EBF) integration scenarios.

Understanding Adding and Editing Draft Integration Scenarios

Integration scenarios that you create and access using the pages covered in this chapter are in draft status. This means that the integration scenarios you create and work with on these pages are in conceptual or development stages and are not considered ready for use in live integrations in your Oracle Application Integration Architecture (AIA).

When you are finished developing an integration scenario, you can place the integration scenario in active status and make it available for use in live integrations. You do this by accessing the Integration Scenario Draft Summary page, selecting the integration scenario, and clicking the Change to Active button. The integration scenario will no longer be in draft status and accessible using the Integration Scenario Draft Summary page. The integration scenario will be in active status and accessible using the Integration Scenario Summary page.

For more information about the Integration Scenario Summary page, see Viewing Integration Scenarios.

You can edit an active integration scenario using the pages covered in this chapter by placing it in draft status. You do this by accessing the Integration Scenario Summary page, selecting the integration scenario, and clicking the Change to Draft button. The selected integration scenario will no longer be active and accessible using the Integration Scenario Summary page. It will be in draft status and accessible using the Integration Scenario Draft Summary page and other pages covered in this chapter.

Declaring a PIP

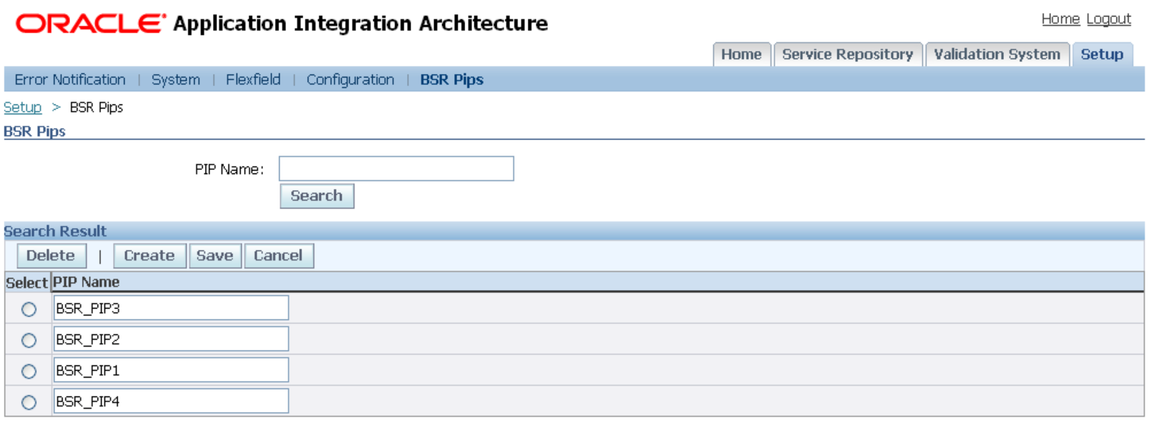
This chapter describes how to declare a PIP in the BSR. After declaring a PIP value using this page, you can specify PIP values on integration scenarios to group all relevant integration scenarios under respective PIPs.

Pages Used to Declare a PIP in the BSR

Page Name	Navigation	Usage
BSR Pips	Access the Oracle AIA Console. Select the Setup tab. Click the PIPs link.	Declare a PIP in the BSR so that it is available for selection on the pages used to add and edit draft integration scenarios.

Declaring a PIP in the BSR

Access the BSR Pips page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Error Notification | System | Flexfield | Configuration | BSR Pips
Setup > BSR Pips
BSR Pips
PIP Name:
Search
Search Result
Delete | Create Save Cancel
Select PIP Name
BSR_PIP3
BSR_PIP2
BSR_PIP1
BSR_PIP4

BSR Pips page

PIP Name	Enter the name of the PIP you are searching for.
Search	Click to execute a search based on the PIP name you have entered.

Search Result

Delete	Displays when PIP names are present. Select the **Select** option for a PIP name and click **Delete** to delete the PIP declaration.
Create	Click to add a PIP declaration and display the PIP name field, where you can enter the name of the PIP you want to declare in the BSR.
Save	Click to save the page.

Copyright © 2009, Oracle. All rights reserved.

Cancel	Click to cancel any unsaved edits to the page.
PIP Name	Name of the PIP that has been declared in the BSR.

Searching for Draft Integration Scenarios

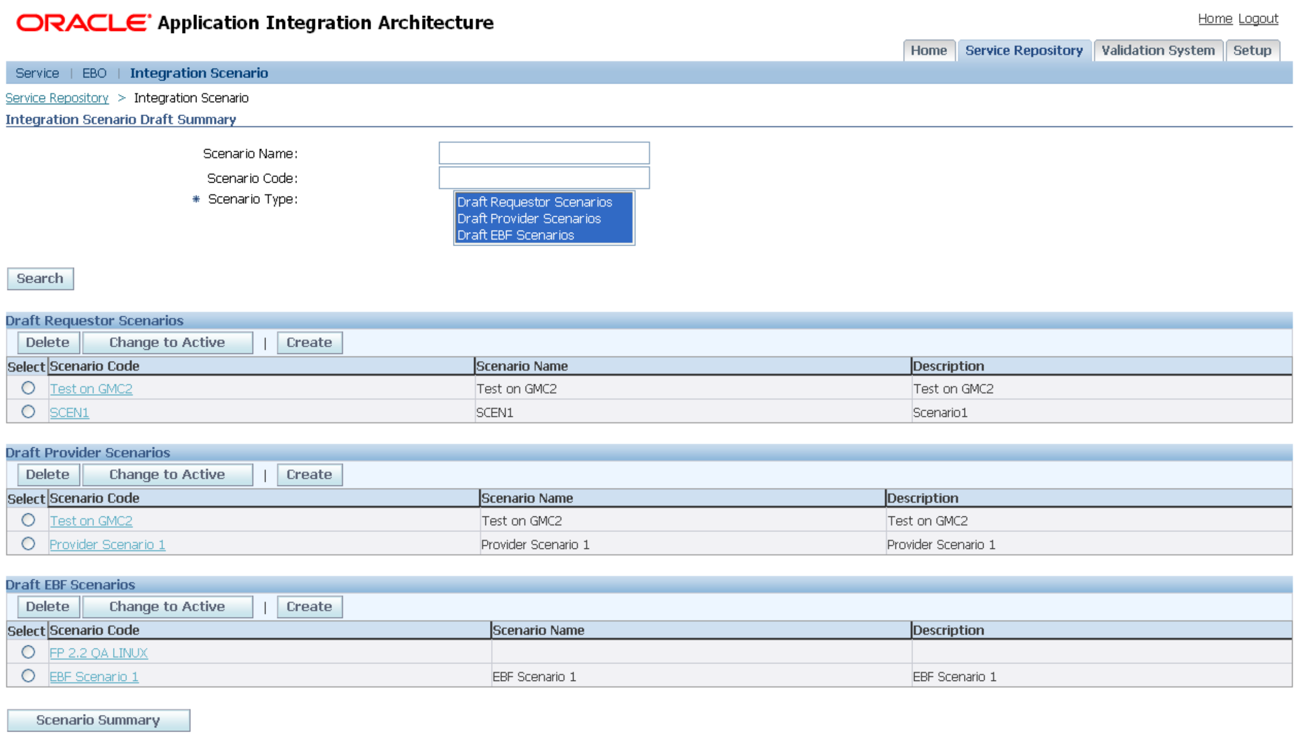
This section discusses how to search for draft integration scenarios.

Page Used to Search for Draft Integration Scenarios

Page Name	Navigation	Usage
Integration Scenario Draft Summary	Access the Oracle AIA Console. Select the Service Repository tab. Click the Integration Scenario link. Click Draft Scenario Summary.	Search for and display a list of integration scenarios that exist in the environment. Search criteria allow you to display integration scenarios by type.

Searching for a Draft Integration Scenario

Access the Integration Scenario Draft Summary page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Service | EBO | Integration Scenario
Service Repository > Integration Scenario
Integration Scenario Draft Summary

Scenario Name:

Scenario Code:

* Scenario Type: Draft Requestor Scenarios / Draft Provider Scenarios / Draft EBF Scenarios

Draft Requestor Scenarios

Delete | Change to Active | Create

Select	Scenario Code	Scenario Name	Description
○	Test on GMC2	Test on GMC2	Test on GMC2
○	SCEN1	SCEN1	Scenario1

Draft Provider Scenarios

Delete | Change to Active | Create

Select	Scenario Code	Scenario Name	Description
○	Test on GMC2	Test on GMC2	Test on GMC2
○	Provider Scenario 1	Provider Scenario 1	Provider Scenario 1

Draft EBF Scenarios

Select	Scenario Code	Scenario Name	Description
○	FP 2.2 QA LINUX		
○	EBF Scenario 1	EBF Scenario 1	EBF Scenario 1

Integration Scenario Draft Summary page

Scenario Name	Enter the name of the draft integration scenario you are searching for.

Scenario Code	Enter the code assigned to the integration scenario.
Scenario Type	Select the type of draft integration scenario you are searching for. ***Draft Requester Scenarios:*** A requestor integration scenario outlines the path from the requesting participating application to an enterprise business service (EBS). The draft table allows you to depict the topology. ***Draft Provider Scenarios:*** A provider integration scenario outlines the path from an EBS to a target providing participating application. The draft table allows you to depict the topology. ***Draft EBF Scenarios:*** An EBF scenario outlines the orchestration layer in the Oracle AIA. An EBF often involves the invocations of multiple canonical services and flows.
Search	Click to execute a search for draft integration scenarios using the search criteria you have entered on the page.

Draft Requestor Scenarios, Draft Provider Scenarios, and Draft EBF Scenarios

These search result grids display on clicking **Search**. Use them to view summaries of draft integration scenarios returned in a search. Search result values that display vary according to scenario type.

Delete	Select the **Select** option for an integration scenario and click **Delete** to delete the integration scenario.
Change to Active	Displays when integration scenarios are displayed in a search result grid. Select the **Select** option and click **Change to Active** to place the draft integration scenario in active status. An integration scenario in active status is no longer accessible using the Integration Scenario Draft Summary page, which displays only draft integration scenarios. Integration scenarios in active status are accessible using the Integration Scenario Summary page, where you can work with active integration scenarios. **For more information** about working with integration scenarios in active status, see Viewing Integration Scenarios.
Create	Click in the **Draft Requestor Scenarios** grid to access the Add/Edit Draft Requestor Scenario page, where you can create a draft requestor integration scenario. **For more information** about creating draft requestor scenarios, see Adding and Editing Requestor Integration Scenarios. Click in the **Draft Provider Scenarios** grid to access the

Copyright © 2009, Oracle. All rights reserved.

	Add/Edit Draft Provider Scenario page, where you can create a draft provider integration scenario.
	For more information about creating draft provider scenarios, see Adding and Editing Provider Integration Scenarios.
	Click in the **Draft EBF Scenarios** grid to access the Add/Edit Draft EBF Scenario page, where you can create a draft EBF integration scenario.
	For more information about creating draft EBF scenarios, see Adding and Editing Enterprise Business Flow Scenarios.
Scenario Code	Click for a draft requestor integration scenario to access the Draft Requestor Scenario Synopsis page for the integration scenario. Click for a draft provider integration scenario to access the Draft Provider Scenario Synopsis page for the integration scenario. Click for a draft EBF integration scenario to access the Draft EBF Scenario Synopsis page for the integration scenario.
	For more information, see Viewing Draft Integration Scenario Synopses.
Scenario Name	Displays the name of the draft integration scenario.
Description	Displays a description of the integration scenario.
Scenario Summary	Click to access the Integration Scenario Summary page, where you can work with active integration scenarios.
	For more information about working with integration scenarios in active status, see Viewing Integration Scenarios.

Viewing Draft Integration Scenario Synopses

This section discusses how to:

- View a draft requestor integration scenario synopsis.
- View a draft provider integration scenario synopsis.
- View a draft EBF integration scenario synopsis.

Copyright © 2009, Oracle. All rights reserved.

Understanding Scenario Synopsis Pages

The scenario synopsis pages provide wizard-like user interfaces to walk you through the addition and editing of draft integration scenarios. The graphical elements that appear on the scenario synopsis pages represent the steps involved in working with the main components that make up each type of integration scenario.

Click the graphical elements on the page to access a series of pages that enable you to add and edit a draft integration scenario.

The scenario synopsis pages display three types of graphical elements:

- A graphical element that is clickable indicates that the page accessed using the graphical element has been populated with content.
- A graphical element that is not clickable with a Create link next to it indicates that populating content on the page accessed using the Create link is the next step in the process. Click Create to access and populate content on the page.
- A graphical element that is not clickable indicates that no content has been populated on the page and that a previous page must be populated with content before you can access the page.

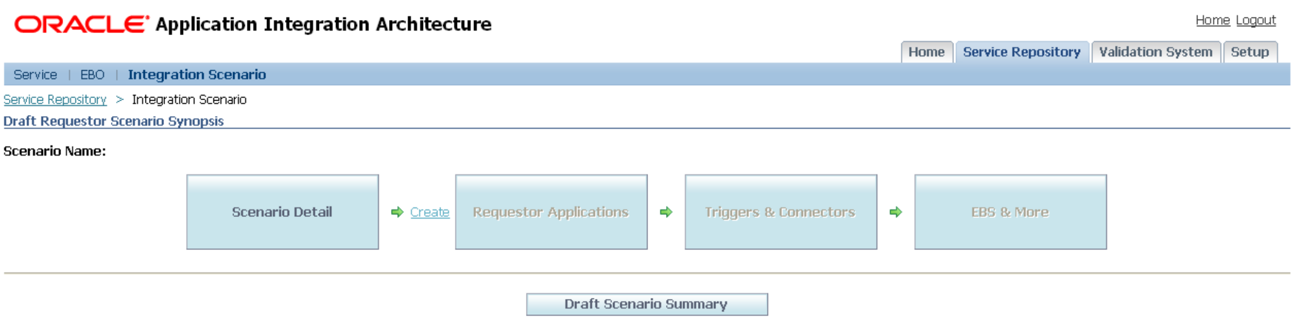
Pages Used to View Draft Integration Scenario Synopses

Page Name	Navigation	Usage
Draft Requestor Scenario Synopsis	Access the Oracle AIA Console. Select the Service Repository tab. Click the Integration Scenario link. Click Draft Scenario Summary. Click a Scenario Code link for a draft requestor integration scenario.	View a synopsis of development progress for a draft requestor integration scenario.
Draft Provider Scenario Synopsis	Access the Oracle AIA Console. Select the Service Repository tab. Click the Integration Scenario link. Click Draft Scenario Summary. Click a Scenario Code link for a draft provider integration scenario.	View a synopsis of development progress for a draft provider integration scenario.
Draft Ebf Scenario Synopsis	Access the Oracle AIA Console. Select the Service Repository tab. Click the Integration Scenario link. Click Draft Scenario Summary. Click a Scenario Code link for a draft EBF scenario.	View a synopsis of development progress for a draft EBF integration scenario.

Viewing a Draft Requestor Scenario Synopsis

Access the Draft Requestor Scenario Synopsis page.

Copyright © 2009, Oracle. All rights reserved.



Draft Requestor Scenario Synopsis page

Scenario Detail	Click to access the Add/Edit Draft Requestor Scenario page.
Requestor Applications	Click to access the Draft Requestor Applications page.
Triggers & Connectors	Click to access the Draft Trigger Events and Connectors page.
EBS & More	Click to access the Draft Enterprise Business Services and More page.

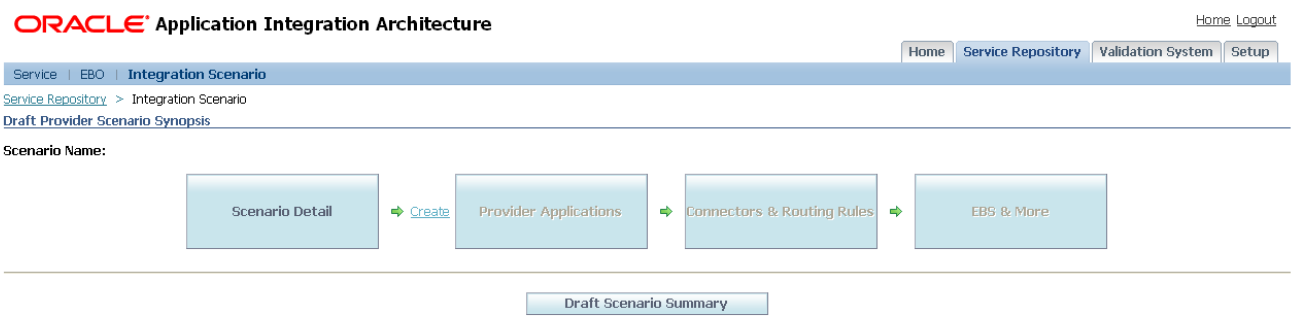
For more information about these pages, see Adding and Editing Requestor Integration Scenarios.

Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.

For more information about the Integration Scenario Draft Summary page, see Searching for Draft Integration Scenarios.

Viewing a Draft Provider Scenario Synopsis

Access the Draft Provider Scenario Synopsis page.


ORACLE Application Integration Architecture
Home Logout
Home
Service Repository
Validation System
Setup
Service | EBO | Integration Scenario
Service Repository > Integration Scenario
Draft Provider Scenario Synopsis
Scenario Name:
Scenario Detail
Create
Provider Applications
Connectors & Routing Rules
EBS & More
Draft Scenario Summary

Draft Provider Scenario Synopsis page

Scenario Detail	Click to access the Add/Edit Draft Provider Scenario page.
Provider Applications	Click to access the Draft Provider Applications page.
Connectors & Routing Rules	Click to access the Draft Provider ABCS and Routing Rules

	page.
EBS & More	Click to access the Draft Enterprise Business Services and More page.

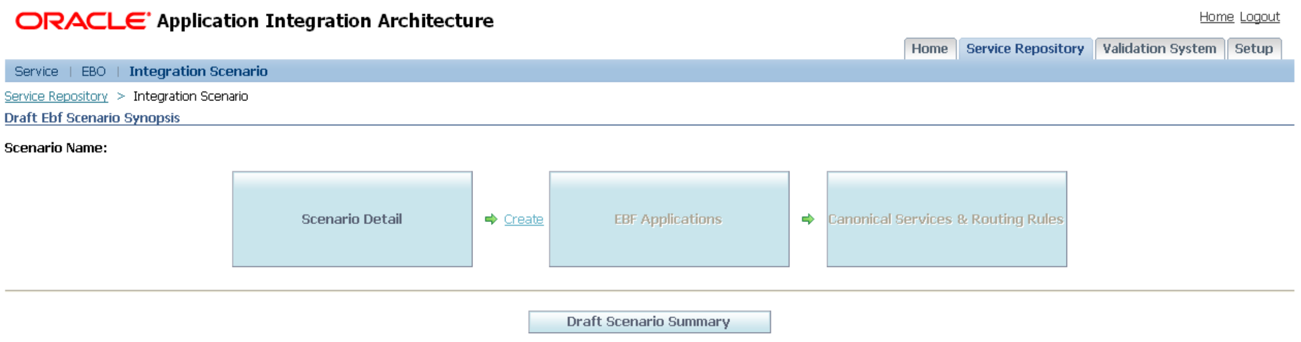
For more information about these pages, see Adding and Editing Provider Integration Scenarios.

Draft Scenario Summary	Click to access the Integration Scenario Draft summary page.

For more information about the Integration Scenario Draft Summary page, see Searching for Draft Integration Scenarios.

Viewing a Draft Ebf Scenario Synopsis

Access the Draft Ebf Scenario Synopsis page.


ORACLE Application Integration Architecture
Home Logout
Home
Service Repository
Validation System
Setup
Service | EBO | Integration Scenario
Service Repository > Integration Scenario
Draft Ebf Scenario Synopsis
Scenario Name:
Scenario Detail
Create
EBF Applications
Canonical Services & Routing Rules
Draft Scenario Summary

Draft Ebf Scenario Synopsis page

Scenario Detail	Click to access the Add/Edit Draft EBF Scenario page.
EBF Applications	Click to access the Draft EBF Applications page.
Canonical Services & Routing Rules	Click to access the Draft EBF Routing Rules and Services page.

For more information about these pages, see Adding and Editing Enterprise Business Flow Scenarios.

Draft Scenario Summary	Click to access the Integration Scenario Draft summary page.

For more information about the Integration Scenario Draft Summary page, see Searching for Draft Integration Scenarios.

Copyright © 2009, Oracle. All rights reserved.

Adding and Editing Requestor Integration Scenarios

A requestor integration scenario outlines the path from the requesting participating application to an EBS.

This section discusses how to:

- Add or edit a draft requestor integration scenario.
- Associate applications with a draft requestor integration scenario.
- Associate trigger events and connectors with a draft requestor integration scenario.
- Associate enterprise business, external, and native services with a draft requestor integration scenario.

Pages Used to Add and Edit Requestor Integration Scenarios

Page Name	Navigation	Usage
Add/Edit Draft Requestor Scenario	• Access the Oracle AIA Console. Select the Service Repository tab. Click the Integration Scenario link. Click Draft Scenario Summary. Click Create in the Draft Requestor Scenarios grid. • Click Scenario Detail on the Draft Requestor Scenario Synopsis page.	Add or edit a draft requestor integration scenario.
BSR Pips	Click the Pip button on the Add/Edit Draft Requestor Scenario page.	Select a PIP that you want to associate with the requestor integration scenario. PIP values available for selection on this page are based on PIP values you have declared on the Setup - BSR Pips page. **For more information** about how to declare PIPs on the BSR Pips page, see Declaring a PIP.
Draft Requestor Applications	• Click Save and Continue on the Add/Edit Draft Requestor Scenario page. • Click Requestor Applications on the Draft Requestor Scenario	Associate applications with the draft requestor integration scenario.

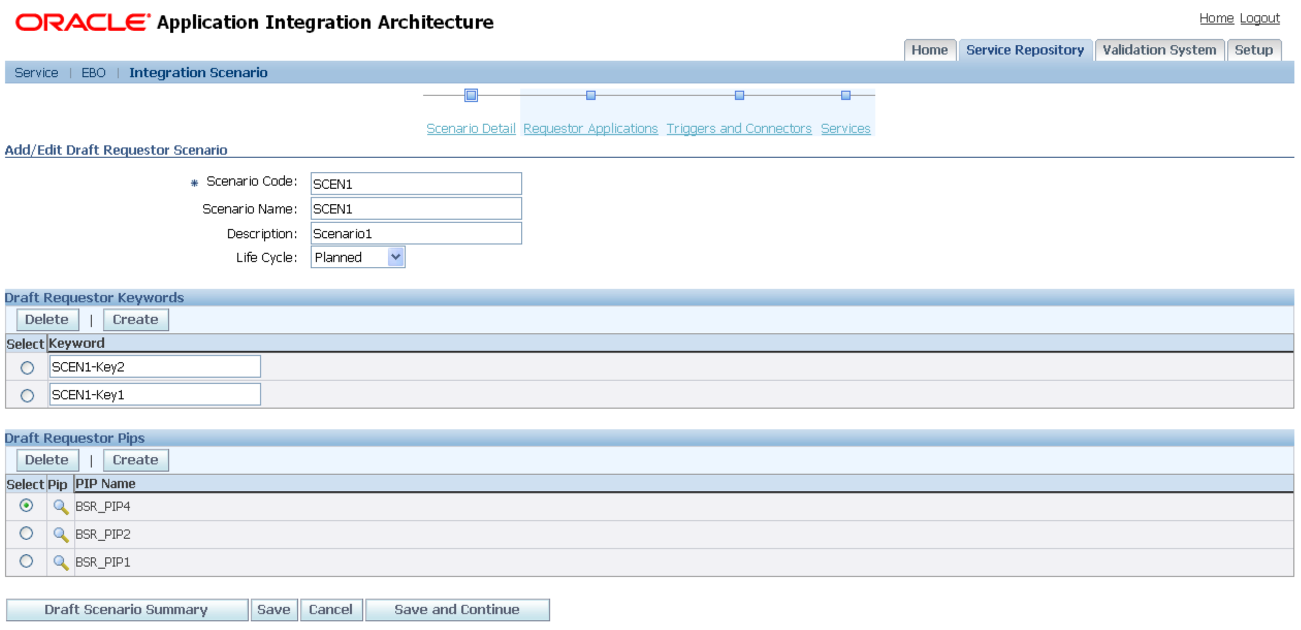
Page Name	Navigation	Usage
	Synopsis page.	
Draft Trigger Events and Connectors	• Click Save and Continue on the Draft Requestor Applications page. • Click Triggers & Connectors on the Draft Requestor Scenario Synopsis page.	Associate triggering events and application business connector (ABC) services with the draft requestor integration scenario. Based on Oracle AIA architecture, the requestor application plugs into the Oracle AIA service-oriented architecture (SOA) layer using the ABC service (connector). This draft page allows you to specify the ABC service and the business events and message that trigger the execution of the ABC service.
Select a Service	Click the Implementation Service Name button on the Draft Trigger Events and Connectors page.	Select the requestor ABC implementation service that you want to associate with the draft requestor integration scenario. If ABC services available for selection on this page do not meet your needs, you can manually enter the desired ABC service name in the Implementation Service Name field on the Draft Trigger Events and Connectors page.
Select an Operation	Click the Implementation Operation Name button on the Draft Trigger Events and Connectors page.	Select the operation on the requestor ABC service that you selected on the Select a Service page that you want to associate with the draft requestor integration scenario. If ABC service operations available for selection on this page do not meet your needs, you can manually enter the desired ABC service operation name in the Implementation Operation Name field on the Draft Trigger Events and Connectors page.
Draft Enterprise Business Services and More	• Click Save and Continue on the Draft Trigger Events and Connectors page. • Click EBOs & More on the Draft Requestor Scenario Synopsis page.	Associate enterprise business, external, and native services with the draft requestor integration scenario.
Draft Requestor Connectors	Click a Connector button on the Draft Enterprise Business Services and	Select the ABC service you selected on the Draft Trigger Events and Connectors page, which is meant to

Copyright © 2009, Oracle. All rights reserved.

Page Name	Navigation	Usage
	More page.	invoke the EBS at runtime. It is possible for a single ABC service to call multiple EBSs. As a result, you may select the same ABC service multiple times on this page by repeatedly clicking Create and choosing the same ABC service.
EBS	Click the EBS button on the Draft Enterprise Business Services and More page.	Select an EBS that you want to associate with the draft requestor integration scenario. This EBS is invoked by the ABC service you selected on the Draft Requestor Connectors page.
External Services	Click the External Service button on the Draft Enterprise Business Services and More page.	Select an external service that you want to associate with the draft requestor integration scenario. This external service is invoked by the ABC service you selected on the Draft Requestor Connectors page.
Native Services	Click the Native Service button on the Draft Enterprise Business Services and More page.	Select a native service that you want to associate with the draft requestor integration scenario. This native service is exposed to the requestor application and is invoked by the ABC service invoked by the ABC service you selected on the Draft Requestor Connectors page.

Adding or Editing a Draft Requestor Integration Scenario

Access the Add/Edit Draft Requestor Scenario page.



Add/Edit Draft Requestor Scenario

* Scenario Code:	SCEN1
Scenario Name:	SCEN1
Description:	Scenario1
Life Cycle:	Planned

Draft Requestor Keywords

Select	Keyword
○	SCEN1-Key2
○	SCEN1-Key1

Draft Requestor Pips

Select	Pip	PIP Name
●	🔍	BSR_PIP4
○	🔍	BSR_PIP2
○	🔍	BSR_PIP1

Add/Edit Draft Requestor Scenario page

Scenario Code	Enter a unique code to identify the requestor integration scenario.
Scenario Name	Enter a name for the requestor integration scenario.
Description	Enter a description of the requestor integration scenario.
Life Cycle	Select a life cycle status for the integration scenario. This is value is not the development status of the integration scenario. It does not affect whether the integration scenario is active and displays on the Integration Scenario Summary page or is a draft and displays using the Integration Scenario Draft Summary page. ***Active*** ***Deprecated*** ***Obsolete*** ***Planned***

Draft Requestor Keywords

Use the **Draft Requestor Keywords** grid to manage keywords for the requestor integration scenario. These keywords can be used as search criteria when searching for requestor integration scenarios using the Integration Scenario Summary page.

Delete	Displays when keywords are present. Select the **Select** option for a keyword and click **Delete** to delete the keyword.
Create	Click to add a keyword.

Copyright © 2009, Oracle. All rights reserved.

Keyword	Enter the keyword associated with the requestor integration scenario.

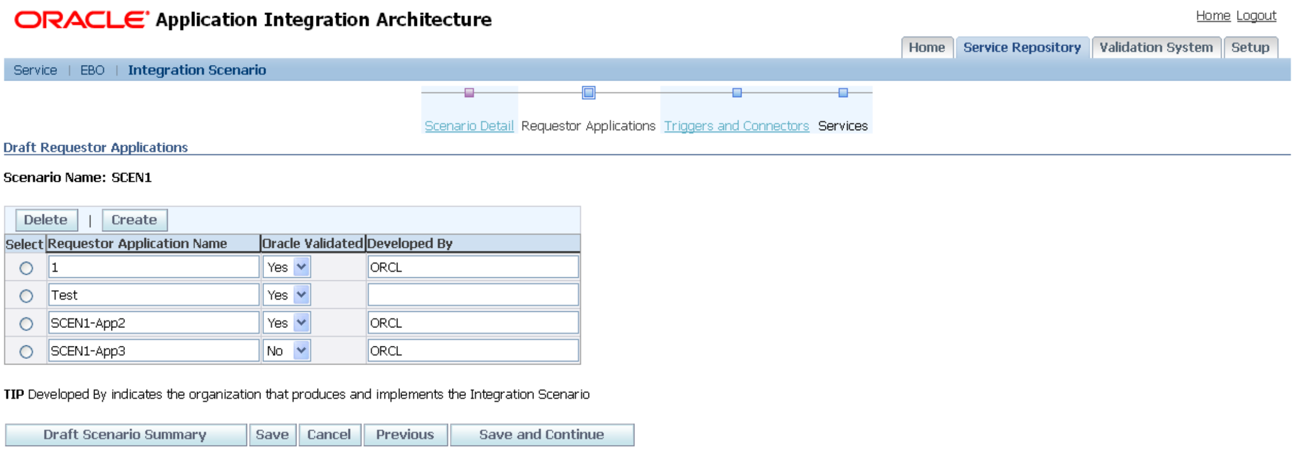
Draft Requestor Pips

Use the **Draft Requestor Pips** grid to manage PIPs associated with the requestor integration scenario.

Delete	Displays when PIPs have been associated with the requestor integration scenario.
Create	Click to associate a PIP with the requestor integration scenario and display the PIP button.
Pip	Click to access the BSR Pips page, where you can select a PIP that you want to associate with the requestor integration scenario. Available PIP values are derived from PIPs declared on the BSR PIPs page. **For more information** about declaring PIPs in the BSR, see Declaring a PIP in the BSR.
PIP Name	Displays the name of a PIP associated with the requestor integration scenario.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft Requestor Scenario Synopsis page.
Save and Continue	Click to save edits to the page and access the Draft Requestor Applications page, where you can associate applications with the requestor integration scenario.

Associating Applications with a Draft Requestor Integration Scenario

Access the Draft Requestor Applications page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Service | EBO | Integration Scenario
Scenario Detail | Requestor Applications | Triggers and Connectors | Services
Draft Requestor Applications
Delete | Create
Draft Scenario Summary | Save | Cancel | Previous | Save and Continue

Scenario Name: SCEN1

Select	Requestor Application Name	Oracle Validated	Developed By
○	1	Yes	ORCL
○	Test	Yes	
○	SCEN1-App2	Yes	ORCL
○	SCEN1-App3	No	ORCL

TIP Developed By indicates the organization that produces and implements the Integration Scenario

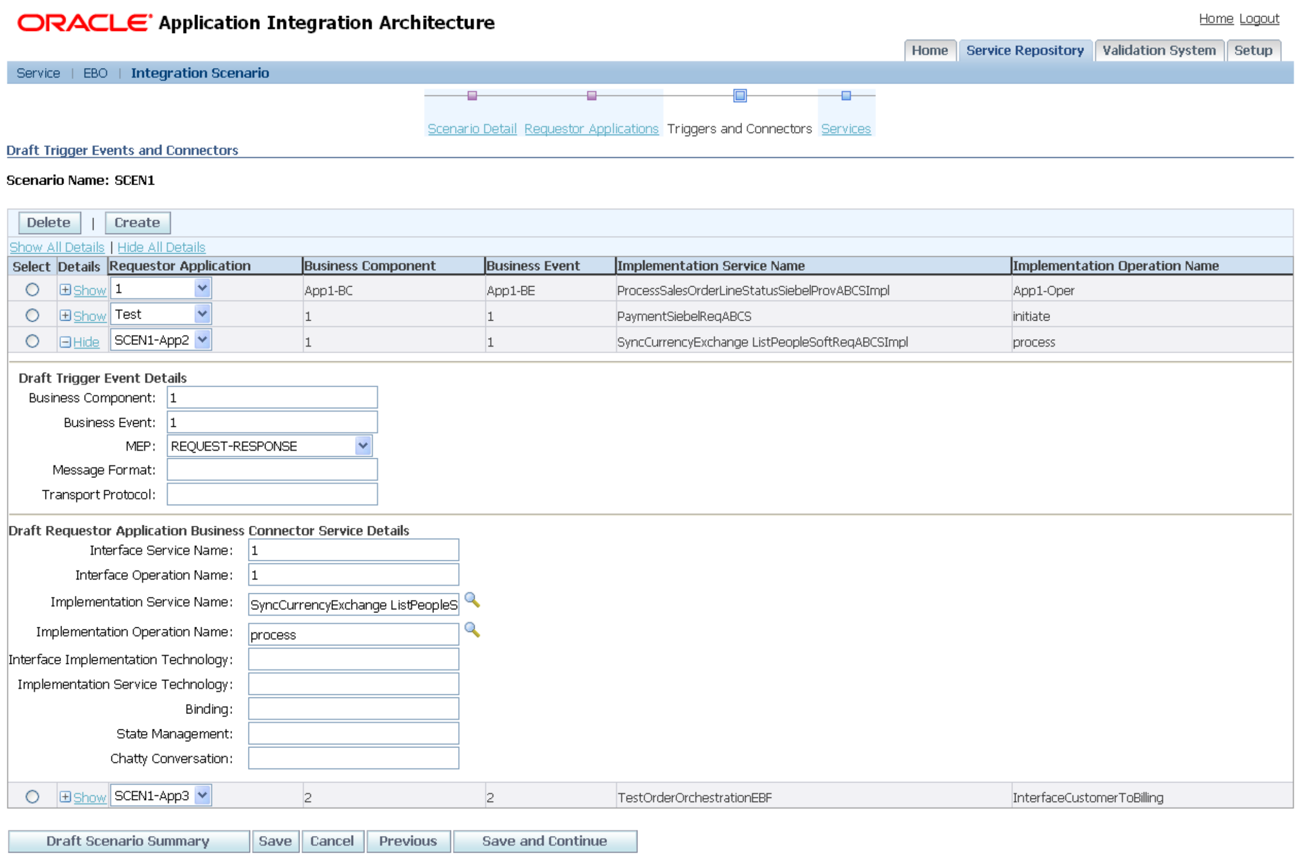
Draft Requestor Applications page

Delete	Displays when application values are present. Select the **Select** option for an application and click **Delete** to delete the association between the application and the requestor integration scenario.
Create	Click to create an association between an application and the requestor integration scenario. This is a requestor application that originates the requestor integration scenario. Click **Create** repeatedly to associate multiple requestor applications.
Requestor Application Name	Enter the application name.
Oracle Validated	Enter a value that indicates whether the requestor integration scenario was built following a formal Oracle AIA methodology program.
Developed By	Enter the name of the vendor or system integrator that built the integration scenario.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft Requestor Scenario Synopsis page.
Previous	Click to cancel any unsaved edits to the page and return to the Add/Edit Draft Requestor Scenario page.
Save and Continue	Click to save edits to the page and access the Draft Trigger Events and Connectors page, where you can define triggering events and ABC services for the requestor integration scenario.

Copyright © 2009, Oracle. All rights reserved.

Associating Trigger Events and Connectors with a Draft Requestor Integration Scenario

Access the Draft Trigger Events and Connectors page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Service | EBO | Integration Scenario
Scenario Detail | Requestor Applications | Triggers and Connectors | Services
Delete | Create
Show All Details | Hide All Details
Draft Scenario Summary | Save | Cancel | Previous | Save and Continue

Draft Trigger Events and Connectors

Scenario Name: SCEN1

Select	Details	Requestor Application	Business Component	Business Event	Implementation Service Name	Implementation Operation Name
○	Show	1	App1-BC	App1-BE	ProcessSalesOrderLineStatusSiebelProvABCSImpl	App1-Oper
○	Show	Test	1	1	PaymentSiebelReqABCS	initiate
○	Hide	SCEN1-App2	1	1	SyncCurrencyExchange ListPeopleSoftReqABCSImpl	process
○	Show	SCEN1-App3	2	2	TestOrderOrchestrationEBF	InterfaceCustomerToBilling

Draft Trigger Event Details

Business Component:	1
Business Event:	1
MEP:	REQUEST-RESPONSE
Message Format:	
Transport Protocol:	

Draft Requestor Application Business Connector Service Details

Interface Service Name:	1
Interface Operation Name:	1
Implementation Service Name:	SyncCurrencyExchange ListPeopleS
Implementation Operation Name:	process
Interface Implementation Technology:	
Implementation Service Technology:	
Binding:	
State Management:	
Chatty Conversation:	

Draft Trigger Events and Connectors page

Delete	Displays when trigger event and connector values are present. Select the **Select** option for a row and click **Delete** to delete the association between the trigger event and connector and the requestor integration scenario.
Create	Click to create an association between a trigger event and connector and the requestor integration scenario.
Show All Details and Hide All Details	Displays when details are present. Click to show and hide details for all rows.
Details	Click **Show** to enter and view details for the row. Details are displayed in the **Draft Trigger Event Details** and **Draft Requestor Application Business Connector Service Details** areas. Click **Hide** to hide details for the row.

Draft Trigger Event Details

Use the **Draft Trigger Event Details** area to enter details about the event that triggers the execution of the requestor integration scenario. These are logical events that do not necessarily correspond to business events in the participating applications.

Business Component	Enter the name of the business component associated with the triggering event. The business component is the module in the requestor participating application that triggers the requestor scenario.
Business Event	Enter the name of the event in the business component that emits the triggering event.
MEP (message exchange pattern)	Select the MEP used by the triggering event. ***REQUEST RESPONSE*** ***FIRE-AND-FORGET*** ***REQUEST-DELAYED RESPONSE***
Message Format	Enter the message format used by the triggering event. For example, ***SOAP*** or ***XML.***
Transport Protocol	Enter the transport protocol used by the triggering event. For example, ***HTTP, JMS,*** or ***AQ.***

Draft Application Business Connector Service Details

Use the **Draft Application Business Connector Service Details** area to enter details about the ABC service you want to associate with the requestor integration scenario. This is the ABC service that connects the requestor participating application with the Oracle AIA canonical layers, including EBSs.

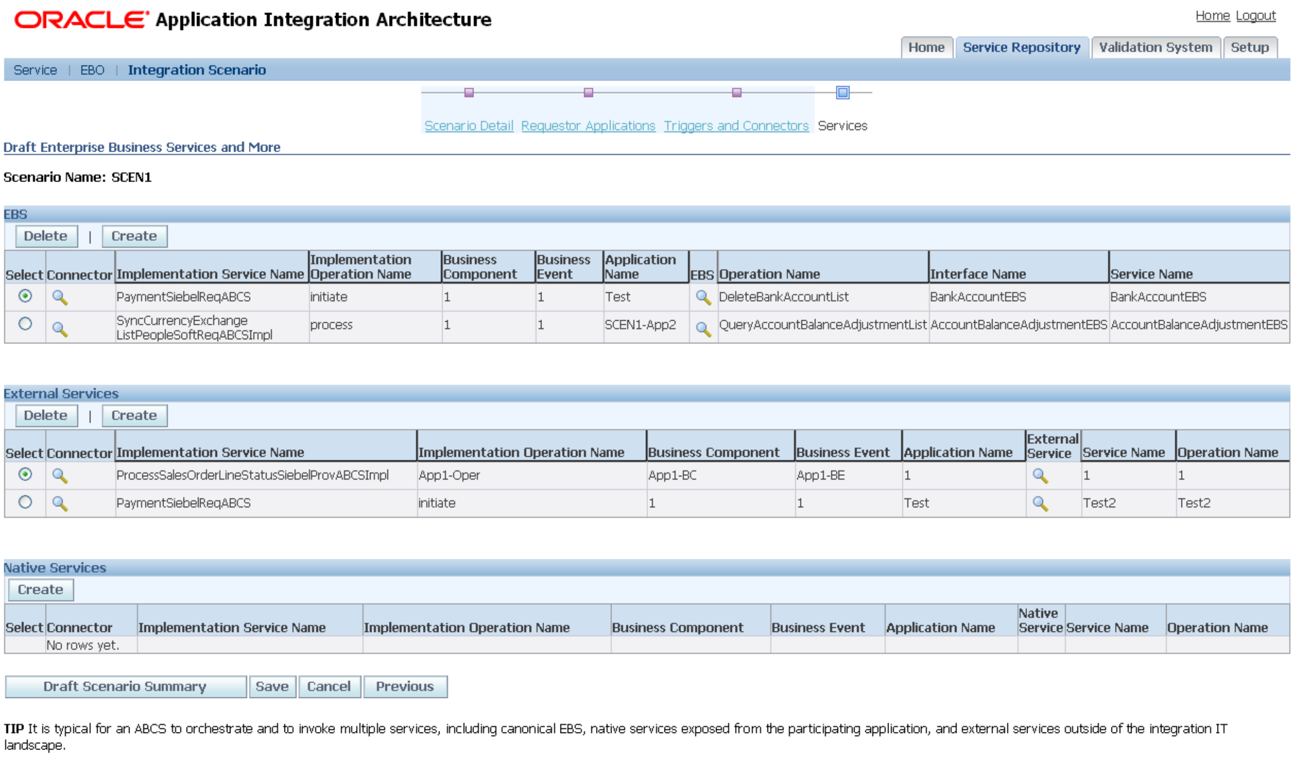
Interface Service Name	Enter the name of the ABC interface service used by the requestor ABC implementation service.
Interface Operation Name	Enter the operation used by the ABC interface service.
Implementation Service Name	Click to access the Select a Service page, where you can select the requestor ABC implementation service that you want to associate with the requestor integration scenario.
Implementation Operation Name	Click to access the Select an Operation page, where you can select the operation used by the implementation service associated with the requestor integration scenario.
Interface Implementation Technology	Enter the technology used to implement the ABC service interface.
Implementation Service Technology	Enter the technology used to implement the requestor ABC implementation service. For example, ***BPEL*** or ***ESB.***
Binding	Enter the type of binding used by the requestor ABC implementation service.
State Management	Indicate whether the requestor ABC service is managing the state while having outbound interactions with the requestor

Copyright © 2009, Oracle. All rights reserved.

	application. For example, ***Yes*** or ***No.***
Chatty Conversation	Indicate whether the requestor ABC implementation service participates in a chatty conversation. For example, ***Yes*** or ***No.***
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft Requestor Scenario Synopsis page.
Previous	Click to cancel any unsaved edits to the page and return to the Draft Requestor Applications page.
Save and Continue	Click to save edits to the page and access the Draft Enterprise Business Services and More page, where you can associate enterprise business, external, and native services with the requestor integration scenario.

Associating Services with a Draft Requestor Integration Scenario

Access the Draft Enterprise Business Services and More page.



Draft Enterprise Business Services and More

Scenario Name: SCEN1

EBS

Delete | Create

Select	Connector	Implementation Service Name	Implementation Operation Name	Business Component	Business Event	Application Name	EBS	Operation Name	Interface Name	Service Name
●	🔍	PaymentSiebelReqABCS	initiate	1	1	Test	🔍	DeleteBankAccountList	BankAccountEBS	BankAccountEBS
○	🔍	SyncCurrencyExchange ListPeopleSoftReqABCSImpl	process	1	1	SCEN1-App2	🔍	QueryAccountBalanceAdjustmentList	AccountBalanceAdjustmentEBS	AccountBalanceAdjustmentEBS

External Services

Delete | Create

Select	Connector	Implementation Service Name	Implementation Operation Name	Business Component	Business Event	Application Name	External Service	Service Name	Operation Name
●	🔍	ProcessSalesOrderLineStatusSiebelProvABCSImpl	App1-Oper	App1-BC	App1-BE	1	🔍	1	1
○	🔍	PaymentSiebelReqABCS	initiate	1	1	Test	🔍	Test2	Test2

Native Services

Create

Select	Connector	Implementation Service Name	Implementation Operation Name	Business Component	Business Event	Application Name	Native Service	Service Name	Operation Name
	No rows yet.								

Draft Scenario Summary | Save | Cancel | Previous

TIP It is typical for an ABCS to orchestrate and to invoke multiple services, including canonical EBS, native services exposed from the participating application, and external services outside of the integration IT landscape.

Draft Enterprise Business Services and More page

EBS, External Services, and Native Services

Use the **EBS**, **External Services**, and **Native Services** grids to associate EBSs, external services, and native services with the requestor integration scenario. In its work to bridge requestor participating applications with the Oracle AIA canonical layer, including EBSs and EBFs), the ABC service may need to call multiple EBSs, native services exposed by requestor participating applications, and external services.

Delete	Displays when service values are present. Select the **Select** option for a row and click **Delete** to delete the association between the service and the requestor integration scenario.
Create	Click to create an association between a service and the requestor integration scenario.
Connector	Click to access the Draft Requestor Connectors page, where you can select an ABC service that you associated with the requestor integration scenario on the Draft Trigger Events and Connectors page. Once selected, the page displays information about the connector and enables you to associate services with the requestor integration scenario.
EBS (enterprise business service)	Click to access the EBS page, where you can select an EBS or EBF that you want to associate with the requestor integration scenario. This is the EBS or EBF that the requestor ABC service will call. Available values are based on EBSs and EBFs that are present in the BSR. If your desired EBS or EBF is not yet present in the BSR, you can add an association by clicking Create on the EBS page and entering values. This association can serve as a requirement for addition of the EBS or EBF to the BSR.
External Service	Click to access the External Services page, where you can select an external service that you want to associate with the requestor integration scenario. Available external service values are based on external services that are called by the ABC service. If your desired external service is not yet present in the BSR, you can add an association by clicking **Create** on the External Services page and entering values. This association can serve as a requirement for addition of the external service to the BSR.
Native Service	Click to access the Native Services page, where you can select a native service that you want to associate with the requestor integration scenario. Native services are exposed by the requestor participating application. ABC services can make callbacks to the requestor participating application using these native services. Available native service values are based on native services that are called by the ABC service. You can add an association to a native service to the ABC service by clicking

Copyright © 2009, Oracle. All rights reserved.

	Create on the Native Services page and entering values.
Operation Name	Displays once a service has been selected. Displays the operation of the selected service that is invoked by the requestor ABC implementation service.
Interface Name	Displays in the **EBS** grid once an EBS has been selected. Displays the port type to which the selected EBS operation belongs.
Service Name	Displays once a service has been selected. Displays the name of the service.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft Requestor Scenario Synopsis page.
Previous	Click to cancel any unsaved edits to the page and return to the Draft Requestor Applications page.

Adding and Editing Provider Integration Scenarios

A draft provider integration scenario outlines the path from an EBS to a target providing participating application.

This section discusses how to:

- Add or edit a draft provider integration scenario.
- Associate applications with a draft provider integration scenario.
- Associate connectors and routing rules with a draft provider integration scenario.
- Associate enterprise business, external, and native services with a draft provider integration scenario.

Pages Used to Add and Edit Provider Integration Scenarios

Page Name	Navigation	Usage
Add/Edit Draft Provider Scenario	• Access the Oracle AIA Console. Select the Service Repository tab. Click the Integration Scenario link. Click Draft Scenario Summary. Click Create in the Draft Provider Scenarios grid. • Click Scenario Detail on the Draft Provider Scenario Synopsis page.	Add or edit a draft provider integration scenario.

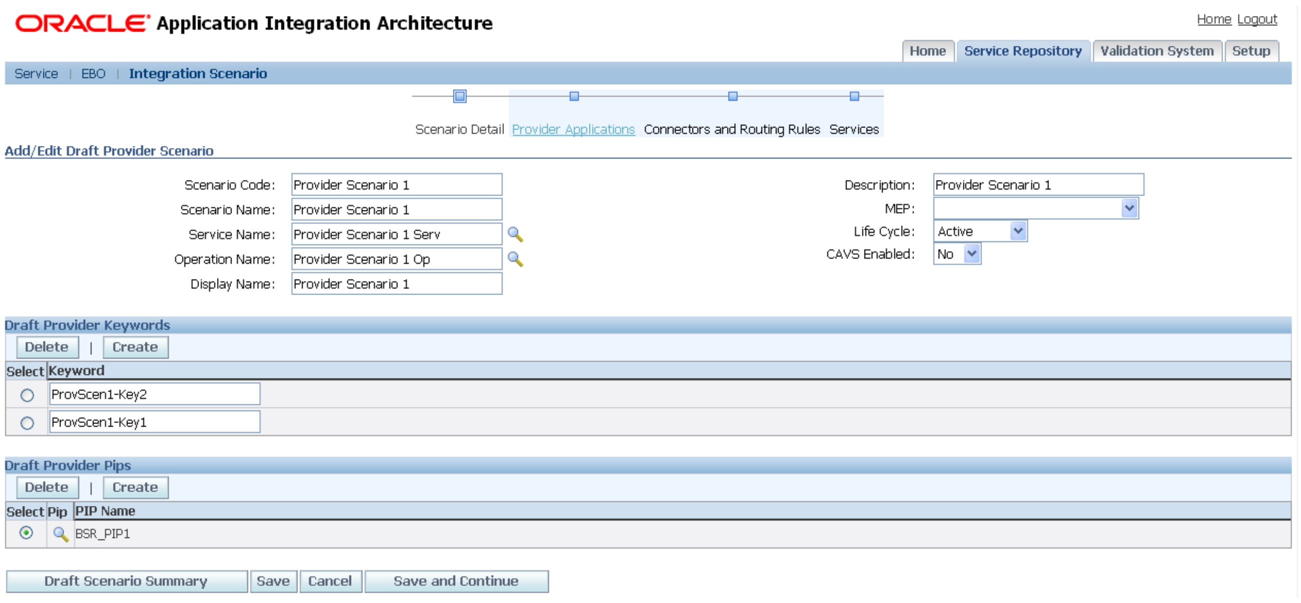
Page Name	Navigation	Usage
Select a Service	Click the Service Name button on the Add/Edit Draft Provider Scenario page.	Select the EBS that you want to associate with the draft provider integration scenario. This is the EBS that serves as an entry point that invokes the provider scenario.
Select an Operation	Click the Operation Name button on the Add/Edit Draft Provider Scenario page.	Select the operation you want to associate with the draft provider integration scenario.
BSR Pips	Click the Pip button on the Add/Edit Draft Provider Scenario page.	Select a PIP that you want to associate with the provider integration scenario. PIP values available for selection on this page are based on PIP values you have declared on the Setup - BSR Pips page. **For more information** about how to declare PIPs on the BSR Pips page, see Declaring a PIP.
Draft Provider Applications	• Click Save and Continue on the Add/Edit Draft Provider Scenario page. • Click Provider Applications on the Draft Provider Scenario Synopsis page.	Associate applications with the draft provider integration scenario.
Draft Provider ABCS and Routing Rules	• Click Save and Continue on the Draft Provider Applications page. • Click Connectors & Routing Rules on the Draft Provider Scenario Synopsis page.	Associate application business connector (ABC) services and routing rules with the draft provider integration scenario. This defines which provider ABC services can be routed to from the EBS.
Select a Service	Click the Implementation Service Name button on the Draft Provider ABCS and Routing Rules page.	Select the provider ABC implementation service that you want to associate with the draft provider integration scenario.
Select an Operation	Click the Implementation Operation Name button on the Draft Provider ABCS and Routing Rules page.	Select the operation you want to associate with the draft provider integration scenario.
Draft Enterprise Business Services and More	• Click Save and Continue on the Draft Provider ABCS and Routing Rules page. • Click EBS & More on the Draft	Associate enterprise business, external, and native services with the draft provider integration scenario.

Copyright © 2009, Oracle. All rights reserved.

Page Name	Navigation	Usage
	Provider Scenario Synopsis page.	
Draft Provider Connectors	Click the Connector button on the Draft Enterprise Business Services and More page.	Select a provider connector (ABC service) that you have associated with the draft provider integration scenario for which you want to provide service details.
EBS	Click the EBS button on the Draft Enterprise Business Services and More page.	Select an EBS that you want to associate with the draft provider integration scenario.
External Services	Click the External Service button on the Draft Enterprise Business Services and More page.	Select an external service that you want to associate with the draft provider integration scenario.
Native Services	Click the Native Service button on the Draft Enterprise Business Services and More page.	Select a native service that you want to associate with the draft provider integration scenario.

Adding or Editing a Draft Provider Integration Scenario

Access the Add/Edit Draft Provider Scenario page.



Add/Edit Draft Provider Scenario

Scenario Code:	Provider Scenario 1
Scenario Name:	Provider Scenario 1
Service Name:	Provider Scenario 1 Serv
Operation Name:	Provider Scenario 1 Op
Display Name:	Provider Scenario 1
Description:	Provider Scenario 1
MEP:	
Life Cycle:	Active
CAVS Enabled:	No

Draft Provider Keywords

Select	Keyword
○	ProvScen1-Key2
○	ProvScen1-Key1

Draft Provider Pips

Select	Pip	PIP Name
●		BSR_PIP1

Add/Edit Draft Provider Scenario page

Scenario Code	Enter a unique code to identify the provider integration scenario.
Scenario Name	Enter a name for the provider integration scenario.
Service Name	Click to access the Select a Service page, where you can select the EBS or EBF that you want to associate with the draft provider integration scenario. This is the EBS that will

	invoke the provider ABC service.
Operation Name	Click to access the Select an Operation page, where you can select the operation of the service you chose to associate with the draft provider integration scenario. This is the operation that will invoke the provide ABC service.
Display Name	Enter the user-friendly name of the provider scenario. For instance, get Account Balance.
Description	Enter a description of the provider integration scenario.
MEP (message exchange pattern)	Select the MEP used by the selected EBS operation. ***REQUEST RESPONSE*** ***FIRE-AND-FORGET*** ***REQUEST-DELAYED RESPONSE***
Life Cycle	Select a life cycle status for the integration scenario. This is value is not the development status of the integration scenario. It does not affect whether the integration scenario is active and displays on the Integration Scenario Summary page or is a draft and displays using the Integration Scenario Draft Summary page. ***Active*** ***Deprecated*** ***Obsolete*** ***Planned***
CAVS Enabled	Indicates whether the integration scenario has been developed to work with testing functionality provided by the Composite Application Validation System (CAVS). **For more information** about enabling CAVS, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Developing CAVS-Enabled Oracle AIA Services."

Draft Provider Keywords

Use the **Draft Provider Keywords** grid to manage keywords for the provider integration scenario. These keywords can be used as search criteria when searching for provider integration scenarios using the Integration Scenario Summary page.

Delete	Displays when keywords are present. Select the **Select** option for a keyword and click **Delete** to delete the keyword.
Create	Click to add a keyword.

Copyright © 2009, Oracle. All rights reserved.

Keyword	Enter the keyword associated with the provider integration scenario.

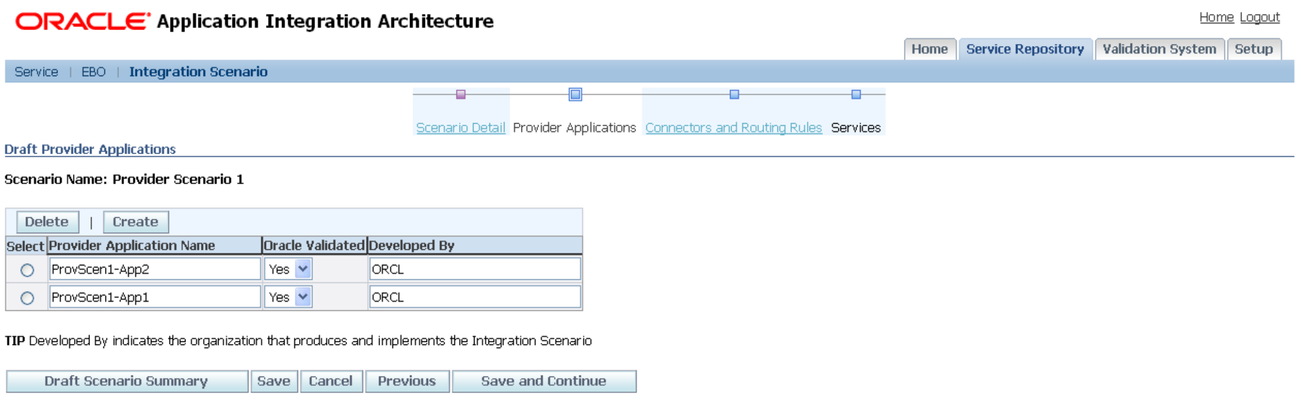
Draft Provider Pips

Use the **Draft Provider Pips** grid to manage PIPs associated with the provider integration scenario.

Delete	Displays when PIPs have been associated with the provider integration scenario.
Create	Click to associate a PIP with the provider integration scenario and display the PIP button.
Pip	Click to access the BSR Pips page, where you can select a PIP that you want to associate with the provider integration scenario. Available PIP values are derived from PIPs declared on the BSR PIPs page. **For more information** about declaring PIPs in the BSR, see Declaring a PIP in the BSR.
PIP Name	Displays the name of a PIP associated with the provider integration scenario.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft Provider Scenario Synopsis page.
Save and Continue	Click to save edits to the page and access the Draft Provider Applications page, where you can associate applications with the provider integration scenario.

Associating Applications with a Draft Provider Integration Scenario

Access the Draft Provider Application page.



Scenario Name: Provider Scenario 1

Select	Provider Application Name	Oracle Validated	Developed By
○	ProvScen1-App2	Yes	ORCL
○	ProvScen1-App1	Yes	ORCL

TIP Developed By indicates the organization that produces and implements the Integration Scenario

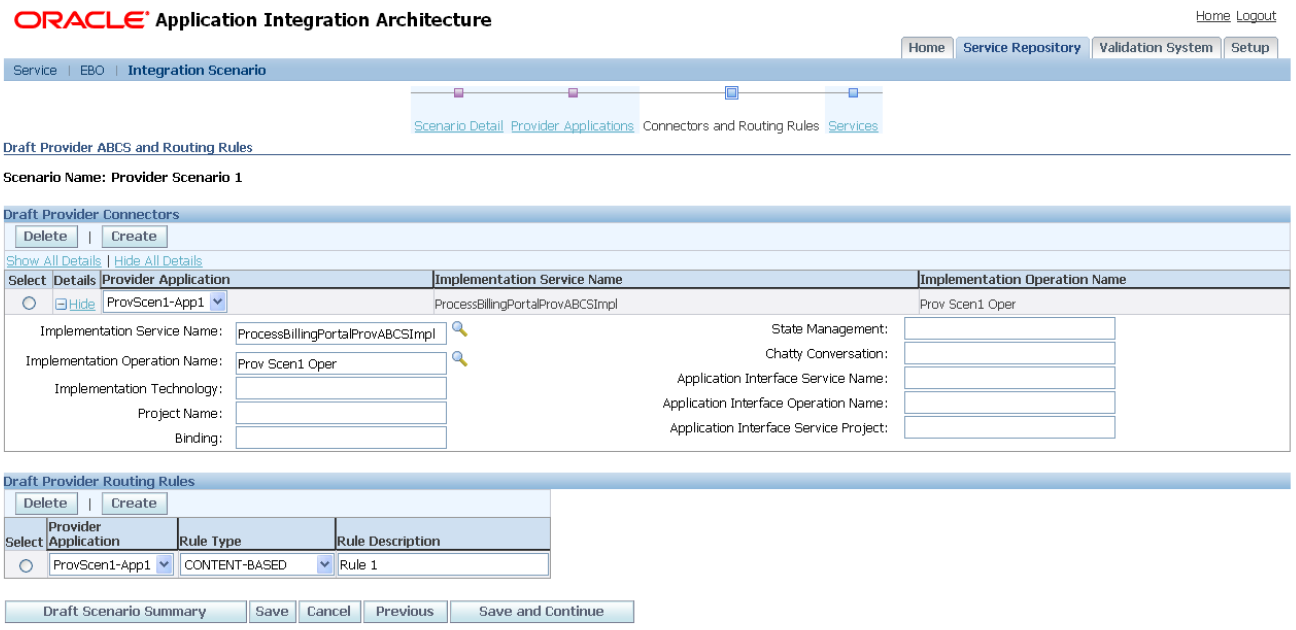
Draft Provider Applications page

Delete	Displays when application values are present. Select the **Select** option for an application and click **Delete** to delete the association between the application and the provider integration scenario.
Create	Click to create an association between an application and the provider integration scenario.
Provider Application Name	Enter the application name.
Oracle Validated	Enter a value that indicates whether the provider integration scenario was built following a formal Oracle Application Integration Architecture methodology program.
Developed By	Enter the name of the vendor or system integrator that built the integration scenario.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft Provider Scenario Synopsis page.
Previous	Click to cancel any unsaved edits to the page and return to the Add/Edit Draft Provider Scenario page.
Save and Continue	Click to save edits to the page and access the Draft Provider ABCS and Routing Rules page, where you can associate routing rules and ABC services with the provider integration scenario.

Associating Connectors and Routing Rules with a Draft Provider Integration Scenario

Access the Draft Provider ABCS and Routing Rules page.

Copyright © 2009, Oracle. All rights reserved.


ORACLE Application Integration Architecture
Home Logout
Home
Service Repository
Validation System
Setup
Service | EBO | Integration Scenario
Scenario Detail
Provider Applications
Connectors and Routing Rules
Services

Draft Provider ABCS and Routing Rules

Scenario Name: Provider Scenario 1

Draft Provider Connectors

Delete | Create

Show All Details | Hide All Details

Select	Details	Provider Application	Implementation Service Name	Implementation Operation Name
○	⊟ Hide	ProvScen1-App1	ProcessBillingPortalProvABCSImpl	Prov Scen1 Oper

Implementation Service Name:	ProcessBillingPortalProvABCSImpl	State Management:	
Implementation Operation Name:	Prov Scen1 Oper	Chatty Conversation:	
Implementation Technology:		Application Interface Service Name:	
Project Name:		Application Interface Operation Name:	
Binding:		Application Interface Service Project:	

Draft Provider Routing Rules

Delete | Create

Select	Provider Application	Rule Type	Rule Description
○	ProvScen1-App1	CONTENT-BASED	Rule 1

Draft Scenario Summary | Save | Cancel | Previous | Save and Continue

Draft Provider ABCS and Routing Rules page

Draft Provider Connectors

Use the **Draft Provider Connectors** area to enter details about the provider ABC services you want to associate with the provider integration scenario.

Delete	Displays when connector values are present. Select the **Select** option for a row and click **Delete** to delete the association between the ABC service and the provider integration scenario.
Create	Click to create an association between an ABC service and the provider integration scenario.
Show All Details and Hide All Details	Displays when details are present. Click to show and hide details for all rows.
Details	Click **Show** to enter and view details for the row. Details are displayed in the **Draft Trigger Event Details** and **Draft Connector Details** areas. Click **Hide** to hide details for the row.
Implementation Service Name	Click to access the Select a Service page, where you can select the provider ABC implementation service that you want to associate with the provider integration scenario.
Implementation Operation Name	Click to access the Select an Operation page, where you can select the operation invoked by the ABC implementation service you associated with the provider integration scenario.
Implementation Technology	Enter the technology used to implement the provider ABC implementation service. For example, ***BPEL*** or ***ESB.***
Project Name	Enter the name of the ESB or BPEL project that houses the

	provider ABC service.
Binding	Enter the type of binding used by the provider ABC implementation service.
State Management	Indicate whether the provider ABC implementation service is managing the state while having outbound interactions with the provider application. For example, ***Yes*** or ***No.***
Chatty Conversation	Indicate whether the provider ABC implementation service participates in a chatty conversation. For example, ***Yes*** or ***No.***
Application Interface Service Name	Enter the name of the ABC interface service used by the provider ABC implementation service.
Application Interface Operation Name	Enter the operation used by the ABC interface service.

Draft Provider Routing Rules

Use the **Draft Provider Routing Rules** area to enter details about the routing rules you want to associate with the provider integration scenario. These are routing rules that are implemented on the EBSs and operations you selected on the Add/Edit Draft Provider Scenario page. The routing rules route runtime messages to respective provider ABC services. Therefore, there is a one-to-one correspondence between routing rules and provider ABC services.

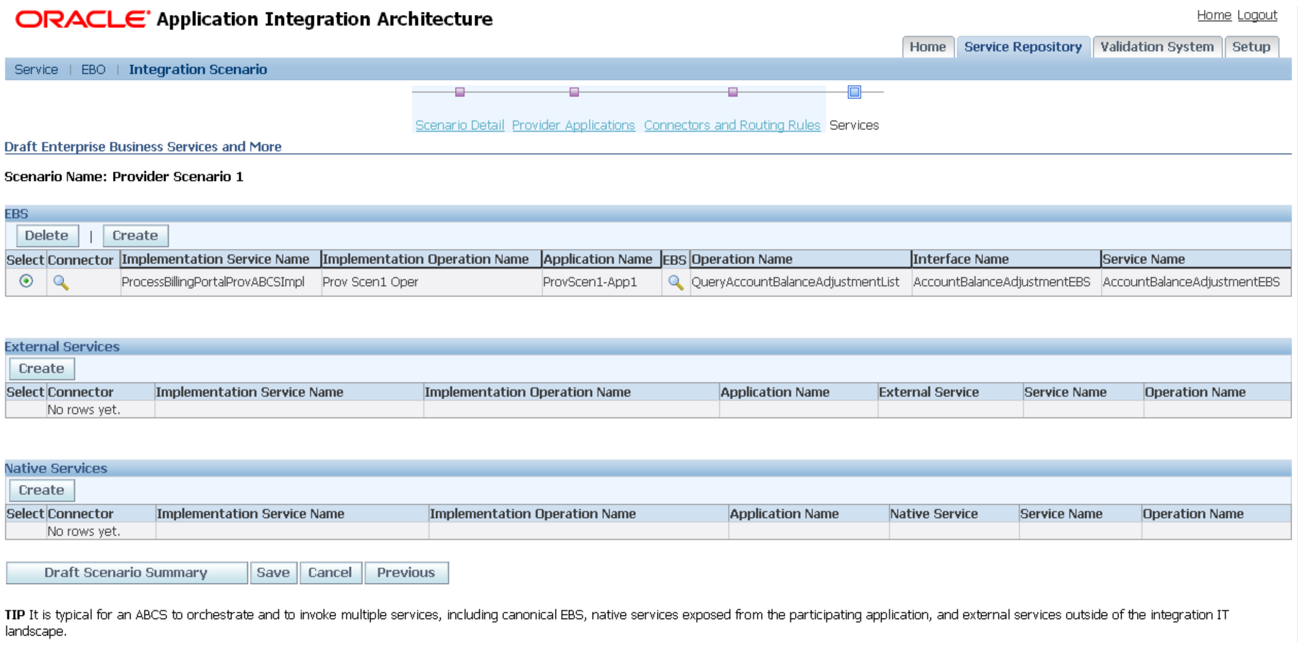
Delete	Displays when routing rule values are present. Select the **Select** option for a row and click **Delete** to delete the association between the routing rule and the provider integration scenario.
Create	Click to create an association between a routing rule and the provider integration scenario.
Provider Application	Select the provider application to which the routing rule will route the message. Available values are derived from the values you entered on the Draft Provider Applications page.
Rule Type	Select a rule type. ***CONTENT-BASED:*** Indicates that content-based routing rules are implemented on the EBS, resulting in runtime message being routed to appropriate provider ABC services. ***ALWAYS:*** Indicates that regardless of runtime message content, the message will always be routed to a predetermined destination. ***TEST-HARNESS:*** Indicates that when CAVS is enabled, the message will be routed to the CAVS.
Rule Description	Enter a description of the routing rule that allows runtime messages to be routed to a destination. This indicates that there is a routing rule associated with the EBS, which results in the appropriate routing of messages.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.

Copyright © 2009, Oracle. All rights reserved.

Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft Provider Scenario Synopsis page.
Previous	Click to cancel any unsaved edits to the page and return to the Draft Provider Applications page.
Save and Continue	Click to save edits to the page and access the Draft Enterprise Business Services and More page, where you can associate enterprise business services, external services, and native services with the provider integration scenario.

Associating Services with a Draft Provider Integration Scenario

Access the Draft Enterprise Business Services and More page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Service | EBO | Integration Scenario
Scenario Detail Provider Applications Connectors and Routing Rules Services

Draft Enterprise Business Services and More

Scenario Name: Provider Scenario 1

EBS

Delete | Create

Select	Connector	Implementation Service Name	Implementation Operation Name	Application Name	EBS	Operation Name	Interface Name	Service Name
●	🔍	ProcessBillingPortalProvABCSImpl	Prov Scen1 Oper	ProvScen1-App1	🔍	QueryAccountBalanceAdjustmentList	AccountBalanceAdjustmentEBS	AccountBalanceAdjustmentEBS

External Services

Create

Select	Connector	Implementation Service Name	Implementation Operation Name	Application Name	External Service	Service Name	Operation Name
	No rows yet.						

Native Services

Create

Select	Connector	Implementation Service Name	Implementation Operation Name	Application Name	Native Service	Service Name	Operation Name
	No rows yet.						

Draft Scenario Summary | Save | Cancel | Previous

TIP It is typical for an ABCS to orchestrate and to invoke multiple services, including canonical EBS, native services exposed from the participating application, and external services outside of the integration IT landscape.

Draft Enterprise Business Services and More page

EBS, External Services, and Native Services

Use the **EBS**, **External Services**, and **Native Services** grids to associate EBSs, external services, and native services with the provider integration scenario. In its work to bridge provider participating applications with the Oracle AIA canonical layer, including EBSs and EBFs), the ABC service may need to call multiple EBSs, native services exposed by provider participating applications, and external services.

Delete	Displays when service values are present. Select the **Select** option for a row and click **Delete** to delete the association between the service and the provider integration scenario.
Create	Click to create an association between a service and the provider integration scenario.

Connector	Click to access the Draft Provider Connectors page, where you can select a provider ABC service that connects the provider participating application to Oracle AIA canonical services. You selected this provider ABC service on the Draft Provider ABCS and Routing Rules page. Once selected, information about the ABC service displays on the page.
EBS (enterprise business service)	Click to access the EBS page, where you can select an EBS that you want to associate with the provider integration scenario. This is the EBS that calls the provider ABC service, based on routing rules. Available values are based on EBSs that are present in the BSR. If your desired EBS is not yet present in the BSR, you can add an association by clicking **Create** on the EBS page and entering values. This association can serve as a requirement for addition of the EBS to the BSR.
External Service	Click to access the External Services page, where you can select an external service that you want to associate with the provider integration scenario. Available external service values are based on external services that are invoked by the provider ABC service. If your desired external service is not yet present in the BSR, you can add an association by clicking **Create** on the External Services page and entering values. This association can serve as a requirement for addition of the external service to the BSR.
Native Service	Click to access the Native Services page, where you can select a native service that you want to associate with the provider integration scenario. Native services are exposed by the provider participating application. Provider ABC services can make calls to the provider participating application using these native services. Available native service values are based on native services that are called by the ABC service. You can add an association to a native service to the ABC service by clicking **Create** on the Native Services page and entering values.
Operation Name	Displays once a service has been selected. Displays the operations available for the selected service.
Interface Name	Displays in the **EBS** grid once an EBS has been selected. Displays the port type to which the selected EBS operation belongs.
Service Name	Displays once a service has been selected. Displays the name of the EBS that, based on routing rules, calls the provider ABC implementation service.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.

Copyright © 2009, Oracle. All rights reserved.

Cancel	Click to cancel edits to the page and access the Draft Provider Scenario Synopsis page.
Previous	Click to cancel any unsaved edits to the page and return to the Draft Provider Applications page.

Adding and Editing Enterprise Business Flow Scenarios

A draft EBF scenario outlines the EBF used in the integration scenario. The EBF represents orchestration logic used in the Oracle AIA canonical layers. The draft EBF scenario pages covered in this section enable you to specify the set of canonical services and flows that are choreographed by the EBF.

This section discusses how to:

- Add or edit a draft enterprise business flow scenario.
- Associate applications with a draft EBF integration scenario.
- Associate routing rules and external and enterprise business services with a draft EBF integration scenario.

Pages Used to Add and Edit EBF Scenarios

Page Name	Navigation	Usage
Add/Edit Draft EBF Scenario	• Access the Oracle AIA Console. Select the Service Repository tab. Click the Integration Scenario link. Click Draft Scenario Summary. Click Create in the Draft EBF Scenarios grid. • Click Scenario Detail on the Draft Ebf Scenario Synopsis page.	Add or edit a draft EBF scenario.
Select a Service	Click the Service Name button on the Add/Edit Draft EBF Scenario page.	Select the EBS that you want to associate with the draft EBF scenario. This is the EBS that serves as a front-façade before triggering the execution of the EBF application. Multiple EBF applications may be available from different vendors, so the routing rules in this EBS are tasked with mediating and routing runtime messages to the appropriate EBF applications.
Select an Operation	Click the Operation Name button on the Add/Edit Draft EBF Scenario page.	Select the operation you want to associate with the draft EBF scenario.

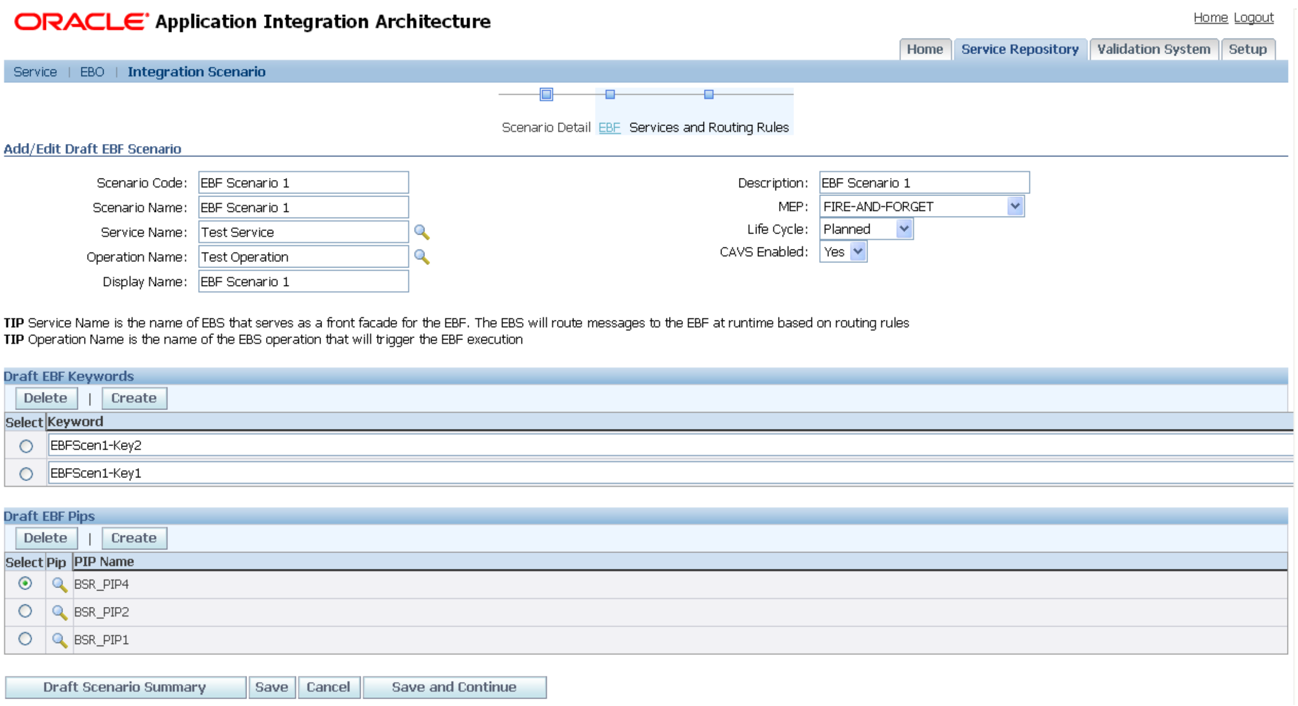
Page Name	Navigation	Usage
		This is the operation on the front-façade EBS. Within the operation, there are routing rules that enable runtime messages to be routed to the backend EBF application.
BSR Pips	Click the Pip button on the Add/Edit Draft EBF Scenario page.	Select a PIP that you want to associate with the EBF scenario. PIP values available for selection on this page are based on PIP values you have declared on the Setup - BSR Pips page. **For more information** about how to declare PIPs on the BSR Pips page, see Declaring a PIP.
Draft EBF Applications	• Click Save and Continue on the Add/Edit Draft EBF Scenario page. • Click EBF Applications on the Draft Ebf Scenario Synopsis page.	Associate EBF applications with the draft EBF scenario. Consider each EBF to be an application. You can have multiple EBF applications all implementing order orchestration, for example. Routing rules in the front-façade EBS route messages to the appropriate EBF application.
Select a Service	Click the Service Name button on the Add/Edit Draft EBF Scenario page.	Select the enterprise business service or enterprise business flow that you want to associate with the draft EBF scenario. These are the EBSs and EBFs that are being orchestrated and called by the EBF application. Behind these EBSs and EBFs are the provider ABC services and their respective provider applications.
Select an Operation	Click the Operation Name button on the Add/Edit Draft EBF Scenario page.	Select an operation on a selected EBS or EBF that is being orchestrated or called by the EBF application.
Draft EBF Routing Rules and Services	• Click Save and Continue on the Draft EBF Applications page. • Click Canonical Services & Routing Rules on the Draft Ebf Scenario Synopsis page.	Associate enterprise business services and external services with the EBF scenario. Associate routing rules with the EBF scenario.

Copyright © 2009, Oracle. All rights reserved.

Page Name	Navigation	Usage
Draft EBF Applications	Click the Applications button on the Draft EBF Routing Rules and Services page.	Select the application associated with the service you want to associate with the EBF scenario.
EBS	Click the Canonical Service button on the Draft EBF Routing Rules and Services page.	Select a canonical service that you want to associate with the EBF scenario.
External Services	Click the External Service button on the Draft EBF Routing Rules and Services page.	Select an external service that you want to associate with the EBF scenario.

Adding or Editing a Draft EBF Scenario

Access the Add/Edit Draft EBF Scenario page.


ORACLE Application Integration Architecture
Home Logout
Home
Service Repository
Validation System
Setup
Service | EBO | Integration Scenario
Scenario Detail
EBF
Services and Routing Rules
Add/Edit Draft EBF Scenario

Scenario Code:	EBF Scenario 1	Description:	EBF Scenario 1
Scenario Name:	EBF Scenario 1	MEP:	FIRE-AND-FORGET
Service Name:	Test Service	Life Cycle:	Planned
Operation Name:	Test Operation	CAVS Enabled:	Yes
Display Name:	EBF Scenario 1		

TIP Service Name is the name of EBS that serves as a front facade for the EBF. The EBS will route messages to the EBF at runtime based on routing rules
TIP Operation Name is the name of the EBS operation that will trigger the EBF execution

Draft EBF Keywords

Select	Keyword
○	EBFScen1-Key2
○	EBFScen1-Key1

Draft EBF Pips

Select	Pip	PIP Name
●	🔍	BSR_PIP4
○	🔍	BSR_PIP2
○	🔍	BSR_PIP1

Add/Edit Draft EBF Scenario page

Scenario Code	Enter a unique code to identify the EBF scenario.
Scenario Name	Enter a name for the EBF scenario.
Service Name	Click to access the Select a Service page, where you can select the enterprise business service that you want to associate with the draft EBF scenario. This is the EBS that serves as a front-façade before triggering the execution of the EBF application.
Operation Name	Click to access the Select an Operation page, where you can select the operation invoked by the front-façade EBS you

	associated with the EBF scenario. Within the operation, there are routing rules that enable runtime messages to be routed to the backend EBF application.
Display Name	Enter the user-friendly name of the service.
Description	Enter a description of the EBF scenario.
MEP (message exchange pattern)	Select the MEP used by the operation on the front-façade EBS. ***REQUEST RESPONSE*** ***FIRE-AND-FORGET*** ***REQUEST-DELAYED RESPONSE***
Life Cycle	Select a life cycle status for the integration scenario. This is value is not the development status of the integration scenario. It does not affect whether the integration scenario is active and displays on the Integration Scenario Summary page or is a draft and displays using the Integration Scenario Draft Summary page. ***Active*** ***Deprecated*** ***Obsolete*** ***Planned***
CAVS Enabled	Indicates whether the scenario has been developed to work with testing functionality provided by the CAVS.

For more information about enabling CAVS, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Developing CAVS-Enabled Oracle AIA Services."

Draft EBF Keywords

Use the **Draft EBF Keywords** grid to manage keywords for the EBF scenario. These keywords can be used as search criteria when searching for EBF scenarios using the Integration Scenario Summary page.

Delete	Displays when keywords are present. Select the **Select** option for a keyword and click **Delete** to delete the keyword.
Create	Click to add a keyword.
Keyword	Enter the keyword associated with the EBF scenario.

Copyright © 2009, Oracle. All rights reserved.

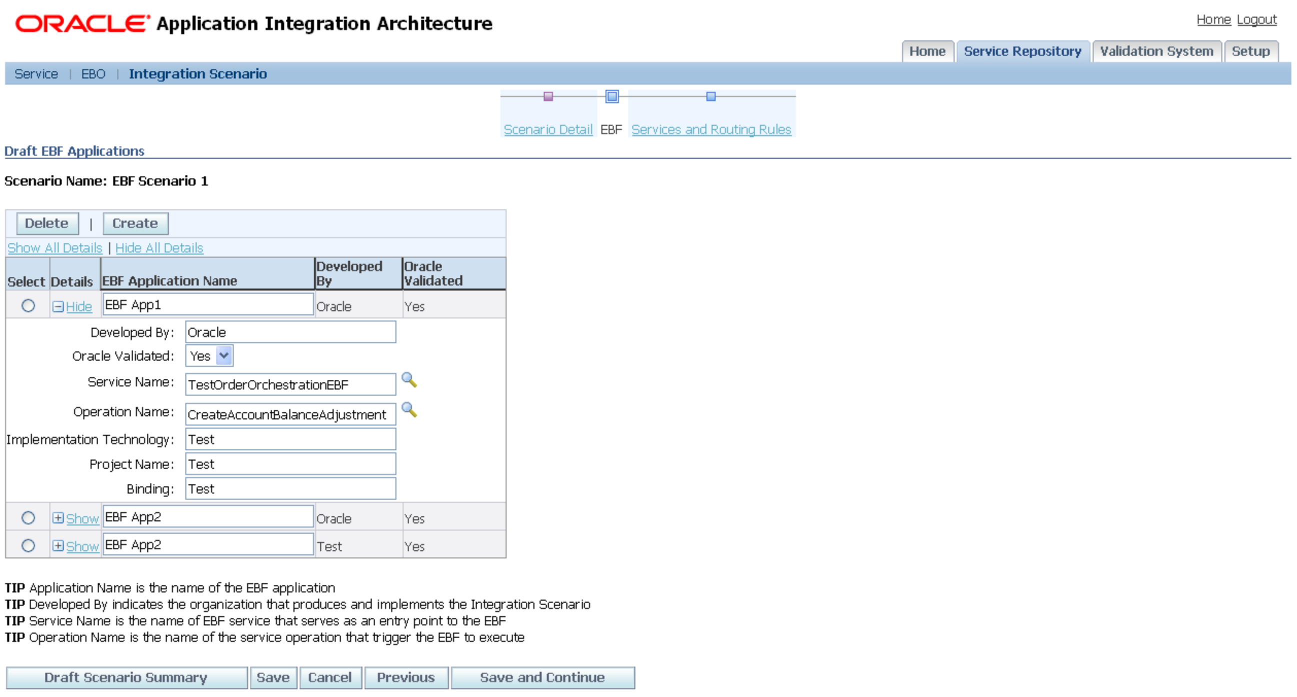
Draft EBF Pips

Use the **Draft EBF Pips** grid to manage PIPs associated with the EBF scenario. These are PIPs that utilize the EBF application.

Delete	Displays when PIPs have been associated with the EBF scenario.
Create	Click to associate a PIP with the EBF scenario and display the **Pip** button.
Pip	Click to access the BSR Pips page, where you can select a PIP that you want to associate with the EBF scenario. Available PIP values are derived from PIPs declared on the BSR PIPs page. **For more information** about declaring PIPs in the BSR, see Declaring a PIP in the BSR.
PIP Name	Displays the name of a PIP associated with the EBF scenario.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft EBF Scenario Synopsis page.
Save and Continue	Click to save edits to the page and access the Draft EBF Applications page, where you can associate applications with the EBF scenario.

Associating Applications with an EBF Scenario

Access the Draft EBF Applications page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Service | EBO | Integration Scenario
Scenario Detail | EBF | Services and Routing Rules

Draft EBF Applications

Scenario Name: EBF Scenario 1

Delete | Create

Show All Details | Hide All Details

Select	Details	EBF Application Name	Developed By	Oracle Validated
○	⊟Hide	EBF App1	Oracle	Yes
○	⊞Show	EBF App2	Oracle	Yes
○	⊞Show	EBF App2	Test	Yes

Developed By:	Oracle
Oracle Validated:	Yes
Service Name:	TestOrderOrchestrationEBF
Operation Name:	CreateAccountBalanceAdjustment
Implementation Technology:	Test
Project Name:	Test
Binding:	Test

TIP Application Name is the name of the EBF application
TIP Developed By indicates the organization that produces and implements the Integration Scenario
TIP Service Name is the name of EBF service that serves as an entry point to the EBF
TIP Operation Name is the name of the service operation that trigger the EBF to execute

Draft Scenario Summary | Save | Cancel | Previous | Save and Continue

Draft EBF Applications page

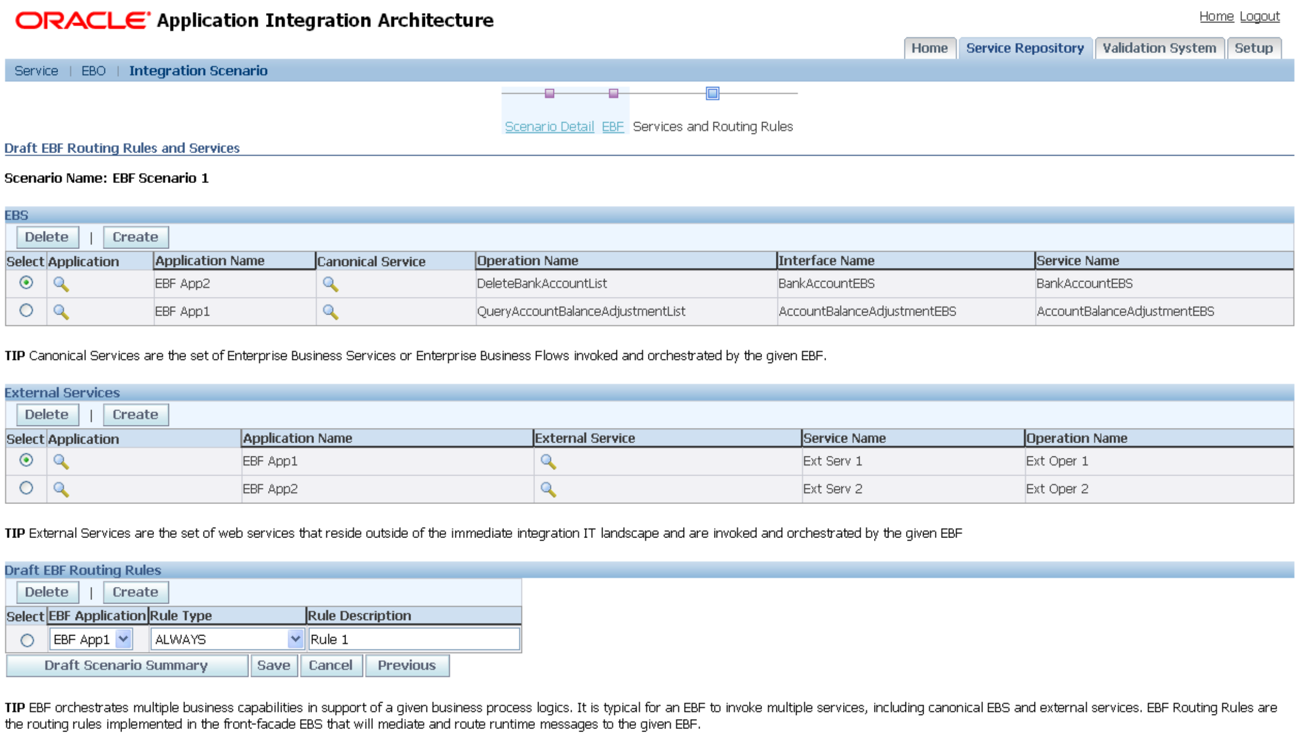
Delete	Displays when application values are present. Select the **Select** option for an application and click **Delete** to delete the association between the application and the EBF scenario.
Create	Click to create an association between an application and the EBF scenario.
Details	Click **Show** to enter and view details for the row. Click **Hide** to hide details for the row.
EBF Application Name	Enter the name of the EBF application. Consider each EBF to be an application. You can have multiple EBF applications all implementing order orchestration, for example. Routing rules in the front-façade EBS route messages to the appropriate EBF application.
Developed By	Enter the name of the vendor or system integrator that built the scenario.
Oracle Validated	Enter a value that indicates whether the EBF scenario was built following a formal Oracle Application Integration Architecture methodology program.
Service Name	Click to access the Select a Service page, where you can select the enterprise business service or enterprise business flow that you want to associate with the draft EBF scenario. These are the EBSs and EBFs that are being orchestrated and called by the EBF application. Behind these EBSs and EBFs are the provider ABC services and their respective

Copyright © 2009, Oracle. All rights reserved.

	provider applications.
Operation Name	Click to access the Select an Operation page, where you can select the operation invoked by the service you associated with the draft EBF scenario.
Implementation Technology	Enter the technology used to implement the EBF application.
Project Name	Enter the name of the ESB or BPEL project that houses the EBF application.
Binding	Enter the type of binding used by the service.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft EBF Scenario Synopsis page.
Previous	Click to cancel any unsaved edits to the page and return to the Add/Edit Draft EBF Scenario page.
Save and Continue	Click to save edits to the page and access the Draft EBF Routing Rules and Services page, where you can associate routing rules and services with the EBF scenario.

Associating Routing Rules and Services with an EBF Scenario

Access the Draft EBF Routing Rules and Services page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Service | EBO | Integration Scenario
Scenario Detail EBF Services and Routing Rules
Delete | Create
Draft Scenario Summary | Save | Cancel | Previous

Draft EBF Routing Rules and Services

Scenario Name: EBF Scenario 1

EBS

Select	Application	Application Name	Canonical Service	Operation Name	Interface Name	Service Name
●		EBF App2		DeleteBankAccountList	BankAccountEBS	BankAccountEBS
○		EBF App1		QueryAccountBalanceAdjustmentList	AccountBalanceAdjustmentEBS	AccountBalanceAdjustmentEBS

TIP Canonical Services are the set of Enterprise Business Services or Enterprise Business Flows invoked and orchestrated by the given EBF.

External Services

Select	Application	Application Name	External Service	Service Name	Operation Name
●		EBF App1		Ext Serv 1	Ext Oper 1
○		EBF App2		Ext Serv 2	Ext Oper 2

TIP External Services are the set of web services that reside outside of the immediate integration IT landscape and are invoked and orchestrated by the given EBF

Draft EBF Routing Rules

Select	EBF Application	Rule Type	Rule Description
○	EBF App1	ALWAYS	Rule 1

TIP EBF orchestrates multiple business capabilities in support of a given business process logics. It is typical for an EBF to invoke multiple services, including canonical EBS and external services. EBF Routing Rules are the routing rules implemented in the front-facade EBS that will mediate and route runtime messages to the given EBF.

Draft EBF Routing Rules and Services page

EBS and External Services

Use the **EBS** and **External Services** grids to designate the EBSs and external services that the EBF application orchestrates and calls.

Delete	Displays when service values are present. Select the **Select** option for a row and click **Delete** to delete the association between the service and the EBF scenario.
Create	Click to create an association between a service and the EBF scenario.
Application	Click to access the Draft EBF Applications page, where you can select from the EBF applications defined on the Draft EBF Applications page. You can then use the EBS page or External Services pages to specify the canonical and external services that the selected EBF application orchestrates.
Application Name	Displays the name of an EBF application you selected on the Draft EBF Applications page.
Canonical Service	Click to access the EBS page, where you can select the canonical services orchestrated by the EBF application. These are the EBSs and EBFs that are called by the selected EBF application. If your desired EBS is not yet present in the BSR, you can add an association by clicking **Create** on the EBS page and entering values. This association can serve as a requirement for addition of the EBS to the BSR.
External Service	Click to access the External Services page, where you can select an external service that you want to associate with the EBF scenario. These are the external services called by the selected EBF application. Available external service values are based on external services that are invoked by the selected EBF application. If your desired external service is not yet present in the BSR, you can add an association by clicking **Create** on the External Services page and entering values. This association can serve as a requirement for addition of the external service to the BSR.
Operation Name	Displays once a service has been selected. Displays the operation on the selected service that is called by the EBF application.
Interface Name	Displays in the EBS grid once an EBS has been selected. Displays the interface followed by the service.
Service Name	Displays once a service has been selected. Displays the name of the EBS or external service that is called by the EBF application.

Draft EBF Routing Rules

Use the **Draft EBF Routing Rules** grid to associate routing rules with the EBF scenario.

Copyright © 2009, Oracle. All rights reserved.

Delete	Displays when routing rule values are present. Select the **Select** option for a row and click **Delete** to delete the association between the routing rule and the EBF scenario.
Create	Click to create an association between a routing rule and the EBF scenario.
EBF Application	Select the EBF application to which the routing rule applies. Available values are derived from the values you entered on the Draft EBF Applications page.
Rule Type	Select a rule type. ***CONTENT-BASED:*** Indicates that based on the runtime message content, the front façade EBS will route calls to one of the EBF applications. ***ALWAYS:*** Indicates that regardless the runtime message content, the EBS will always route calls to an EBF application. ***TEST-HARNESS:*** Indicates that when CAVS is enabled for the front-façade EBS, a testing message can be routed to the CAVS.
Rule Description	Enter a description of the routing rule that allows runtime messages to be routed to a given EBF application. This indicates that there is a routing rule associated with the front façade EBS.
Draft Scenario Summary	Click to access the Integration Scenario Draft Summary page.
Save	Click to save edits to the page.
Cancel	Click to cancel edits to the page and access the Draft EBF Scenario Synopsis page.
Previous	Click to cancel any unsaved edits to the page and return to the Draft EBF Applications page.

Chapter 5: Managing the Oracle AIA Application Registry

This chapter discusses how to manage the Oracle Application Integration Architecture (AIA) application registry.

Managing the Application Registry

The purpose of the application registry is to identify and capture information about the requestor and provider systems that are participating in your Oracle AIA ecosystem. The application registry captures system attributes that are relatively static.

While the application registry for your Oracle AIA ecosystem is initially loaded during the Oracle AIA installation process, you can use the user interface to manage information in the registry.

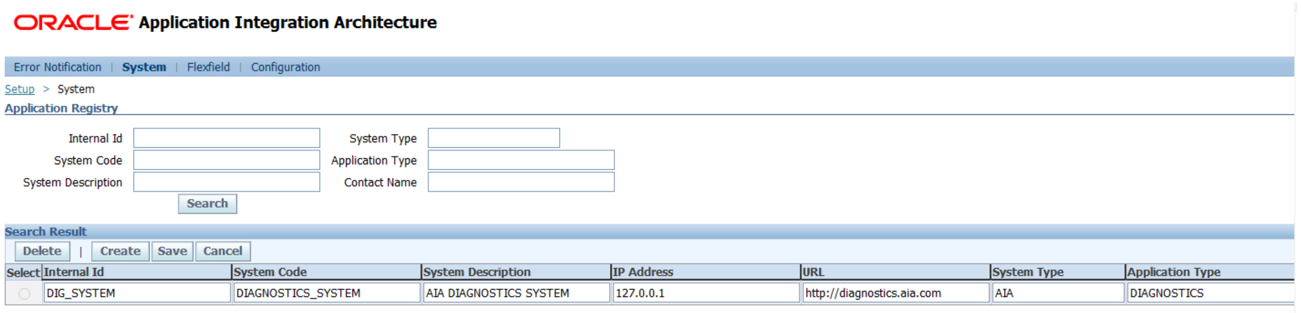
One of the benefits of storing an application registry is that certain system attributes can be easily defaulted to enterprise business message headers. Requestor and target systems may have multiple instances, each with different attributes. The application registry enables you to capture attributes for each instance so that they can be identified in subsequent processing.

Page Used to Manage the Application Registry

Page Name	Navigation	Usage
Application Registry	Access the Oracle AIA Console. Select the Setup tab. Click the System link.	Manage the application registry for your Oracle AIA ecosystem.

Managing Your Oracle AIA Application Registry

Access the Application Registry page.


ORACLE Application Integration Architecture

Error Notification | System | Flexfield | Configuration

Setup > System

Application Registry

Internal Id		System Type	
System Code		Application Type	
System Description		Contact Name	

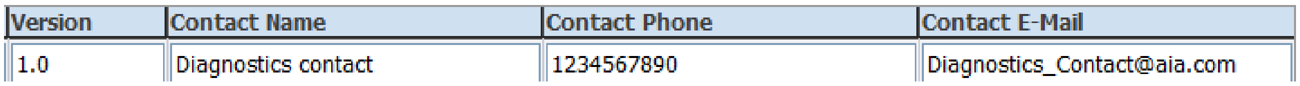
Search

Search Result

Delete | Create Save Cancel

Select	Internal Id	System Code	System Description	IP Address	URL	System Type	Application Type
○	DIG_SYSTEM	DIAGNOSTICS_SYSTEM	AIA DIAGNOSTICS SYSTEM	127.0.0.1	http://diagnostics.aia.com	AIA	DIAGNOSTICS

Application Registry page (1 of 2)



Version	Contact Name	Contact Phone	Contact E-Mail
1.0	Diagnostics contact	1234567890	Diagnostics_Contact@aia.com

Application Registry page (2 of 2)

Enter your search criteria and click **Search** to execute the search.

Search Result

Use the **Search Result** grid to work with participating system information.

Note: If a system is referenced by an error notification record defined on the Error Notifications page, then the Select option for the system row is not selectable.

For more information, see Setting Up Error Notifications for Oracle AIA Processes.

Delete	Select the **Select** option for a system row and click **Delete** to execute the deletion.
Create	Click to add a system row that you can use to add a new system to the application registry.
Save	Click to save all entries on the page.
Internal Id	System instance name assigned by an implementer, for example ***ORCL 01***. This value would be used in transformation and domain value map column names to indicate a system instance. For example, the value can be used in transformations to identify the requestor or provider system.
System Code	This is a required value. This value populated to the enterprise business message (EBM) header.
System Description	Long description of the requestor or provider system instance identified in the System Code field. This value populated to the EBM header.
IP Address	Provides the IP address of the system. This value populated to the EBM header.
URL	Provides the URL of the system. This value is populated to the EBM header.
System Type	Provides the system type. For example, ***Oracle EBIZ***. This is a required value.
Application Type	Application being run within the specified system. For example, a system type of ***Oracle EBIZ*** may have an application type value of ***FMS***. This value populated to the EBM header.
Version	Provides the version of the application being run by the system. This value populated to the EBM header.
Contact Name	Provides the name of the contact responsible for the system.

Copyright © 2009, Oracle. All rights reserved.

	This value populated to the EBM header.
Contact Phone	Provides the phone number of the contact responsible for the system. This value populated to the EBM header
Contact E-Mail	Provides the email address of the contact responsible for the system. This value is populated to the EBM header.

Chapter 6: Working with the BSR

This chapter discusses:

- Creating flex fields
- Searching in the Business Service Repository (BSR)
- Viewing enterprise business object and message details outside of the BSR user interface (UI)
- Performing an impact analysis

Creating Flex Fields

This section provides an overview of flex fields and discusses how to create flex fields.

Understanding Flex Fields in the BSR

In addition to the Oracle Application Integration Architecture (AIA) pre-defined annotations (in source artifacts, including WSDLs and XSDs), you can declare and enter custom fields, known as flex fields, from the UI. Flex fields allow you to introduce functional annotations on service artifacts and use them as enhanced search filters to provide easier access to the service-oriented architecture (SOA) assets in the Oracle AIA ecosystem. Flex fields support Oracle AIA first-class artifacts, including services, objects, and integration scenarios

Using flex fields is a three-step process:

- Declare

  In this step you associate the flex field with a given category, including an asset type and level.

- Define

  You enter or modify the value for a flex field instance on a given SOA asset.

- Search

  You can search based on the flex field instance value.

Declaring the Flex Field

In order to be able to declare a flex field you must have the appropriate access roles. Access is restricted to aiabsr.admin.role (AIAIntegrationAdmin) to ensure that an end-user does not arbitrarily change the fields.

You can associate flex fields with different asset types and on different levels. As part of the declaration flow, you specify which level or category of a given artifact type to attach the declared flex field to. For example, services can have flex fields on the service, interface, operation, and message levels.

This table display levels of respective artifact types that a flex field can potentially attach to.

- Service
- Service interface
- Service operation
- Service message
- Scenario requestor
- Scenario requestor application
- Scenario requestor connector
- Scenario enterprise business service (EBS)
- Scenario native
- Scenario external
- Scenario provider
- Scenario provider application
- Scenario provider connector

Note: There are no levels for enterprise business objects (EBOs).

You can format the flex fields as either a plain text entry field or a single select drop-down list. Default values can be entered for both plain text entry and drop down list types of flex fields.

Flex fields are translatable; however the Master Name must be in English. Search filters based on the flex fields are also translatable.

Defining the Flex Field

Once flex fields are declared and attached to a given artifact type, they are made available as part of the artifact's detailed page on UI. Initially, these flex fields appear as empty on the respective UI pages.

Once you enter and change values for a flex field for a given artifact, the flex field value is available and is displayed as part of the UI page of that artifact. The following sections provide details about declaring and defining flex fields.

Searching Using Flex Fields

Flex fields add additional filters that can simplify the user's search experience. Flex fields are automatically made available as search filters on the asset types they are associated with.

Copyright © 2009, Oracle. All rights reserved.

Creating Flex Fields

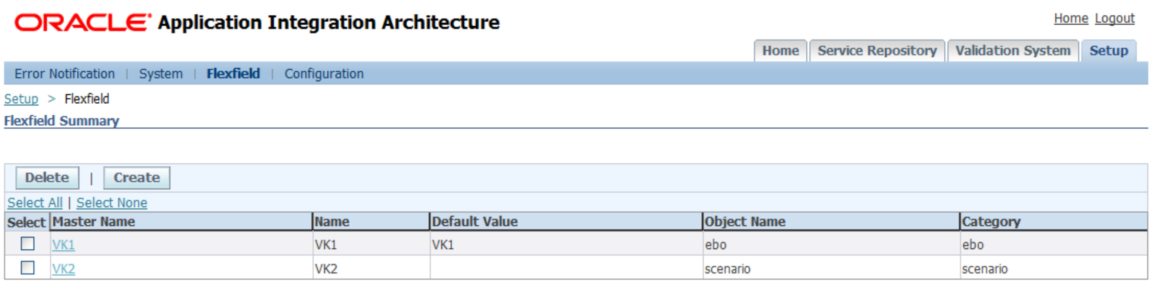
This section discusses how to create flex fields.

Pages Used to Create Flex Fields

Page Name	Navigation	Usage
Flex Fields Summary	On the Oracle AIA Console, Setup tab, click the Flexfield link.	View a list of the flex fields in your system.
Flex Field Setup	On the Flexfield Summary page, click Create.	Create a new flex field and associate with a category, asset type, and level.

Viewing the Flex Fields

Access the Flexfield Summary page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Error Notification | System | Flexfield | Configuration
Setup > Flexfield
Flexfield Summary
Delete | Create
Select All | Select None

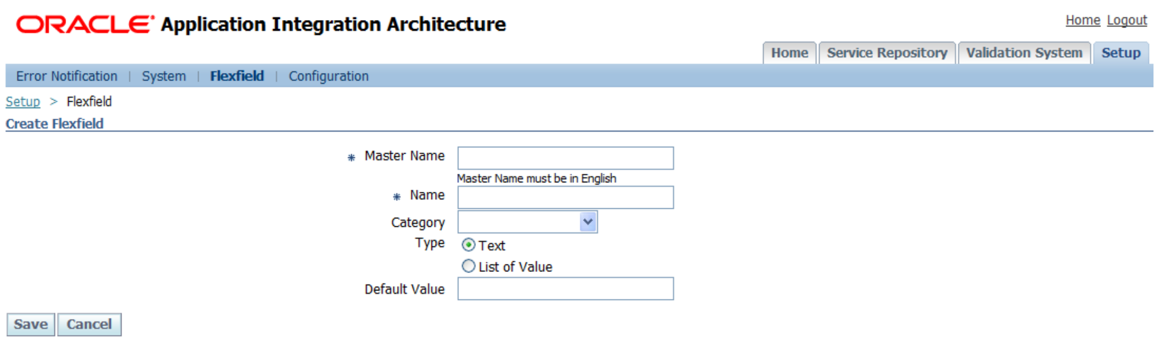
Select	Master Name	Name	Default Value	Object Name	Category
☐	VK1	VK1	VK1	ebo	ebo
☐	VK2	VK2		scenario	scenario

Flexfield Summary page

Create	Click **Create** to go to the Flexfield Setup page to create a new flex field.
Select	Click to select a value to delete or add.
Master Name	Displays the master flex field name as a link. Click the link to access the flex field page.
Name	Displays the user-defined name for the flex field. This is the name that will appear as the search filter and it can be translated.
Default Value	Displays the default value for the field, if one was defined.
Object Name	Displays the object name applicable to the flex field.
Category	Displays the category applicable to the flex field. Add the information, choose the category, and then particular levels.

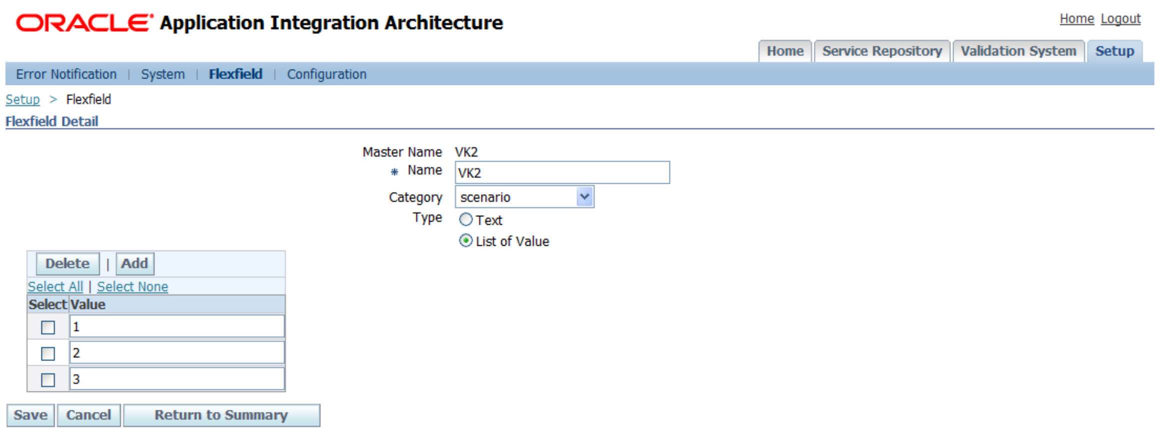
Creating a New Flex Field

Access the Flexfield Setup page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Error Notification | System | Flexfield | Configuration
Setup > Flexfield
Create Flexfield
Save
Cancel

* Master Name	
	Master Name must be in English
* Name	
Category	
Type	[✓ Text] List of Value
Default Value	

Flexfield Setup page 1 of 2


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Error Notification | System | Flexfield | Configuration
Setup > Flexfield
Flexfield Detail
Delete | Add
Select All | Select None
Save
Cancel
Return to Summary

Master Name	VK2
* Name	VK2
Category	scenario
Type	Text [✓ List of Value]

Select	Value
☐	1
☐	2
☐	3

Flexfield Setup page 2 of 2

Master Name	Enter a master name for the flex field. The master name must be in English.
Name	Enter the name that will appear as a search filter. This name can be translated.
Category	Enter the category applicable to the flex field.
	Values are:
	EBO
	Scenario; application, connector, ebs, external, native
	Service; interface, message, operation
Type	Select the type applicable to the flex field. (text or list of values)
Default Value	Enter an optional default value

Copyright © 2009, Oracle. All rights reserved.

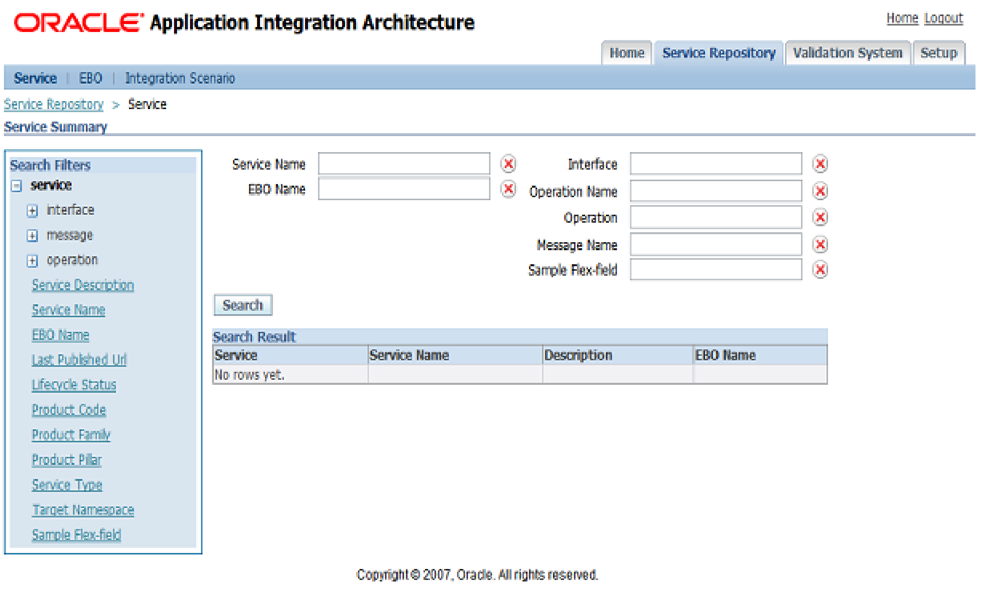
Drop Down List Box

Use this box to enter values for a single-select drop down list box.

Delete	Click to delete a value or set of values for the drop down list.
Add	Click **Add** to add additional lines to enter values.
Select All/Select None	Click to select or clear all the value check boxes.
Select	Click to select a value to delete or add.
Value	Enter a value to appear in the drop down list.
Save	Click **Save** to save the flex field data.

Using Flex Fields as Search Filters

Access the Service Search page.



Service	Service Name	Description	EBO Name
No rows yet.			

Copyright © 2007, Oracle. All rights reserved.

Service Search page

This page illustrates the flex field used as a search filter on a Service Summary page. The new flex field appears in the **Search Filters** tree and as a field label for data entry.

Searching in the BSR

This section discusses searches in the BSR.

Understanding Searches in the BSR

Search is one of the basic access paradigms to locate Oracle AIA-related SOA assets, including WSDLs, XSDs, and integration scenarios. Since the default search filters are based on pre-defined annotations in the source code of WSDLs and XSDs, these filters and annotations are more relevant to developers. By adding customer-defined search filters based on flex fields, you can create filters that are more relevant to the end user.

BSR enables you to search all first class assets:

- Services – application business connector (ABC) services and enterprise business services (EBSs)
- Objects – enterprise business objects (EBOs) and enterprise business messages (EBMs)
- Integration scenarios

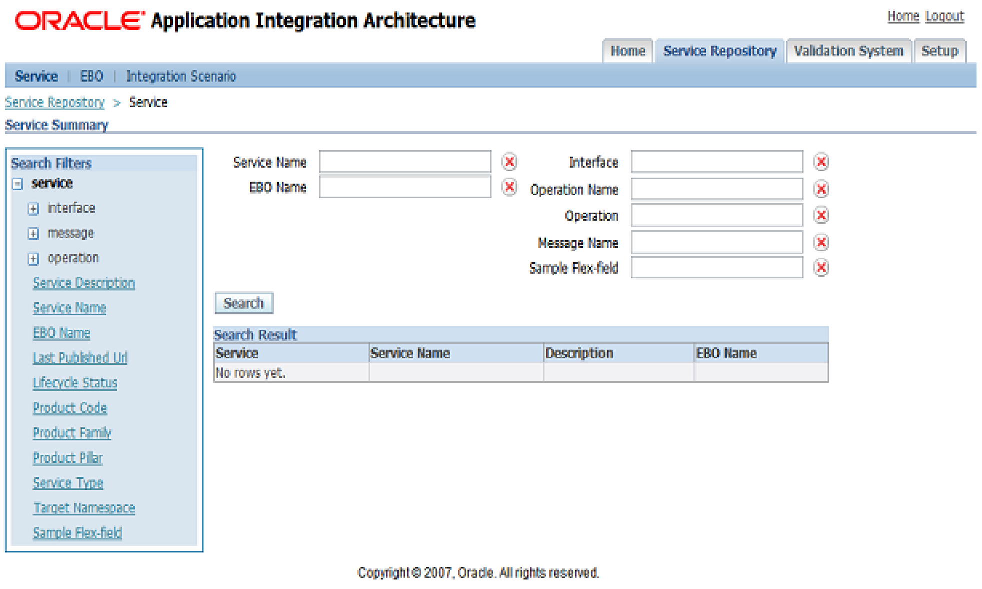
Assets are searchable at multiple levels. For example, services can be searched at the:

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

User-defined flex fields become additional search filters.

Copyright © 2009, Oracle. All rights reserved.

This page illustrates the search filter tree on services:



Search Result

Service	Service Name	Description	EBO Name
No rows yet.			

Copyright © 2007, Oracle. All rights reserved.

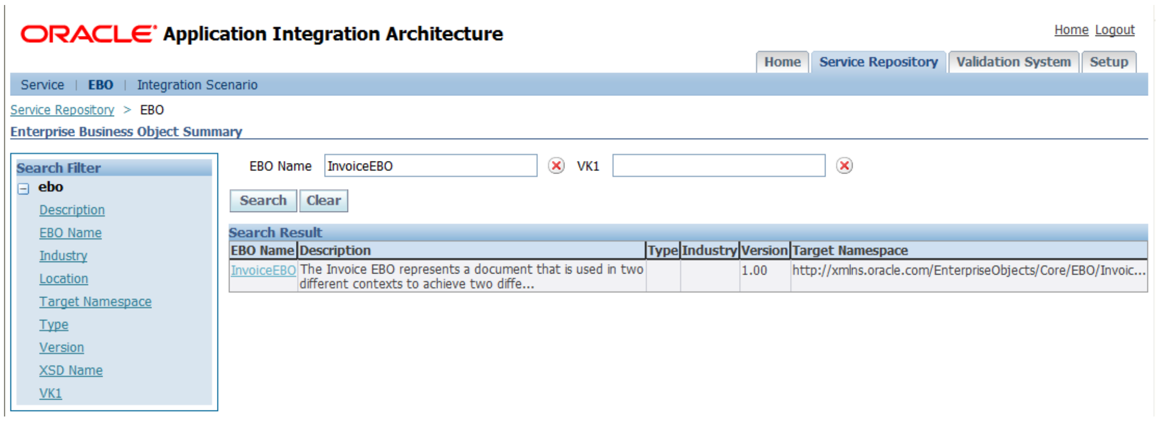
Service search showing search filter tree

The tree lists the default search filters derived from annotations in the source WSDLs. You can see that there are additional structural levels within the service level. There are different pre-defined annotations on each of these levels so the tree displays search filters for each level. This page also shows a user-defined flex field that has been added as an additional filter

Note: Any search filter, whether based on default annotations or user-defined flex fields is translatable.

Integration Scenario Search

Search filters for an integration scenario are available at multiple levels including scenarios, connectors, EBS, external services, and native services. Searches can also be performed for a specific process integration pack (PIP) tag. This enables the user to search for integration scenarios used in specific process integration packs. Search filters for EBOs and EBMs, as illustrated in this page, are restricted to the top level.



EBO Name: InvoiceEBO

EBO Name	Description	Type	Industry	Version	Target Namespace
InvoiceEBO	The Invoice EBO represents a document that is used in two different contexts to achieve two diffe...			1.00	http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/Invoic...

EBO Search

For more information, see Chapter 3: Using the BSR UI to View Integration Scenarios.

Viewing EBO and EBM Details Outside of the BSR UI

The Oracle AIA Foundation Pack includes HTML files that provide details about the EBOs and EBMs delivered in the release. These HTML files are accessible using the UI of the BSR.

However, you can also access these HTML files without having to use the BSR UI.

This section discusses how to directly access the following HTML files:

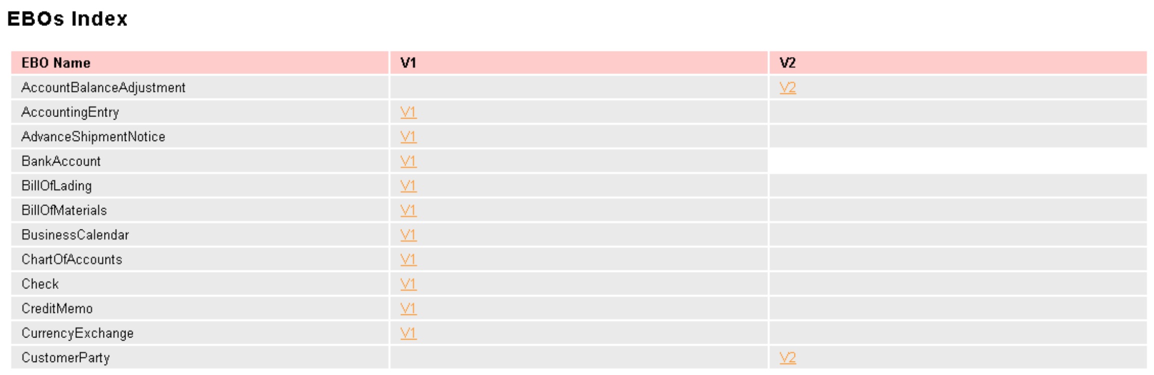
- EBO index HTML file.
- EBM index HTML file.

Accessing the EBO Index

The EBO index HTML file lists all EBOs included in the Foundation Pack release. Each EBO name listed in the index has an associated link to an HTML file that provides detailed information about the EBO.

You can access the EBO index HTML file here: http://[HOST:PORT]/AIAComponents/EnterpriseObjectLibrary/Core/EBOIndex.html

Copyright © 2009, Oracle. All rights reserved.



EBOs Index

EBO Name	V1	V2
AccountBalanceAdjustment		V2
AccountingEntry	V1	
AdvanceShipmentNotice	V1	
BankAccount	V1	
BillOfLading	V1	
BillOfMaterials	V1	
BusinessCalendar	V1	
ChartOfAccounts	V1	
Check	V1	
CreditMemo	V1	
CurrencyExchange	V1	
CustomerParty		V2

EBOIndex.HTML

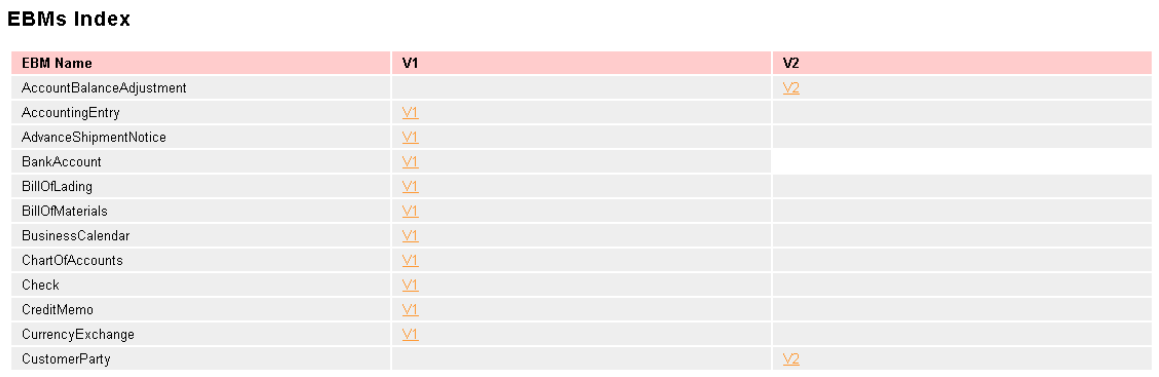
If there is a V1 link for the EBO, the EBO is in version 1. If there is a V2 link for the EBO, the EBO is in version 2.

Note: If both V1 and V2 links are present for a single EBO, use the details provided by the V2 link.

Accessing the EBM Index

The EBM index HTML file lists all EBMs included in the Foundation Pack release. Each EBM name listed in the index has an associated link to an HTML file that provides detailed information about the EBM.

You can access the EBM index HTML file here: http://[HOST:PORT]/AIAComponents/EnterpriseObjectLibrary/Core/EBMIndex.html



EBMs Index

EBM Name	V1	V2
AccountBalanceAdjustment		V2
AccountingEntry	V1	
AdvanceShipmentNotice	V1	
BankAccount	V1	
BillOfLading	V1	
BillOfMaterials	V1	
BusinessCalendar	V1	
ChartOfAccounts	V1	
Check	V1	
CreditMemo	V1	
CurrencyExchange	V1	
CustomerParty		V2

EBMIndex.html

If there is a V1 link for the EBM, the EBM is in version 1. If there is a V2 link for the EBM, the EBM is in version 2.

Performing an Impact Analysis

This section provides an overview and discusses how to perform an impact analysis.

Understanding Impact Analysis

Oracle AIA introduces a large set of SOA artifacts, including WSDLs, XSDs, XMLs, and so forth. These artifacts are intricately linked. Specifically, there are two types of dependencies:

- Artifact dependencies
- Functional dependencies

Artifact Dependencies

Dependencies exist among artifacts; for instance, a service WSDL imports an EBM XSD. The EBM XSD file in turn includes a corresponding EBO XSD, which subsequently imports or includes other component XSDs (that is, common components). In this case, there is a chain of relationships going in both directions. Any artifact or structure changes along this chain will impact the related artifacts.

Functional Dependencies

Multiple services and objects collectively support a functional flow such as an integration scenario. While they are distinct artifacts in their own right, they are interwoven together based on business or integration logic.

Functional impact analysis with respect to Oracle AIA-style process integration is not based on physical artifact dependencies such as import/include relationship among WSDLs and XSDs source artifacts or BPEL partnerLinks, but rather is based on logical relationship with respect to the integration flows: As such, Oracle AIA impact analysis can achieve:

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

Impact Analysis in the BSR

BSR delivers impact analysis focused on service and operation changes. For instance, if a service interface, either an EBS or an ABC services, were modified, what would be impact on the rest of integration flows. Namely,

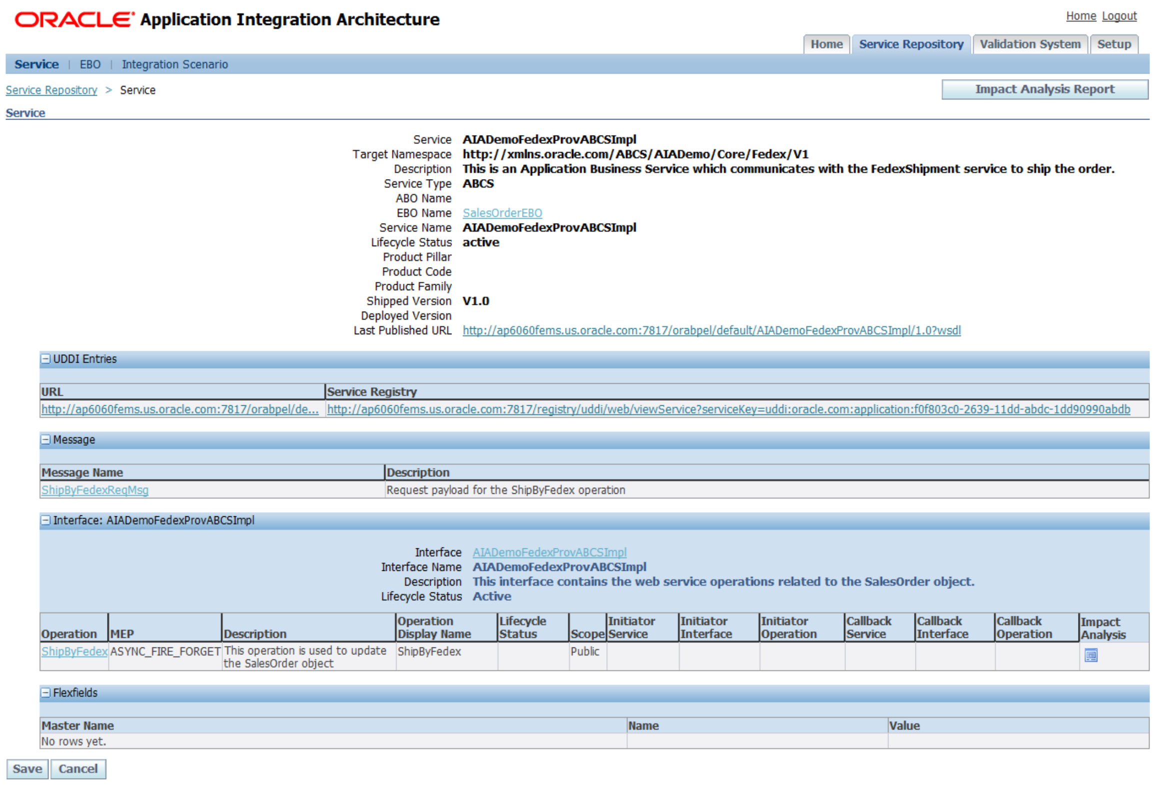
- What are the possible neighboring services affected?
- Which integration scenarios are impacted given that the changed services participate in the flows?
- Which PIPs should be re-evaluated due the service changes?

Note: The originating factor of service change is on its interface, not internal implementation or logic.

Copyright © 2009, Oracle. All rights reserved.

You have two options for performing an impact analysis.

- Select Impact Analysis Report to run an impact analysis report that evaluates the potential impact of changing the given service's interface (WSDL), and what would be the possible impacts to the ecosystems, such as neighboring services, integration scenarios, and PIPs.
- Click the Impact Analysis button on an Operations line to gauge the potential impact of changing the given operation on the particular service.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Service | EBO | Integration Scenario
Service Repository > Service
Impact Analysis Report
Service

Service	AIADemoFedexProvABCSImpl
Target Namespace	http://xmlns.oracle.com/ABCS/AIADemo/Core/Fedex/V1
Description	This is an Application Business Service which communicates with the FedexShipment service to ship the order.
Service Type	ABCS
ABO Name	
EBO Name	SalesOrderEBO
Service Name	AIADemoFedexProvABCSImpl
Lifecycle Status	active
Product Pillar	
Product Code	
Product Family	
Shipped Version	V1.0
Deployed Version	
Last Published URL	http://ap6060fems.us.oracle.com:7817/orabpel/default/AIADemoFedexProvABCSImpl/1.0?wsdl

UDDI Entries

URL	Service Registry
http://ap6060fems.us.oracle.com:7817/orabpel/de...	http://ap6060fems.us.oracle.com:7817/registry/uddi/web/viewService?serviceKey=uddi:oracle.com:application:f0f803c0-2639-11dd-abdc-1dd90990abdb

Message

Message Name	Description
ShipByFedexReqMsg	Request payload for the ShipByFedex operation

Interface: AIADemoFedexProvABCSImpl

Interface	AIADemoFedexProvABCSImpl
Interface Name	AIADemoFedexProvABCSImpl
Description	This interface contains the web service operations related to the SalesOrder object.
Lifecycle Status	Active

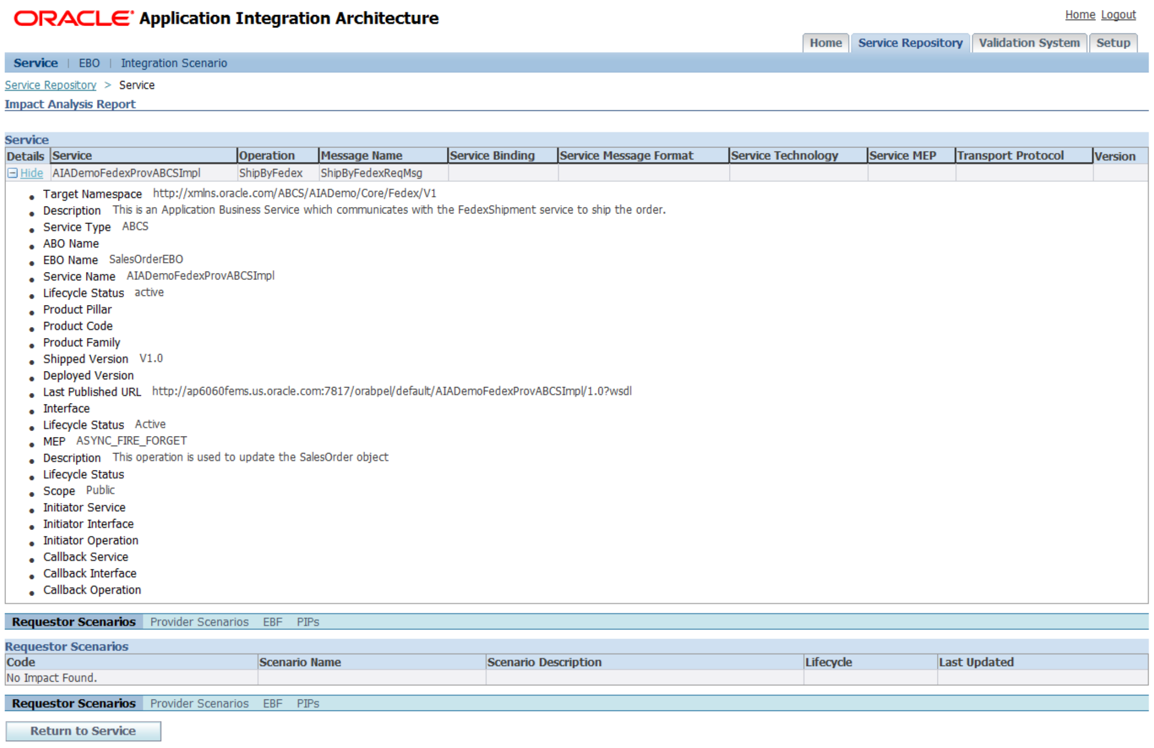
Operation	MEP	Description	Operation Display Name	Lifecycle Status	Scope	Initiator Service	Initiator Interface	Initiator Operation	Callback Service	Callback Interface	Callback Operation	Impact Analysis
ShipByFedex	ASYNC_FIRE_FORGET	This operation is used to update the SalesOrder object	ShipByFedex		Public							▤

Flexfields

Master Name	Name	Value
No rows yet.		

Service page showing Impact Analysis options

The Impact Analysis results are presented in UI based reports as illustrated on this page:


ORACLE Application Integration Architecture
Home Logout
Home
Service Repository
Validation System
Setup
Service | EBO | Integration Scenario
Service Repository > Service

Impact Analysis Report

Service

Details	Service	Operation	Message Name	Service Binding	Service Message Format	Service Technology	Service MEP	Transport Protocol	Version
Hide	AIADemoFedexProvABCSImpl	ShipByFedex	ShipByFedexReqMsg						

- Target Namespace http://xmlns.oracle.com/ABCS/AIADemo/Core/Fedex/V1
- Description This is an Application Business Service which communicates with the FedexShipment service to ship the order.
- Service Type ABCS
- ABO Name
- EBO Name SalesOrderEBO
- Service Name AIADemoFedexProvABCSImpl
- Lifecycle Status active
- Product Pillar
- Product Code
- Product Family
- Shipped Version V1.0
- Deployed Version
- Last Published URL http://ap6060fems.us.oracle.com:7817/orabpel/default/AIADemoFedexProvABCSImpl/1.0?wsdl
- Interface
- Lifecycle Status Active
- MEP ASYNC_FIRE_FORGET
- Description This operation is used to update the SalesOrder object
- Lifecycle Status
- Scope Public
- Initiator Service
- Initiator Interface
- Initiator Operation
- Callback Service
- Callback Interface
- Callback Operation

Requestor Scenarios

Code	Scenario Name	Scenario Description	Lifecycle	Last Updated
No Impact Found.				

Impact Analysis report

Copyright © 2009, Oracle. All rights reserved.

Chapter 7: Loading Oracle AIA Configuration File Updates

This chapter discusses how to load manual updates made to the Oracle Application Integration Architecture (AIA) configuration properties file.

Loading Manual Updates to the Oracle AIA Configuration Properties File

Various configurations that apply to the entire Oracle AIA system, Core Infrastructure Components, and specific process integration services are stored in the AIAConfigurationProperties.xml file located in <aia.home>/config/.

The Java API and XPath functions that provide access to these property values at runtime do not read the properties from the XML file. Instead, they read the data from a hashmap stored in memory. Therefore, if a property is manually updated in AIAConfigurationProperties.xml, the new property value must be updated to the cache to be reflected in the applications or services that use the property.

To update the cache, you can do a manual reboot of the SOA server, or you can click Reload on the Configuration page. When you click Reload, the system reads the AIAConfigurationProperties.xml file and loads the properties into the cache.

If you have updated Composite Application Validation System (CAVS) routing configurations using the Routing Setup pages, you do not need to use the Reload button on this page to load the updates to AIAConfigurationProperties.xml and the hashmap file. The Routing Setup pages automatically perform the reload.

For more information about updating CAVS routing configurations using the Routing Setup pages, see Chapter 15: Defining CAVS Routing Setup IDs.

You can use this page to quickly set up a CAVS routing configuration without having to create routing setup IDs. This is particularly useful when you are only interested in using CAVS simulators without CAVS test definitions. For example, you may only need to use the CAVS simulator feature for your development purposes and you may not need to uptake the complexity involved in setting up routing setup IDs. In this case, you can use this page to directly modify service routing configurations in the AIAConfigurationProperties.xml file.

However, if you are using CAVS for extensive testing purposes, we recommend that you using the Routing Setup pages to create your routing setups.

Page Used to Load Oracle AIA Configuration File Updates

Page Name	Navigation	Usage

Page Name	Navigation	Usage
Configuration	Access the Oracle AIA Console. Select the Setup tab. Click the Configuration link.	Click Reload to load any updates to the Oracle AIA configuration properties file to the hashmap without having to reboot the server. Quickly set up a CAVS routing configuration without having to create routing setup IDs.

Copyright © 2009, Oracle. All rights reserved.

Part 2: Working with the CAVS

Chapter 8: Understanding the CAVS

This chapter provides overviews of:

- The purpose of the Composite Application Validation System (CAVS).
- Key components of the CAVS framework.
- CAVS design assumptions and knowledge prerequisites.

Describing the Purpose of the CAVS

The CAVS is a framework that provides a structured approach to test integration of Oracle Application Integration Architecture (AIA) services. The CAVS includes test initiators that simulate web service invocations and simulators that simulate service endpoints.

In the context of Oracle AIA, where there is a sequence of service invocations; spanning application business connector (ABC) services, enterprise business services (EBSs), enterprise business flows (EBFs), and participating applications; the CAVS test initiators and simulators enable a layered testing approach. Each component in an integration can be thoroughly tested without having to account for dependencies by using test initiators and simulators on either end.

Consequently, when you build an integration, you have the ability to add new components to an already tested subset, allowing any errors to be constrained to the new component or to the interface between the new component and the existing component. This ability to isolate and test individual web services within an integration provides the benefit of narrowing the test scope, thereby distancing the service test from possible faults in other components.

Test initiators and simulators can be used independent of each other, thereby allowing users to effectively substitute them for non-available Oracle AIA services or participating applications.

The CAVS provides a repository that stores these test initiator and simulator definitions created by the CAVS user, as well as an interactive user interface to create and manage the same. Tests can be configured to run individually or in a single-threaded batch.

The CAVS provides value as a testing tool throughout the integration development life cycle:

- Development

  Because integration developers working with Oracle AIA are dealing with integrating disparate systems, they typically belong to different teams. To this end, the CAVS provides an effective way to substitute dependencies, letting developers focus on the functionality of their own service rather than being preoccupied with integrations to other services.

- Quality assurance

  The CAVS allows quality assurance engineers to unit and flow test integrations, thereby providing a way to easily certify different pieces of an integration. The reusability of test definitions, simulators, and test groups helps in regression testing and provides a quick way to certify new versions of services.

Describing Key Components of the CAVS Framework

The CAVS framework operates using the following key components:

- Test definition
- Simulator definition

Test Definition

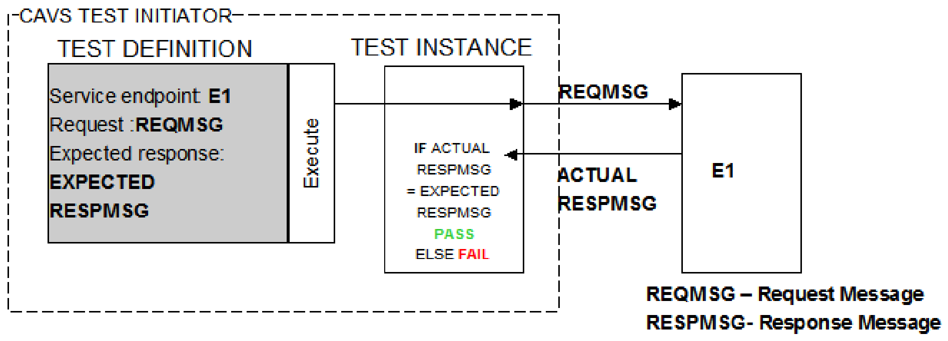
The CAVS test initiator reads test data and feeds it to the web service being tested. You create the test data as a part of a test definition. The test definition is a configuration of the test initiator and contains test execution instructions.

The CAVS user creates a definition using the CAVS user interface (UI) to define the service endpoint URL that needs to be invoked, as well as the request message that will be passed along with metadata about the test definition itself.

For more information about creating test definitions, see Creating and Modifying Test Definitions.

The test initiator is a logical unit that executes test definitions to call the endpoint URL defined and creates test instances. This call is no different from any other request initiated by other clients. If the test definition Service Type value is set to Synchronous or Asynchronous two way, the actual response can be verified against predefined response data to validate the accuracy of the response.

This diagram illustrates the high-level concept of the test initiator:



CAVS Test Definition

Simulator Definition

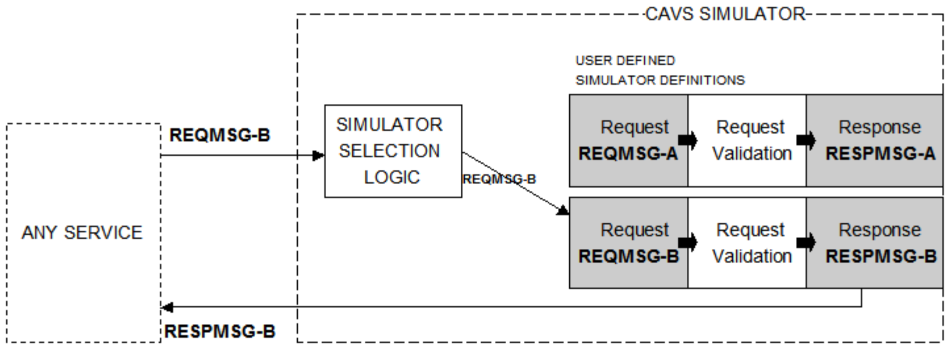
The CAVS simulator is used to simulate a web service. Simulators typically contain predefined responses for a specific request. CAVS users create several simulator definitions, each for a specific set of input.

At runtime, the CAVS simulator framework receives data from the service being tested. Upon receiving the request, CAVS locates the appropriate simulator definition, validates the input against predefined request values, and then returns predefined response data so that the web service being tested can continue processing.

Copyright © 2009, Oracle. All rights reserved.

For more information about creating simulator definitions, see Creating and Modifying Simulator Definitions.

This diagram illustrates the high-level concept of the CAVS simulator:



CAVS Simulator

Describing the CAVS Design Assumptions and Knowledge Prerequisites

The CAVS operates with the following design assumptions:

- The CAVS assumes that the requester and provider ABC services it is testing are implemented using BPEL.
- The CAVS is designed to initiate requests and simulate responses as SOAP messages using SOAP over HTTP.

  The request and response messages that you define in test and simulator definitions must contain the entire XML SOAP document, including the SOAP envelope, message header, and body (payload).
- The correlation logic between the test initiator and the response simulator is based on timestamps only.

  For this reason, test and simulator instances generated in the database schema will not always be reconcilable, especially when the same web service is invoked multiple times during a very short time period, as in during performance testing.
- The CAVS does not provide or authenticate security information for web services that are initiated by a test initiator or received by a response simulator.

  However, security information passed through the system by the web service can be used as a part of verification and validation logic.
- When a participating application is involved in a CAVS testing flow, execution of tests can potentially modify data in a participating application.

Therefore, consecutive running of the same test may not generate the same results. The CAVS is not designed to prevent this kind of data tampering because it supports the user’s intention to include a real participating application in the flow. The CAVS has no control over modifications that are performed in participating applications.

This issue does not apply if your CAVS test scenario uses test definitions and simulator definitions to replace all participating applications and other dependencies. In this case, all cross-reference data is purged once the test scenario has been executed. This enables rerunning of the test scenario.

Note: CAVS cross reference data is purged at the end of a test execution when executing a test definition and at the end of a test group execution when executing a test group definition. Therefore, if you want to execute test definitions that are dependent on cross referencing data created by earlier test executions, ensure that you include all dependent test definitions in a test group and execute the test group.

For more information about how to make test scenarios rerunnable, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* “Purging CAVS-Related Cross Reference Entries to Enable Rerunning of Test Scenarios.”

To work effectively with the CAVS, users must have working knowledge of the following concepts and technologies:

- Oracle AIA
- XML
- XPath
- SOAP

Copyright © 2009, Oracle. All rights reserved.

Chapter 9: Preparing to Use the CAVS

Before you start creating and running tests in the Composite Application Validation System (CAVS), take the time to gather your test requirements and plan your approach to using CAVS. This chapter provides a high-level discussion of the following questions you should answer to help gather requirements for the tests you want to create and run in the CAVS.

- What can I test using CAVS?
- What are the Oracle Application Integration Architecture (AIA) components that I need to test?
- Which message exchange pattern is being used by the components being tested?
- Do I need to unit test or flow test?
- Do I have the content I need to create the definitions?

What Can I Test Using CAVS?

The CAVS supports the following testing scenarios:

- Create and execute test definitions against actual services in participating applications.
- Create and execute test definitions that call services that call simulators, which simulate actual services in participating applications.
- Use actual services in participating applications in cooperation with simulators to simulate any unavailable services.

What Are the Oracle AIA Components That I Need to Test?

Examine the components involved in the scenario that you need to test. Which of the following components does the scenario include?

- Requester application business connector (ABC) services
- Provider ABC services
- Enterprise business flows (EBFs)
- Exposed services in participating applications

Which Message Exchange Pattern Is Being Used by the Components Being Tested?

Once you have assessed which components you need to test, identify the message exchange pattern (MEP) being used by the components to help you determine which types of CAVS test and simulator definitions you need to create. Based on the sequence of service calls and the MEPs employed, you can determine if you need to use synchronous, notify, or asynchronous two-way test definitions and simulator definitions.

This section discusses CAVS process flows for testing the following MEPs:

- Synchronous (request-and-response)
- Asynchronous (notify)
- Asynchronous two-way

Note: The information in this chapter provides CAVS processing details that can inform your creation of test and simulator definitions in the CAVS user interface (UI). As you prepare to define and run tests for a particular web service, refer to the section in this chapter that corresponds to the message exchange pattern of the service you want to test.

Describing CAVS Process Flows for Testing the Synchronous Message Exchange Pattern

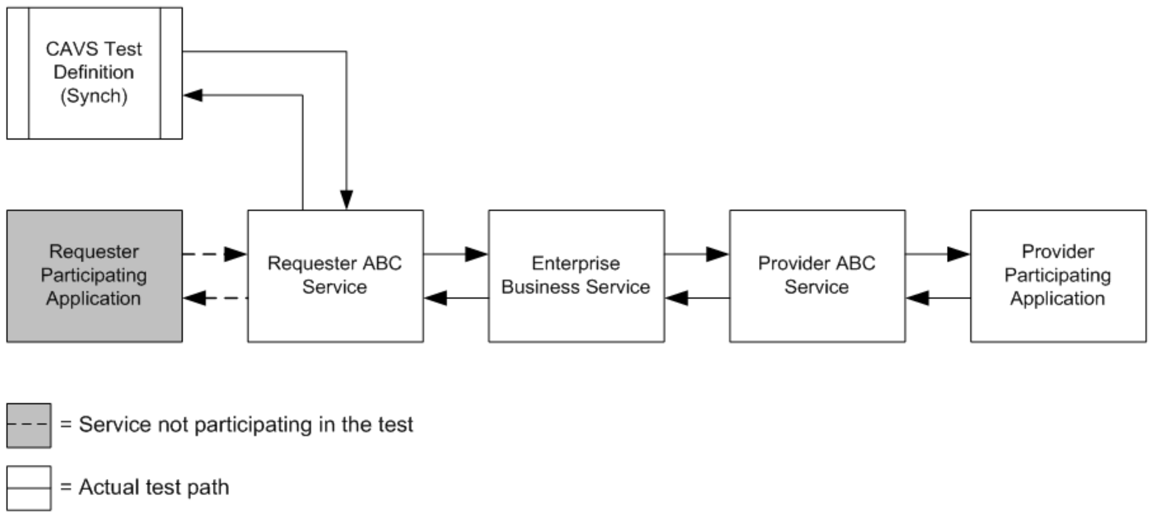
The following diagrams describe CAVS process flows for testing a provider ABC service using a synchronous MEP.

These sample flows can be used as the basis for testing other artifacts as well, such as requester ABC services, enterprise business flows, or provider services.

Synchronous MEP Testing Flow Using a Test Definition

The requester participating application is replaced by the CAVS test definition. The test definition points to the URL of the requester ABC service. It uses a composed request message to invoke the ABC service and expects a message in response.

Copyright © 2009, Oracle. All rights reserved.


CAVS Test Definition (Synch)
Requester Participating Application
Requester ABC Service
Enterprise Business Service
Provider ABC Service
Provider Participating Application
= Service not participating in the test
= Actual test path

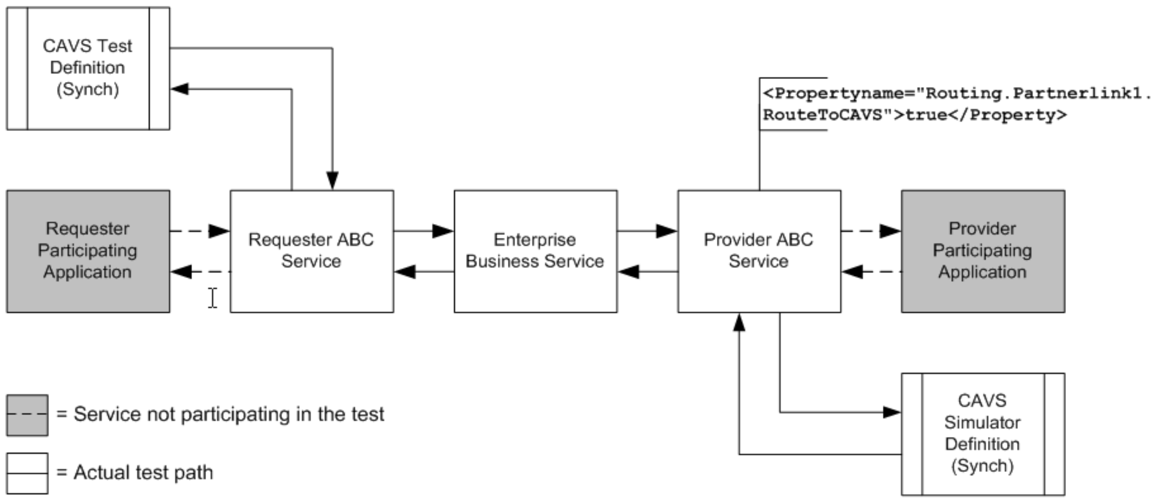
Testing a synchronous MEP using a CAVS test definition

Synchronous MEP Testing Flow Using a Test Definition and Simulator Definition

The requester participating application is replaced by the CAVS test definition. The test definition points to the URL of the requester ABC service. It uses a composed request message to invoke the ABC service and expects a message in response.

The provider participating application is replaced by the CAVS simulator definition. The provider ABC service is programmed to route to this simulator instead of the provider participating application. The simulator definition contains a predefined request and response message pair.

The simulator definition performs validations on message input from the provider ABC service and sends the message back to the provider ABC service. The provider ABC service sends the message back to the test definition, which validates this actual response against its predefined expected response.

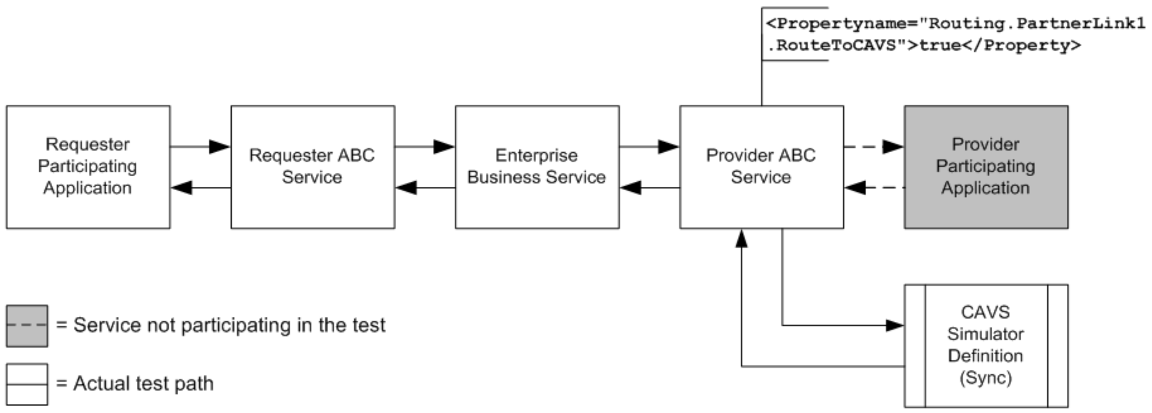

CAVS Test Definition (Synch)
<Propertyname="Routing.Partnerlink1.RouteToCAVS">true</Property>
Requester Participating Application
Requester ABC Service
Enterprise Business Service
Provider ABC Service
Provider Participating Application
CAVS Simulator Definition (Synch)
= Service not participating in the test
= Actual test path

Testing a synchronous MEP using a CAVS test definition and simulator definition

Synchronous MEP Testing Flow Using a Simulator Definition

The provider participating application is replaced by the CAVS simulator definition. The provider ABC service is programmed to route to this simulator instead of the provider participating application. The simulator definition contains a predefined request and response message pair.

The simulator definition performs validations on message input from the provider ABC service and sends the message back to the provider ABC service. The provider ABC service sends the message back to requester participating application.



<Propertyname="Routing.PartnerLink1
.RouteToCAVS">true</Property>
Requester Participating Application
Requester ABC Service
Enterprise Business Service
Provider ABC Service
Provider Participating Application
CAVS Simulator Definition (Sync)
= Service not participating in the test
= Actual test path

Testing a synchronous MEP using a CAVS simulator definition

Describing CAVS Process Flows for Testing the Asynchronous (Notify) Message Exchange Pattern

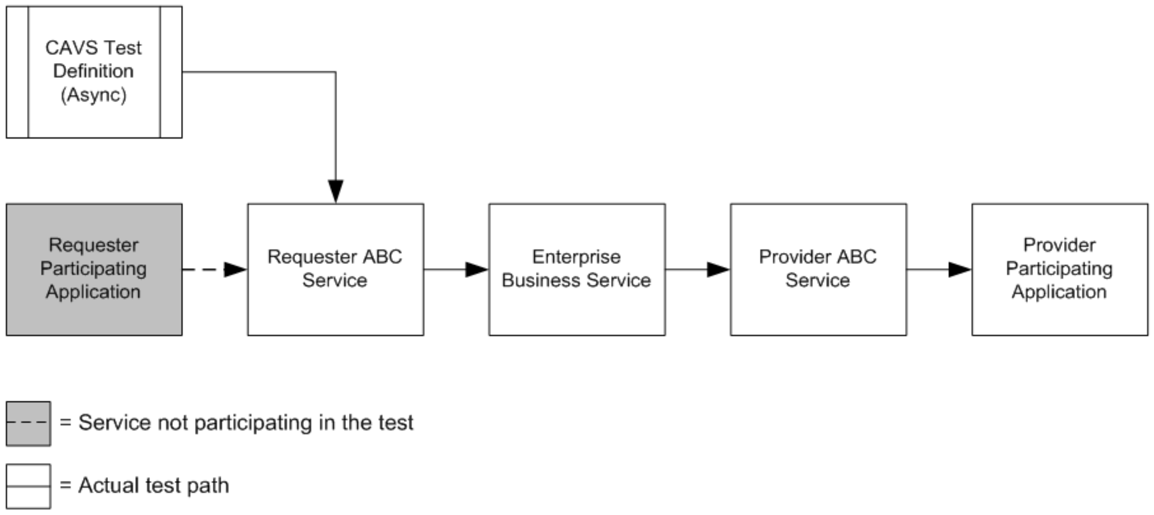
The following diagrams describe CAVS process flows for testing a provider ABC service using an asynchronous (notify) MEP.

These sample flows can be used as the basis for testing other artifacts as well, such as the requester ABC service, enterprise business flow, or the provider service itself.

Asynchronous (Notify) MEP Testing Flow Using a Test Definition

The requester participating application is replaced by the CAVS test definition. The test definition points to the URL of the requester ABC service. It uses a composed request message to invoke the ABC service and does not expect a message in response.

Copyright © 2009, Oracle. All rights reserved.


CAVS Test Definition (Async)
Requester Participating Application
Requester ABC Service
Enterprise Business Service
Provider ABC Service
Provider Participating Application
= Service not participating in the test
= Actual test path

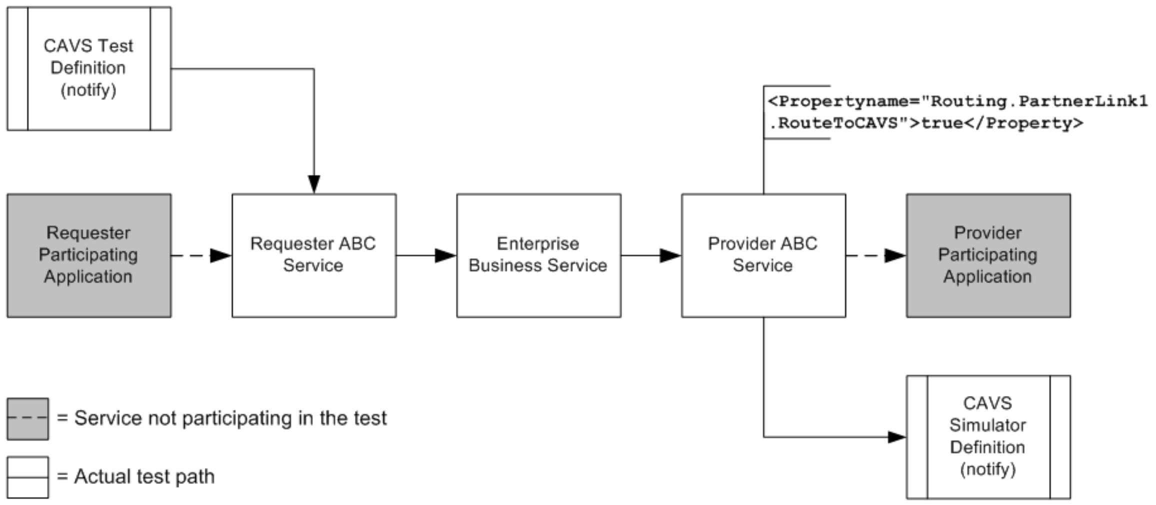
Testing an asynchronous (notify) MEP using a CAVS test definition

Asynchronous (Notify) MEP Testing Flow Using a Test Definition and Simulator Definition

The requester participating application is replaced by the CAVS test definition. The test definition points to the URL of the requester ABC service. It uses a composed request message to invoke the ABC service and does not expect a message in response.

The provider participating application is replaced by the CAVS simulator definition. The provider ABC service is programmed to route to this simulator instead of the provider participating application. The simulator definition contains a predefined expected request message.

The simulator definition performs validations on message input from the provider ABC service.

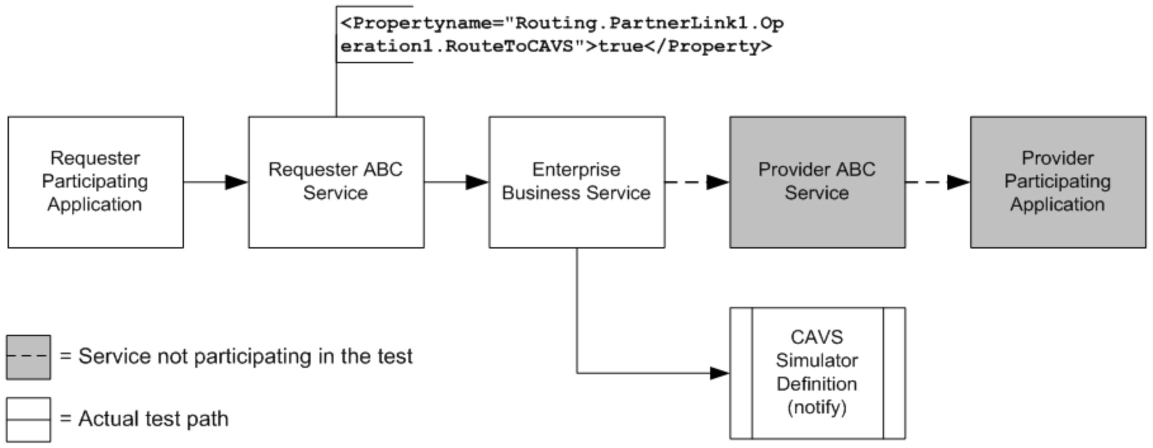

CAVS Test Definition (notify)
<Propertyname="Routing.PartnerLink1.RouteToCAVS">true</Property>
Requester Participating Application
Requester ABC Service
Enterprise Business Service
Provider ABC Service
Provider Participating Application
CAVS Simulator Definition (notify)
= Service not participating in the test
= Actual test path

Testing an asynchronous (notify) MEP using a CAVS test definition and simulator definition

Asynchronous (Notify) MEP Testing Flow Using a Simulator Definition

The provider participating application is replaced by the CAVS simulator definition. The requester ABC service is programmed to route the enterprise business service (EBS) to this simulator instead of the provider participating application. The simulator definition contains a predefined expected request message.

The simulator definition performs validations on message input from the provider ABC service


<Propertyname="Routing.PartnerLink1.Operation1.RouteToCAVS">true</Property>
Requester Participating Application
Requester ABC Service
Enterprise Business Service
Provider ABC Service
Provider Participating Application
CAVS Simulator Definition (notify)
= Service not participating in the test
= Actual test path

Testing an asynchronous (notify) MEP using a CAVS simulator definition

Describing CAVS Process Flows for Testing the Asynchronous Two-Way Message Exchange Pattern

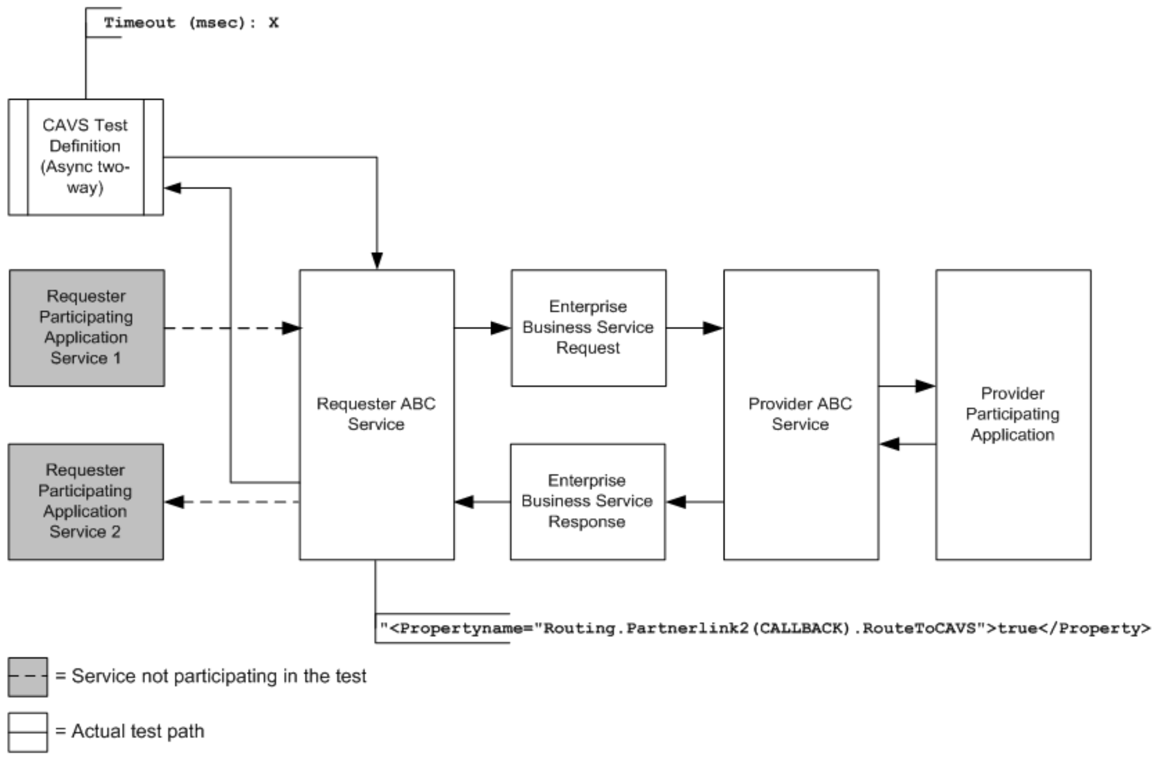
The following diagrams describe CAVS process flows for testing a provider ABC service using an asynchronous two-way MEP.

These sample flows can be used as the basis for testing other artifacts as well, such as the requester ABC service, enterprise business flow, or the provider service itself.

Asynchronous Two-Way MEP Testing Flow Using a Test Definition

The requester participating application is replaced by the CAVS test definition. The test definition points to the URL of the requester ABC service. It uses a composed request message to invoke the ABC service and expects an eventual message in response. The test definition includes a timeout value. If no response message is received within this timeout value, the test definition will experience a timeout failure.

Copyright © 2009, Oracle. All rights reserved.


Timeout (msec): X
CAVS Test Definition (Async two-way)
Requester Participating Application Service 1
Requester Participating Application Service 2
Requester ABC Service
Enterprise Business Service Request
Enterprise Business Service Response
Provider ABC Service
Provider Participating Application
"<Propertyname="Routing.Partnerlink2(CALLBACK).RouteToCAVS">true</Property>
= Service not participating in the test
= Actual test path

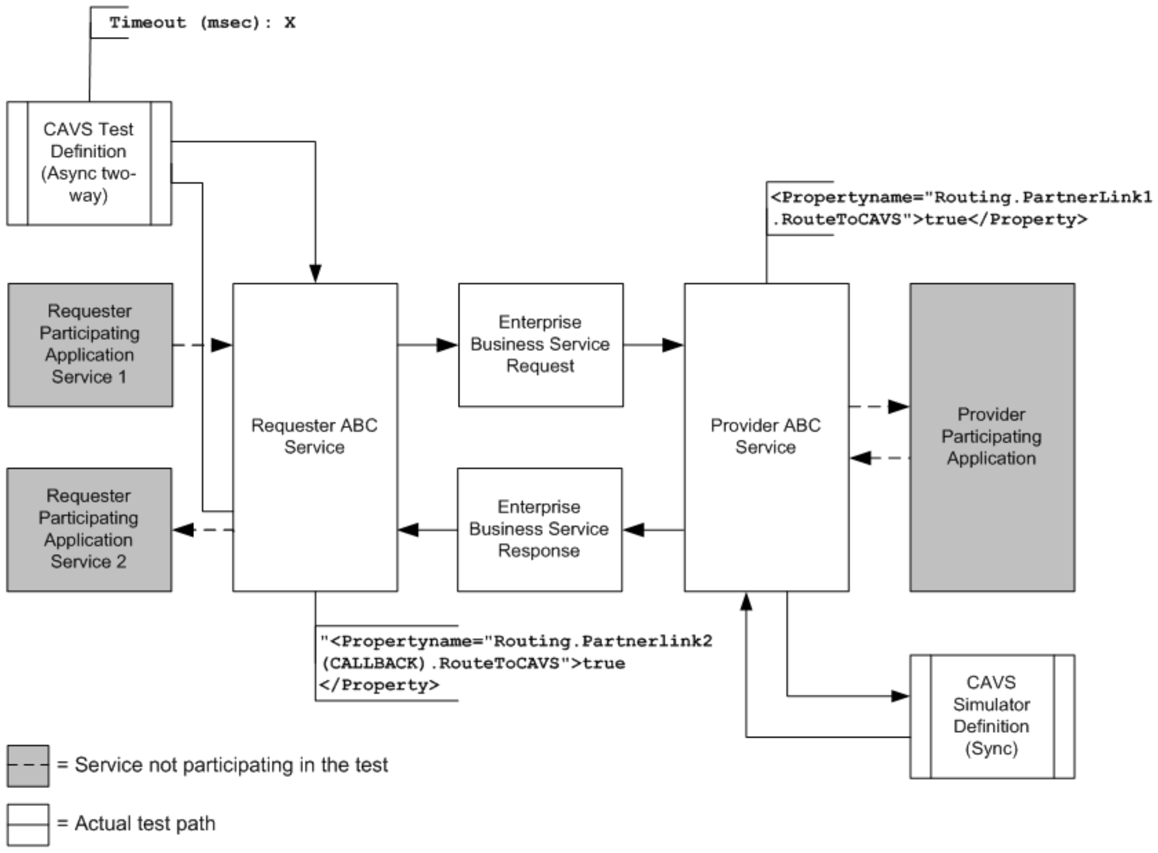
Testing an asynchronous two way MEP using a CAVS test definition

Asynchronous Two-Way MEP Testing Flow Using a Test Definition and Simulator Definition

The requester participating application is replaced by the CAVS test definition. The test definition points to the URL of the requester ABC service. It uses a composed request message to invoke the ABC service and expects an eventual message in response. The test definition includes a timeout value. If no response message is received within this timeout value, the test definition will experience a timeout failure.

The provider participating application is replaced by the CAVS simulator definition. The provider ABC service is programmed to route to this simulator instead of the provider participating application. The simulator definition contains a predefined request and response message pair.

The simulator definition performs validations on message input from the provider ABC service and sends the message back to the provider ABC service. The provider ABC service sends the message back to the test definition, which validates this actual response against its predefined expected response.


Timeout (msec): X
CAVS Test Definition (Async two-way)
Requester Participating Application Service 1
Requester Participating Application Service 2
Requester ABC Service
Enterprise Business Service Request
Enterprise Business Service Response
Provider ABC Service
Provider Participating Application
<Propertyname="Routing.PartnerLink1.RouteToCAVS">true</Property>
"<Propertyname="Routing.Partnerlink2 (CALLBACK).RouteToCAVS">true </Property>
CAVS Simulator Definition (Sync)
= Service not participating in the test
= Actual test path

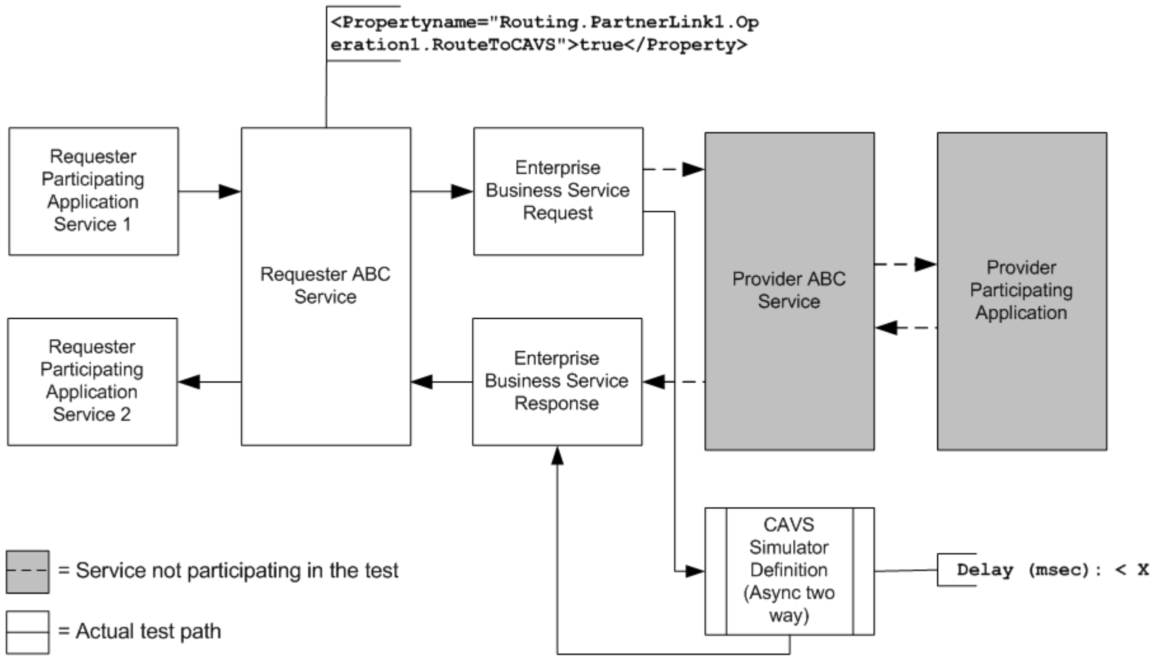
Testing an asynchronous two way MEP using a CAVS test definition and simulator definition

Asynchronous Two-Way MEP Testing Flow Using a Simulator Definition

The provider participating application is replaced by the CAVS simulator definition. The provider ABC service is programmed to route to this simulator instead of the provider participating application. The simulator definition contains a predefined request and response message pair, as well as a user-defined delay value. The simulator definition will delay its response by this amount of time to simulate the asynchronous two-way nature of the provider participating application.

The simulator definition performs validations on message input from the provider ABC service and sends the message back to the provider ABC service. The provider ABC service sends the message back to the requester participating application.

Copyright © 2009, Oracle. All rights reserved.



<Propertyname="Routing.PartnerLink1.Operation1.RouteToCAVS">true</Property>
Requester Participating Application Service 1
Requester ABC Service
Enterprise Business Service Request
Provider ABC Service
Provider Participating Application
Requester Participating Application Service 2
Enterprise Business Service Response
CAVS Simulator Definition (Async two way)
Delay (msec): < X
= Service not participating in the test
= Actual test path

Testing an asynchronous two way MEP using a CAVS simulator definition

Note: When defining the simulator in this case, ensure that the EnvironmentCode field in the enterprise business message (EBM) header of the simulator response message is assigned the value of ***PRODUCTION***. This will ensure that the message continues to reach the requester ABC service. This value assignment is required whenever a simulator calls the response EBS, as in this case.

If instead of a simulator, a provider ABC service calls the response EBS, then the EnvironmentCode in the EBM header will automatically be set to the appropriate value based on the RouteToCAVS property for that callback in the AIAConfigurationProperties.xml file.

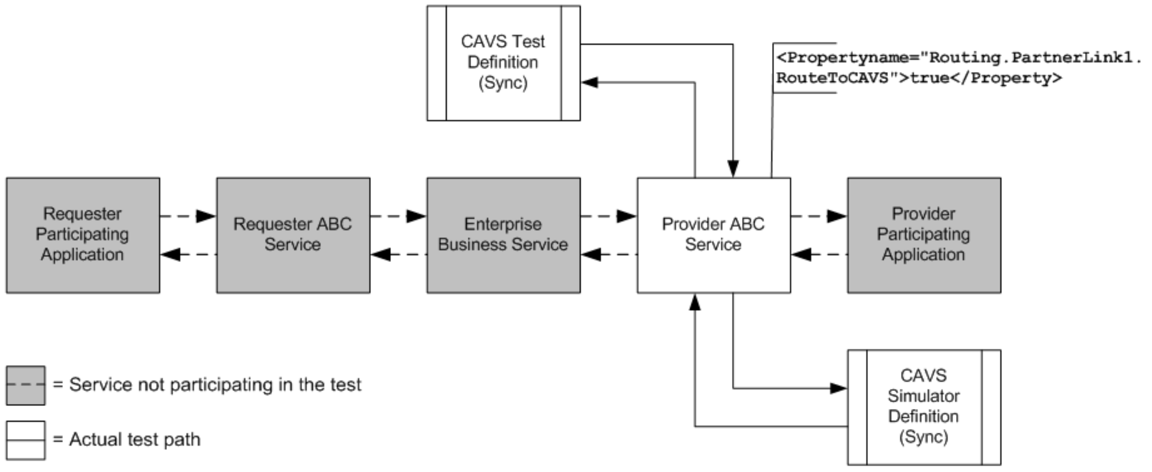
Does the Scenario Need to be Unit or Flow Tested?

This section discusses different configurations for test and simulator definitions to achieve unit and flow tests.

Describing a Unit Test Configuration

This section will use a synchronous provider ABC service as the focus of the test example. However, this test configuration is not specific to message exchange patterns, so it can be applied to asynchronous (notify) and asynchronous two-way components as well.

To unit test a component, place a test definition before the component and a simulator definition after it. This isolates the focus of the test to the single component.

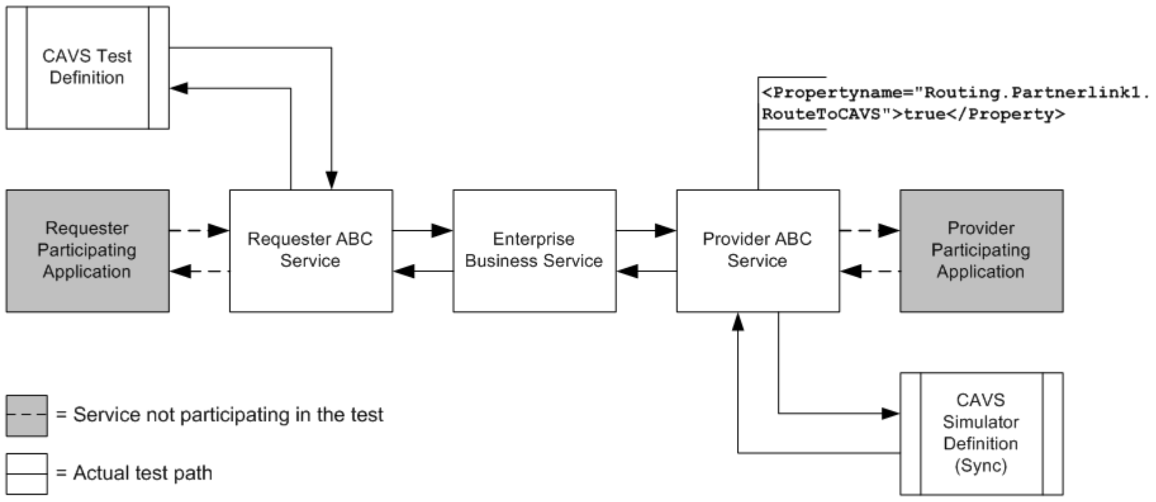

CAVS Test Definition (Sync)
<Propertyname="Routing.PartnerLink1.RouteToCAVS">true</Property>
Requester Participating Application
Requester ABC Service
Enterprise Business Service
Provider ABC Service
Provider Participating Application
CAVS Simulator Definition (Sync)
= Service not participating in the test
= Actual test path

Unit testing a provider ABC service

Describing a Flow Test Configuration

This section will use a synchronous provider ABC service as the focus of the test example. However, this test configuration is not specific to message exchange patterns, so it can be applied to asynchronous (notify) and asynchronous two-way components as well.

Once you have unit tested the components in a scenario, you can flow test the scenario. To flow test a scenario, place a test definition before the requester ABC service at the front of the scenario and a simulator definition after the provider ABC service at the end of the scenario.


CAVS Test Definition
<Propertyname="Routing.Partnerlink1.RouteToCAVS">true</Property>
Requester Participating Application
Requester ABC Service
Enterprise Business Service
Provider ABC Service
Provider Participating Application
CAVS Simulator Definition (Sync)
= Service not participating in the test
= Actual test path

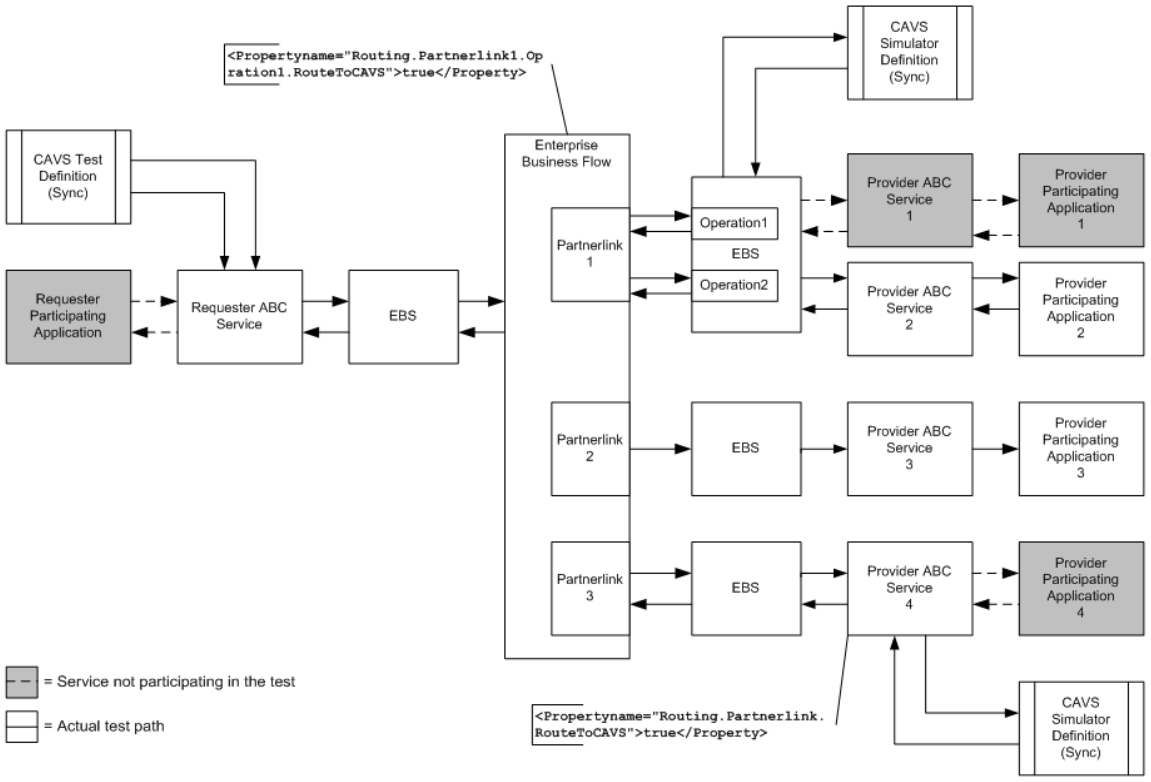
Flow testing a scenario

Describing a Complex Flow Test Configuration

This section will use an EBF as the focus of the test example. However, this test configuration is not specific to EBFs, so it can be applied to any service that conducts chatty conversations.

Copyright © 2009, Oracle. All rights reserved.

You can place a test definition before the requester ABC service at the front of a scenario and enable the EBF to make calls out to simulator definitions whenever required. You can then go on to replace some of the provider applications with simulators at the end of the scenario.


CAVS Simulator Definition (Sync)
<Propertyname="Routing.Partnerlink1.Operation1.RouteToCAVS">true</Property>
CAVS Test Definition (Sync)
Enterprise Business Flow
Provider ABC Service 1
Provider Participating Application 1
Partnerlink 1
Operation1
EBS
Operation2
Requester Participating Application
Requester ABC Service
EBS
Provider ABC Service 2
Provider Participating Application 2
Partnerlink 2
EBS
Provider ABC Service 3
Provider Participating Application 3
Partnerlink 3
EBS
Provider ABC Service 4
Provider Participating Application 4
= Service not participating in the test
= Actual test path
<Propertyname="Routing.Partnerlink.RouteToCAVS">true</Property>
CAVS Simulator Definition (Sync)

Complex flow testing an enterprise business flow

Do I Have the Content I Need to Create the Definitions?

Once you know what you need to test and which CAVS definitions you need to create and assess whether or not you have all of the content you need to create the definitions.

To create your test definitions and simulator definitions, you will primarily need request and response XML text.

If you are creating a Test Definition (or an asynchronous two-way simulator), you will need the endpoint URL of the web service you are testing.

The endpoint URL value can be found in the WSDL of the web service that you want to test.

When the endpoint URL is provided, CAVS will present you with available SOAP actions. Once you select the required SOAP action, CAVS will automatically generate the message stub for the service being called. You can then include data within the XML tags generated.

In either case, you can use the BPEL Console to obtain request and response XML text. Run the processes you are testing at least once with all participating applications and services in place and with the desired results. The XML messages generated by this successful run of the processes will provide your request and response XML for test and simulator definitions. The following section describes how to use the BPEL console to obtain these XML messages.

Note: Obtaining response message XML text is only applicable when testing synchronous and asynchronous two-way processes.

Obtaining Message XML Text from a BPEL Process

To obtain request and response message XML text:

1. Access the BPEL Console for your Oracle AIA implementation.
2. Select the **Instances** tab.
3. In the BPEL Process drop-down list box, select the BPEL process for which you are creating a test definition and click **Go.** BPEL process instances for the selected BPEL process display in the **List of BPEL Process Instances** frame. Sort by ***Last Modified,*** if you want to access the most recent instance.
4. Click the link for the instance you want to use for your request and response XML message text.
5. Click the **Flow** link.

Note: If the instance you selected contains any faults, you may want to consider selecting a different instance. However, if you are trying to test fault messaging in the BPEL, you must select a BPEL process instance that contains the fault.

6. To obtain the Request Message XML text:
   a. Click the receiveInput element to get your test definition request message XML text.
   b. Click the **Copy details to clipboard** link at the bottom of the pop-up box displaying input Variable data.
   c. Open an XML editor and paste the XML text into a blank document.
   d. Remove the opening and closing ***inputVariable*** (***XYZ_InputVariable,*** in the case of a non-BPEL service, such as a participating application service) and ***part*** elements.
   e. Copy and paste the remaining XML text in the default SOAP envelope provided in the **Request Message** field on the Test Definition page or Simulator Definition page. Paste the XML text into the area indicated by the “Paste your SOAP Message Content here” placeholder text.
7. To obtain the Response Message XML text:

Note: Obtaining response message XML text is only applicable when testing a synchronous process.

   a. Click the **r**eply Output element to get your test definition response message XML text.
   b. Click the Copy details to clipboard link at the bottom of the pop-up box displaying output Variable data.
   c. Open an XML editor and paste the XML text into a blank document.
   d. Remove the opening and closing ***inputVariable*** (***XYZ_InputVariable,*** in the case of a non-BPEL service, such as a participating application service) and ***part*** elements.

Copyright © 2009, Oracle. All rights reserved.

e. Copy and paste the remaining XML text in the default SOAP envelope provided in the **Response Message** field on the Test Definition page or Simulator Definition page

Chapter 10: Overview of Defining and Running CAVS Tests Using the CAVS User Interface

This chapter provides overviews of:

- The Composite Application Validation System (CAVS) user interface (UI).
- Executing CAVS definitions using a web service.
- Executing CAVS definitions using ANT.
- The basics of defining and running CAVS tests.

Understanding the CAVS UI

The CAVS enables you to configure test data, schedule tests, execute tests, review test results, and migrate tests using the following UI components.

Test Definitions

A test definition is a configuration of a single execution of the test initiator service. The test definition stores test data and test execution instructions. A test definition can be executed alone, or in a single-threaded batch as a part of a group definition.

You will find that the values you set for a test definition and simulator definition are similar. The test definition differs from the simulator definition in that it is an active participant in the CAVS framework, initiating tests. The test definition carries the following values that are not a part of the simulator definition. These values inform the active state of the test definition:

- Endpoint URL
- SOAP Action

For more information about test definitions, see Creating and Modifying Test Definitions.

If required by the business service pattern of the web service you are testing, you can assign a simulator definition to the test definition.

Simulator Definitions

A simulator definition is a configuration of a single execution of response simulator service. The simulator definition simulates a web service and receives data from the tested web service and returns previously defined data so that the tested web service can continue processing.

You will find that the values you set for a simulator definition and test definition are similar. The simulator definition differs from the test definition in that it is a passive participant in the CAVS framework, awaiting initiation.

The simulator definition carries the following additional XPath attributes that are not a part of the test definition. These values participate in simulator definition request matching:

- Is Node Key
- Key Node Value

For more information about simulator definitions, see Creating and Modifying Simulator Definitions.

The success of the test is verified based on the simulator definition's previously defined data being accurately returned and matched to the expected response results defined in the test definition.

Group Definitions

A group definition is a configuration of a single execution of one or more test definitions in a single-threaded batch.

Test Instances

A test instance captures the details of the execution of a test definition.

Simulator Instances

A simulator instance captures the details of a simulator definition's behavior during the execution of a test definition with which it is associated.

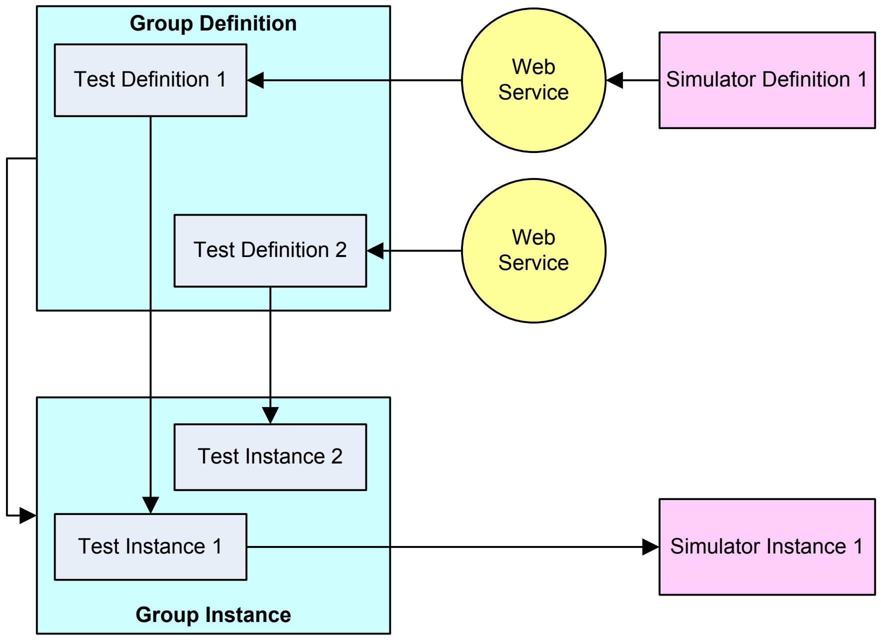
Group Instances

A group instance captures the details of the execution of a group definition.

Copyright © 2009, Oracle. All rights reserved.

Component Overview

This diagram provides a high-level overview of the relationships among the CAVS components discussed in this section.



Overview of CAVS component relationships

Executing CAVS Definitions Using a Web Service

The CAVS provides a web service that enables you to execute test definitions and test group definitions without the use of the CAVS user interface.

For example, this web service is utilized by the Diagnostics Framework to provide some of its test data.

For more information about the Diagnostics Framework, see The Diagnostics Framework.

You can call this CAVS web service from any system: http://<hostname>:<port>/AIAValidationSystemAPIService/AIAValidationSystemAPIServiceSoapHttpPort.

This web service provides two operations:

- executeDefinition

  Executes a given test definition ID.

- executeGroupDefinition

  Executes a given test group definition ID.

Copyright © 2009, Oracle. All rights reserved.

Typically, these operations can be consumed by third-party testing tools or other systems to execute test definitions and test group definitions whenever desired, without the use of the CAVS user interface.

The WSDL that defines the service contract is: http://<hostname>:<port>/AIAValidationSystemAPIService/AIAValidationSystemAPIServiceSoapHttpPort?wsdl.

Executing CAVS Definitions Using ANT

CAVS provides an ANT script that enables you to execute test definitions without the use of the CAVS user interface. This functionality is useful when trying to automate test execution as a part of automated deployment processes.

CAVS test definitions can also be executed as web services.

For more information about the CAVS web service, see Executing CAVS Definitions Using a Web Service.

To execute a CAVS definition using ANT:

1. Define your test definition in the AIA Console. In this example, the test definition ID is ***601***.
2. Navigate to AIAHOME/Infrastructure/CAVS.
3. Run `source AIAHOME/bin/aiaenv.sh`.
4. Run `ant -f AIACAVSInvoke.xml -Did=601`.

Overview of Defining and Running CAVS Tests

This high-level procedure provides the steps involved in defining and running tests using the CAVS UI.

1. Assess your test requirements and gather required content.

For more information, see Chapter 9: Preparing to Use the CAVS.

2. If your test requires a test definition, access the Create Test page to create your test definition.

For more information, see Creating and Modifying Test Definitions.

3. If your test requires a simulator definition, access the Create Simulator page to create your simulator definition.

Copyright © 2009, Oracle. All rights reserved.

For more information, see Creating and Modifying Simulator Definitions.

4. If your test requires a test definition and simulator definition, you can link their definitions on either Modify Test Definition page or Modify Simulator Definition page.

5. If your test requires multiple (single-threaded) executions of the same or different test definitions, create a group definition on the Create Group Definition page.

For more information, see Creating Group Definitions.

6. If your test or group definition utilizes a simulator definition, you must set the application business connector (ABC) service being tested to route to the response simulator. To do this:

    a. Access the Routing Setup page.

    b. Create a routing setup ID for the invoking service being tested.

    c. Associate this routing setup ID with the test definition being used to test your scenario.

For more information about routing setup IDs, see Chapter 15: Defining CAVS Routing Setup IDs.

    d. Alternatively, you can quickly set up a routing configuration on the Configurations page.

For more information about routing configurations, see Chapter 7: Loading Oracle AIA Configuration File Updates.

7. To run a single test, access the test definition on the Definitions page or Modify Test Definitions page to run the test. On the Definitions page, you may choose to schedule the running of the test.

For more information, see Searching for and Working with Test and Simulator Definitions.

To run a group test, access the Group Definition page or Group Definition Detail page to run the group test. On the Group Definitions page, you may choose to schedule the running of the test group.

For more information, see Working with Group Definitions.

8. Once an individual test has been executed, view test results generated for the test instance on the Test Instances Detail page. If the test definition includes an associated simulator definition, view the simulator instance on the Simulator Instances Detail page.

For more information, see Chapter 17: Working with Test and Simulator Instances.

Once a group of tests has been executed, view the test results generated for the group instance on the Group Instance Detail page. If the test group was scheduled, you can view schedule details and test results on the Test Executive Summary page.

For more information, see Working with Group Instances and Viewing Scheduled Test Group Results.

9. Once testing is complete for a test that involved a simulator definition for which you defined a routing configuration on the Configurations page, be sure to reset the ABC service tested to return to routing to its usual production destination and no longer route to the response simulator. To do this:

   a. Access the Configurations page.

   b. Clear the Route To CAVS option.

   c. Click Reload.

For more information about the Configurations page, see Chapter 7: Loading Oracle AIA Configuration File Updates.

Copyright © 2009, Oracle. All rights reserved.

Chapter 11: Creating and Modifying Test Definitions

This chapter discusses how to:

- Create and modify test definitions.
- Provide multiple request and response message sets in a single test definition.

Creating and Modifying Test Definitions

A test definition is a configuration of a single execution of the test initiator service. The test definition stores test data and test execution instructions. A test definition can be executed alone, or in a single-threaded batch as a part of a group definition.

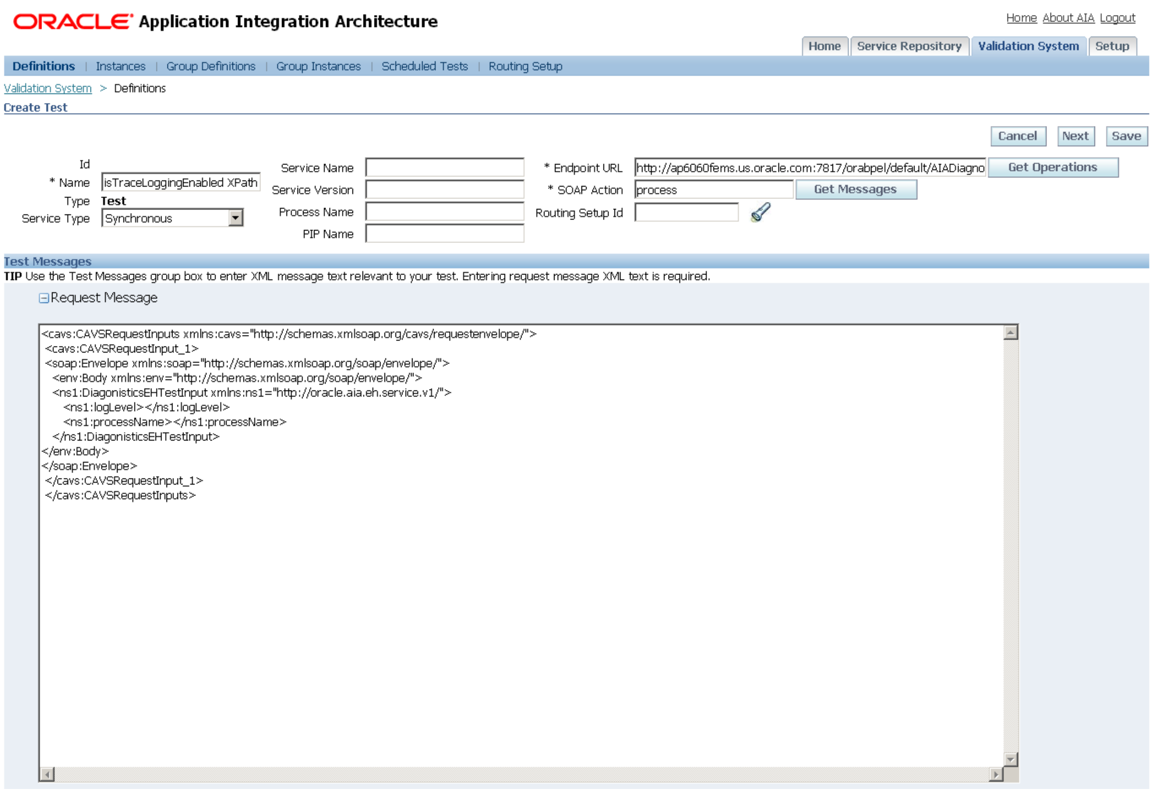
Pages Used to Create and Modify Test Definitions

Page Name	Navigation	Usage
Create Test	Access the Oracle AIA Console. Select the Validation System tab. Click the Definitions link. Click Create Test.	Create test definitions.
View Test Definition	• Click the Id link for a locked test definition on the Definitions page. • Select ***Lock*** in the Action drop-down list box on the Modify Test Definition page and click Go. • Click a Definition Id link for a locked test definition on the Instances page. • Click a Definition Id link for a locked test definition on the Test Instances Detail page.	Access a read-only view of the test definition.
Modify Test Definition	• Enter required values on the Create Test page and click Next. • Click a Definition Id link for an unlocked test definition in the Search Results grid on the Definitions page. • Click a Definition Id link for an unlocked test definition on the	Modify an existing test definition. Execute and manage existing test definitions.

Page Name	Navigation	Usage
	Instances page. • Click a Definition Id link for an unlocked test definition on the Test Instances Detail page.	
Search Definitions – Simulator	Click Assign in the Simulator Definitions grid on the Modify Test Definitions page.	Search for simulator definitions to assign to a test definition. This assignment facilitates linking of test instances to simulator instances.

Creating a Test Definition

Access the Create Test page.



Id	
* Name	isTraceLoggingEnabled XPath
Type	Test
Service Type	Synchronous
Service Name	
Service Version	
Process Name	
PIP Name	
* Endpoint URL	http://ap6060fems.us.oracle.com:7817/orabpel/default/AIADiagno
* SOAP Action	process
Routing Setup Id	

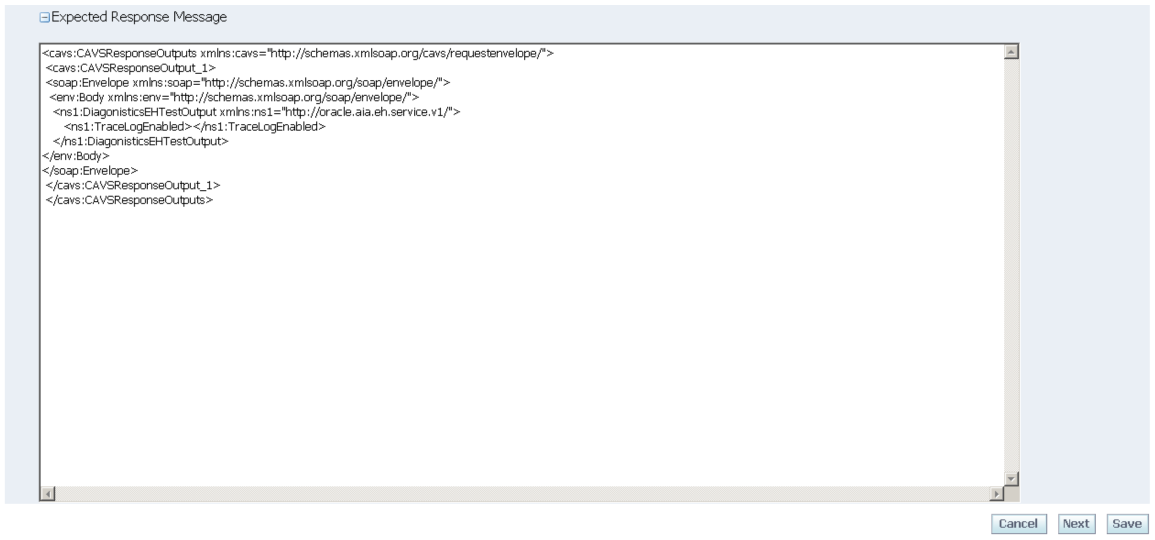
TIP Use the Test Messages group box to enter XML message text relevant to your test. Entering request message XML text is required.

Request Message

```
<cavs:CAVSRequestInputs xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
 <cavs:CAVSRequestInput_1>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Body xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <ns1:DiagonisticsEHTestInput xmlns:ns1="http://oracle.aia.eh.service.v1/">
      <ns1:logLevel></ns1:logLevel>
      <ns1:processName></ns1:processName>
   </ns1:DiagonisticsEHTestInput>
</env:Body>
</soap:Envelope>
 </cavs:CAVSRequestInput_1>
 </cavs:CAVSRequestInputs>
```

Create Test page (1 of 2)

Copyright © 2009, Oracle. All rights reserved.



```
<cavs:CAVSResponseOutputs xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
 <cavs:CAVSResponseOutput_1>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Body xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <ns1:DiagonisticsEHTestOutput xmlns:ns1="http://oracle.aia.eh.service.v1/">
     <ns1:TraceLogEnabled></ns1:TraceLogEnabled>
   </ns1:DiagonisticsEHTestOutput>
</env:Body>
</soap:Envelope>
 </cavs:CAVSResponseOutput_1>
 </cavs:CAVSResponseOutputs>
```

Create Test page (2 of 2)

Id	Upon saving the test definition, displays a unique key identifier that is assigned to the test definition.
Name	Enter a descriptive name that you want to use for the test definition.
Type	Displays the type of definition you chose to create. On the Create Test page, this value will always be set to ***Test.***
Service Type	Select the business service pattern of the web service that you want to test using the test definition. ***Synchronous*** (request-and-reply) ***Notify*** (asynchronous request-only) ***Asynchronous two way***
Service Name	Enter the name of the web service that you want to test using the test definition. This is the name of the web service being called by the URL provided in the **Endpoint URL** field.
Service Version	Enter the version of the web service that you want to test using the test definition. This is the version of the web service being called by the URL provided in the **Endpoint URL** field.
Process Name	Enter the name of the process that includes the web service that you want to test using the test definition.
PIP Name (Process Integration Pack name)	Enter the name of the PIP that includes the web service that you want to test using the test definition.
Endpoint URL	Enter the URL of the web service that you want to test using the test definition. The endpoint URL value can be found in the WSDL of the web service that you want to test.
Get Operations	Click to display the list of operations supported by the WSDL associated with the **Endpoint URL** value you provided. Supported operations display in the Select WSDL Operations window.

	Select the operation that you want to test using the test definition. The selected operation displays in the **SOAP Action** field.
SOAP Action	If you clicked **Get Operations** to select an operation in the Select WSDL Operations window, selected operation displays here. Alternatively, you can manually enter the operation called by the web service that you want to test using the test definition. The value you enter must match an action provided in the WSDL of the web service that you want to test.
Get Messages	Click to generate a request stub message for the operation specified in the **SOAP Action** field. For test definitions with the **Service Type** field set to ***Synchronous***, the response stub message will also be generated.
Routing Setup Id	Select a routing configuration that you want to use for the test.

For more information about routing configurations, see Chapter 15: Defining CAVS Routing Setup IDs.

Test Messages

Use the **Test Messages** group box to enter request and response XML message text. By default, SOAP envelope XML text is provided in these fields. You can use the **Get Messages** button to generate request and response stub messages based on selected endpoint URL and operation values. Alternatively, you can paste XML text within this default SOAP envelope or paste your own XML text already enclosed in an envelope into these fields.

For more information about obtaining request and response XML message text, see Obtaining Message XML Text from a BPEL Process.
For more information about how to create test request and response messages that hold multiple sets of test data in a single definition, see Providing Multiple Request and Response Message Sets in a Single Test Definition.

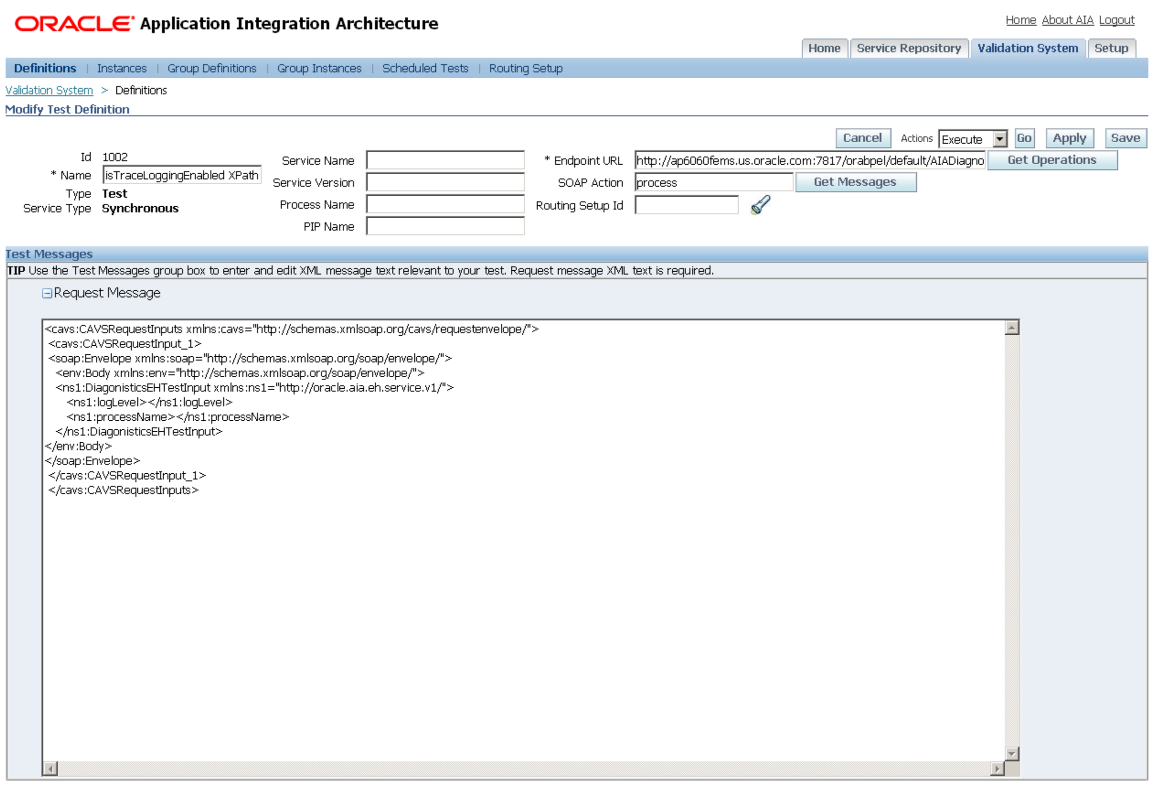
Request Message	Entering request message XML text for a test definition is required, whether the **Service Type** field value is set to ***Synchronous, Notify,*** or ***Asynchronous two way.*** When you first access the Create Test page, the **Request Message** text box is populated with a SOAP stub message. You can use the **Get Messages** button to generate a request stub message based on selected endpoint URL and operation values. If you are manually entering your request message, the “Paste your SOAP Message Content here” text in the stub message indicates where you should paste your actual request message text. This request message should mimic the XML message text sent by the service that normally initiates the service.

Copyright © 2009, Oracle. All rights reserved.

Expected Response Message	The ability to enter response message XML text is available when the **Service Type** field value is set to ***Synchronous*** or ***Asynchronous two way.*** When you first access the Create Test page, the **Expected Response Message** text box is populated with a SOAP stub message. For test definitions with the **Service Type** field set to ***Synchronous***, the response message stub will have been generated when you clicked **Get Messages** during request message generation. If you are manually entering your request message, the “Paste your SOAP Message Content here” text in the stub message indicates where you should paste your actual response message text. Enter a response message that is the expected response message XML. This facilitates the generation of XPath values, which are used to validate the actual response message returned in the test. You may also choose to manually enter or modify the XPath values directly on the Modify Test Definition page. If you are manually entering XPath values, you do not need to enter response message XML text. When you enter response message XML text on this page, you can click **Generate Xpath** on the Modify Test Definition page to generate the XPath values that will be used to validate the expected response message you entered on this page against the actual response returned by the test. If the **Service Type** field value is set to ***Synchronous*** or ***Asynchronous two way,*** you may choose to not enter response message XML text in this field. You do not need to enter response message XML if you are manually entering XPath values directly on the Modify Test Definition or if the test you are running does not require validation of the response message. For example, your test may be focused on just populating data. The **Expected Response Message** text box is unavailable when the **Service Type** field value is set to ***Notify.***
Cancel	Click to exit the page and return to the Definitions page.
Next	Click to save entries on the Create Test page and go to the Modify Test Definition page, where you can further edit your test definition, generate XPaths, and execute the test.
Save	Click to save entries on the Create Test page and return to the Definitions page.

Modifying a Test Definition

Access the Modify Test Definition page.


ORACLE Application Integration Architecture
Home About AIA Logout
Home Service Repository Validation System Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup
Validation System > Definitions
Cancel Actions Execute Go Apply Save
Get Operations
Get Messages
Request Message

Modify Test Definition

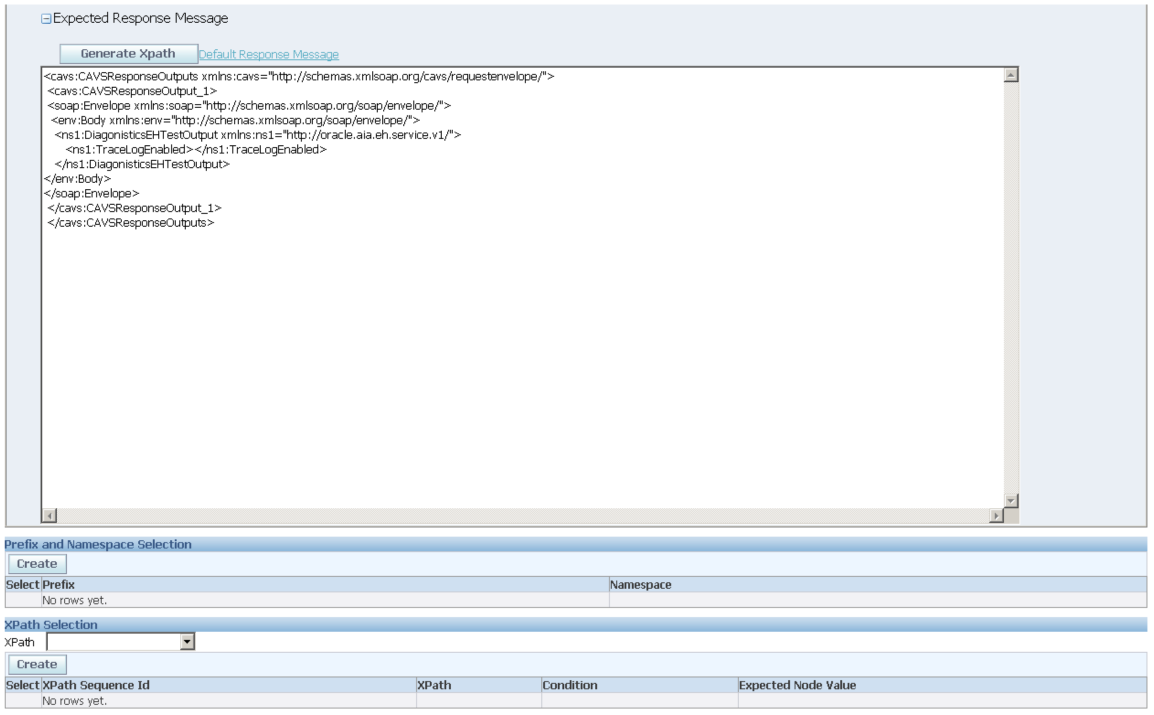
Id	1002	Service Name		* Endpoint URL	http://ap6060fems.us.oracle.com:7817/orabpel/default/AIADiagno
* Name	isTraceLoggingEnabled XPath	Service Version		SOAP Action	process
Type	Test	Process Name		Routing Setup Id	
Service Type	Synchronous	PIP Name			

Test Messages

TIP Use the Test Messages group box to enter and edit XML message text relevant to your test. Request message XML text is required.

```
<cavs:CAVSRequestInputs xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
 <cavs:CAVSRequestInput_1>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Body xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <ns1:DiagonisticsEHTestInput xmlns:ns1="http://oracle.aia.eh.service.v1/">
      <ns1:logLevel></ns1:logLevel>
      <ns1:processName></ns1:processName>
   </ns1:DiagonisticsEHTestInput>
</env:Body>
</soap:Envelope>
 </cavs:CAVSRequestInput_1>
 </cavs:CAVSRequestInputs>
```

Modify Test Definition page (1 of 3)



```
<cavs:CAVSResponseOutputs xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
 <cavs:CAVSResponseOutput_1>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Body xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <ns1:DiagonisticsEHTestOutput xmlns:ns1="http://oracle.aia.eh.service.v1/">
      <ns1:TraceLogEnabled></ns1:TraceLogEnabled>
   </ns1:DiagonisticsEHTestOutput>
</env:Body>
</soap:Envelope>
 </cavs:CAVSResponseOutput_1>
 </cavs:CAVSResponseOutputs>
```

Prefix and Namespace Selection

Select	Prefix	Namespace
	No rows yet.	

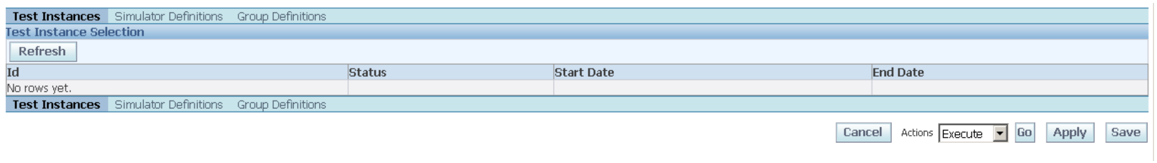
XPath Selection

XPath

Select	XPath Sequence Id	XPath	Condition	Expected Node Value
	No rows yet.			

Modify Test Definition page (2 of 3)

Copyright © 2009, Oracle. All rights reserved.



Id	Status	Start Date	End Date
No rows yet.			

Modify Test Definition page (3 of 3)

The page displays values you have defined for the test definition. You can modify the values in editable fields.

For more information, see Creating a Test Definition.

Cancel	Click to discard any updates you have made and return to the Definitions page.
Actions	Select the action you want to take with the test definition. ***Execute:*** Select and click **Go** to execute the test definition. The status of the test execution appears at the top of the page. When a test definition has successfully executed, you can view details of the test instance on the Test Instance Details page. **For more information** about the Test Instance Details page, see Chapter 17: Working with Test and Simulator Instances. ***Lock:*** Select and click **Go** to lock the test definition and view the test definition on the View Test Definition page. A locked definition cannot be edited. ***Duplicate:*** Select and click **Go** to duplicate the test definition. The duplicate definition is created using the exact values of the original, with the exception of being given a unique Id value.
Go	Make a selection in the **Action** drop-down list box and click **Go** to execute the action.
Apply	Click to apply and save any changes you have made to values on the page.
Save	Click to save entries on the page and go to the Definitions page. **For more information** about the Definitions page, see Searching for and Working with Test and Simulator Definitions.
Time-out (msec) (in milliseconds)	This field displays only for a test definition with a **Service Type** value of ***Asynchronous two way.*** Enter the number of milliseconds that you want the test definition to remain available for the asynchronous reply before timing out. If this length of time passes before the

asynchronous response is returned, a failure will be issued.

If your test includes a simulator definition, the Time-out (msec) value you provide here must be greater than the Delay (msec) value defined on the simulator definition.

For more information about the Delay (msec) field, see Creating a Simulator Definition.

Test Messages

Use the **Test Messages** group box to generate XPath values based on provided response XML message text. By default, SOAP envelope XML text is provided in these fields. You can use the **Get Messages** button to generate request and response stub messages based on selected endpoint URL and operation values. Alternatively, you can paste XML text within this default SOAP envelope, or paste your own XML text already enclosed in an envelope into these fields.

For more information about obtaining request and response XML message text, see Obtaining Message XML Text from a BPEL Process.
For more information about how to create test request and response messages that hold multiple sets of test data in a single definition, see Providing Multiple Request and Response Message Sets in a Single Test Definition.

Request Message

If request message XML text was entered on the Create Test page, it is accessible and editable on this page.

Entering request message XML text for a test definition is required, whether the **Service Type** field value is set to ***Synchronous, Notify,*** or ***Asynchronous two way***.

You can use the **Get Messages** button to generate a request stub message based on selected endpoint URL and operation values.

If you are manually entering your request message, the “Paste your SOAP Message Content here” text in the stub message indicates where you should paste your actual request message text. This request message should mimic the XML message text sent by the service that normally initiates the service.

Request CorrelationId Message

This field only displays for a test definition with the **Service Type** field value set to ***Asynchronous two way.*** For this service type, entering a correlation ID value ensures that when the asynchronous response is actually received, the CAVS is able to correlate it to the correct request.

If your request message is an enterprise business message (EBM), leave this field blank, as the EBM header ID is automatically used as the correlation ID. In this case, because the EBM header ID is used as the correlation ID, do not use it as a key column in the simulator definition, if applicable.

If your request message is not an EBM, you must enter a correlation ID value. This correlation must be based on a unique key of the message. For example, CreateOrder can use

Copyright © 2009, Oracle. All rights reserved.

Order ID as the correlation ID.

Click **Lookup** to access the Choose Request Correlation Id page, where you can select a correlation ID from Xpath variables available in the message.

Expected Response Message

The ability to enter response message XML text is available when the **Service Type** field value is set to ***Synchronous*** or ***Asynchronous two way.***

If expected response message XML test was entered on the Create Test page, it is accessible and editable on this page.

You can manually enter the response message text on this page, or for test definitions with the **Service Type** field set to ***Synchronous***, you can use the **Get Messages** button to generate a response stub message based on selected endpoint URL and operation values.

Entering the expected response message XML facilitates the generation of XPath values, which are used to validate the actual response message returned in the test. You may also choose to manually enter or modify the XPath values directly on the Modify Test Definition page. If you are manually entering XPath values, you do not need to enter response message XML text.

When you enter response message XML text on this page, you can click **Generate Xpath** on the Modify Test Definition page to generate the XPath values that will be used to validate the expected response message you entered on this page against the actual response returned by the test.

If the **Service Type** field value is set to ***Synchronous*** or ***Asynchronous two way,*** you may choose to not enter response message XML text in this field. You do not need to enter response message XML if you are manually entering XPath values directly on the Modify Test Definition or if the test you are running does not require validation of the response message. For example, your test may be focused on just populating data.

The **Expected Response Message** text box is unavailable when the **Service Type** field value is set to ***Notify.*** In this case, a response message is not a test requirement.

Generate Xpath

Click to generate namespace and XPath values based on available **Endpoint URL** and **Response Message** values.

Note: Once you have generated XPath values, consider deleting any rows that will not be used in the testing effort.

The **Generate Xpath** button is unavailable when the **Service Type** field value is set to ***Notify.*** In this case, a response message is not a test requirement.

Response Message

This field only displays for a test definition with the **Service Type** field value set to ***Asynchronous two way.*** For this

Correlation ID	service type, entering a correlation ID value ensures that when the asynchronous response is actually received, the CAVS is able to correlate it to the correct request. If your response message is an EBM, leave this field blank, as the EBM header ID is automatically used as the correlation ID. In this case, because the EBM header ID is used as the correlation ID, do not use it as a key column in the simulator definition, if applicable. If your response message is not an EBM, you must enter a correlation ID value. This correlation must be based on a unique key of the message. For example, CreateOrder can use Order ID as the correlation ID. Click **Lookup** to access the Choose Response Correlation Id page, where you can select a correlation ID from Xpath variables available in the message.

Prefix and Namespace Selection

Use the **Prefix and Namespace Selection** grid to define namespace data that will be used in the XPath values defined in the **XPath Selection** grid.

Delete	Select the **Select** check box for one or more namespace rows and click **Delete** to execute the deletion. This button only appears when namespace rows are present.
Create	Click to manually add and populate a namespace row.
Prefix	Prefix that should be used for the namespace.
Namespace	Namespace to be used in the XPath data for the test definition.

XPath Selection

Use the **XPath Selection** grid to work with XPath values that are used to compare the actual response message returned in the test to the expected response message defined in the **Response Message** text box on this page. The values in this grid use the namespace values set in the **Prefix and Namespace Selection** grid.

A common adjustment you will likely need to make to XPath conditions and expected node values in this grid is to genericize certain specific values, such as enterprise business message (EBM) IDs. For example, an EBM ID is unique for each transaction, so your test definition will likely not want to specify a particular EBM ID as response criteria. Instead, you may want to genericize the criteria to just verify that the EBM ID is a number greater than zero or use the ***Is Valid*** condition value.

Note: If you are entering XPath values manually, it is important to maintain correlations with the values entered in the **Prefix and Namespace Selection** grid. Each XPath node must have a prefix (namespace alias) that has been defined in the **Prefix and Namespace Selection** grid, unless it is an XPath expression.

The **XPath Selection** grid is unavailable when the **Service Type** field value is set to ***Notify.*** In this case, a response message is not a test requirement.

Xpath	When working with a test definition that contains multiple request and response data sets, use the **Xpath** drop-down list

Copyright © 2009, Oracle. All rights reserved.

	box to select the data set you want to use to run the test.
	For more information about providing multiple data sets in a test definition, see Providing Multiple Request and Response Message Sets in a Single Test Definition.
Delete	Select the **Select** check box for one or more XPath rows and click **Delete** to execute the deletion. This button only appears when XPath rows are present.
Create	Click to manually add and populate an XPath row.
XPath Sequence Id	Indicates the sequence of the XPath expressions. This value is required. This value is read-only when it has been generated using **Generate Xpath**.
Xpath	XPath data to be used in the test definition. These values can include Xpath nodes and expressions. This value is read-only when it has been generated using the **Generate Xpath** button.
Condition	***Is Valid:*** The value provided in the XPath field is valid and no Expected Node Value is supplied. ***Equals To:*** The value provided in the XPath field is valid and an Expected Node Value is supplied. ***Not Equal To*** ***Less Than*** ***Greater Than*** ***Less Than Equal*** ***Greater Than Equal*** ***Not Null***
Expected Node Value	The value expected in the response XML message. When you use the **Generate Xpath** button to generate XPath data, this value may be populated, but can be modified as necessary. The **Condition** field value is used to qualify this value.

Test Instance Selection

Click the **Test Instances** link to display the **Test Instance Selection** grid, which displays information about test instances generated using the test definition.

Id	Click to access the test instance on the Test Instance Detail page.
	For more information about the Test Instance Detail page, see Viewing Test Instance Details.

Linked Simulator Definition Selection

Click the **Simulator Definitions** link to display the **Linked Simulator Definition Selection** grid, which displays information about simulator definitions that are linked to the selected test definition.

Unassign	Select the **Select** check box for one or more simulator definition rows that you want to disassociate with the test definition. Click **Unassign** to execute the disassociation.
Assign	Click to access the Search Definitions - Simulator page, where you can search for a simulator definition that you want to assign to the test definition. Making this assignment facilitates reporting. Once the test definition runs and generates a test instance, all simulator instances generated by the simulator definition associated with the test definition will automatically be linked to the test instance. Once you have assigned a simulator definition using the Search Definitions - Simulator page, the Modify Test Definition page appears, and displays the selected simulator definition.
Refresh	Click to refresh the Modify Test Definition page.
Simulator Definition Id	Click for an unlocked simulator definition to access the Modify Simulator Definition page. Click for a locked simulator definition to access the View Simulator Definition page.

Group Definition Selection

Click the **Group Definitions** link to display the **Group Definition Selection** grid, which displays information about group definitions that include the test definition.

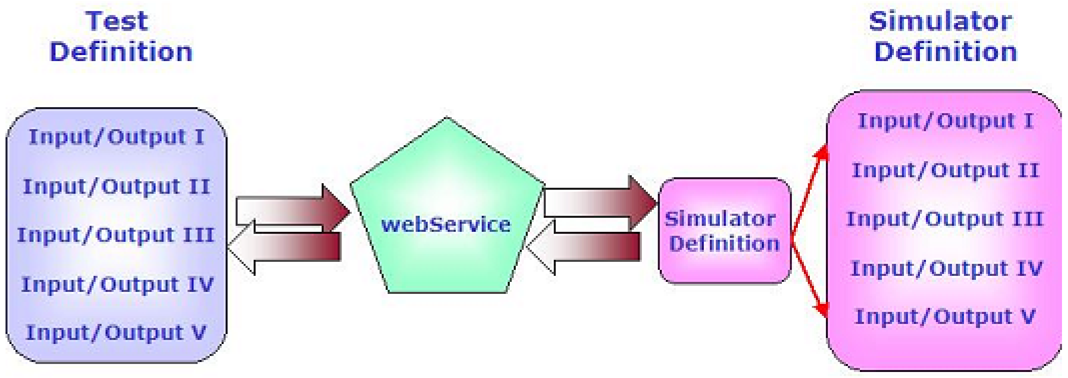
Definition Sequence Id	Displays the sequence in which the test definition is initiated by the group definition.
Group Definition Id	Click to access the group definition on the Group Definition Detail page. **For more information** about the Group Definition Detail page, see Chapter 14: Working with Group Definitions.
Name	Displays the descriptive name assigned to the group definition.

Providing Multiple Request and Response Message Sets in a Single Test Definition

You can create a test definition that contains multiple pairs of request and response message data. This means that test definitions only need to be created per usage requirements, not per test data requirements.

Copyright © 2009, Oracle. All rights reserved.

For example, if you want to test a process against five sets of test data, you can create a single test definition to test the process and include in it all five sets of test data against which you want the process to operate. This is as opposed to creating five separate test definitions, one per combination of process and set of test data.



Providing multiple request and response message sets in a single test definition

When multiple sets of test data are included in a test definition, each set will be executed in sequence. Separate test instances will be generated for each set of data. Test instances will reflect the success or failure of each segment of the test run using each set of test data.

Request Message Format

Use the following format to include multiple sets of request data in the test definition.

The `CAVSRequestInputs` and `CAVSR`equestInput_1 envelope are autogenerated. Use copy and paste commands to create more sets; `CAVSRequestInput_2` and `CAVSRequestInput_3`, for example.

```
<cavs:CAVSRequestInputs
xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
<cavs:CAVSRequestInput_1>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body xmlns:ns1="http://xmlns.oracle.com/SimpleProcess">
   <ns1:SimpleProcessProcessRequest>
…
   </ns1:SimpleProcessProcessRequest>
   </soap:Body>
</soap:Envelope>
</cavs:CAVSRequestInput_1>
```

```
<cavs:CAVSRequestInput_2>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body xmlns:ns1="http://xmlns.oracle.com/SimpleProcess">
   <ns1:SimpleProcessProcessRequest>
   …
   </ns1:SimpleProcessProcessRequest>
   </soap:Body>
</soap:Envelope>
</cavs:CAVSRequestInput_2>

</cavs:CAVSRequestInputs>
```

Response Message Format

Use the following format to include multiple sets of response data in the test definition.

```
<cavs:CAVSResponseOutput_1>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body xmlns:ns1="http://xmlns.oracle.com/SimpleProcess">
   <ns1:SimpleProcessProcessResponse>
       …
   </ns1:SimpleProcessProcessResponse>
   </soap:Body>
</soap:Envelope>
</cavs:CAVSResponseOutput_1>

<cavs:CAVSResponseOutput_2>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body xmlns:ns1="http://xmlns.oracle.com/SimpleProcess">
   <ns1:SimpleProcessProcessResponse>
       …
   </ns1:SimpleProcessProcessResponse>
   </soap:Body>
</soap:Envelope>
</cavs:CAVSResponseOutput_2>
</cavs:CAVSResponseOutputs>
```

After entering request and response data sets and clicking the Generate Xpath button on the Modify Test Definition page, the XPath Selection grid provides access to the Please select an Xpath drop-down list box, where you can select the set of test data you want to use to run the test.

For more information about the Modify Test Definition page, see Modifying a Test Definition.

If your testing scenario includes simulator definitions, you can likewise create simulator definitions that contain multiple request and response message sets that work with the sets defined in your test definition.

For more information, see Providing Multiple Request and Response Message Sets in a Single Simulator Definition.

Copyright © 2009, Oracle. All rights reserved.

Chapter 12: Creating and Modifying Simulator Definitions

This chapter discusses how to:

- Create and modify simulator definitions.
- Provide multiple request and response message sets in a single simulator definition.
- Create a simulator definition that supports chatty services.
- Send dynamic responses in a simulator response.

Creating and Modifying Simulator Definitions

A simulator definition is created by the Composite Application Validation System (CAVS) user to simulate a particular service in an Oracle Application Integration Architecture (AIA) integration or a participating application. A simulator receives data from the tested web service and returns predefined data so that the tested web service can continue processing.

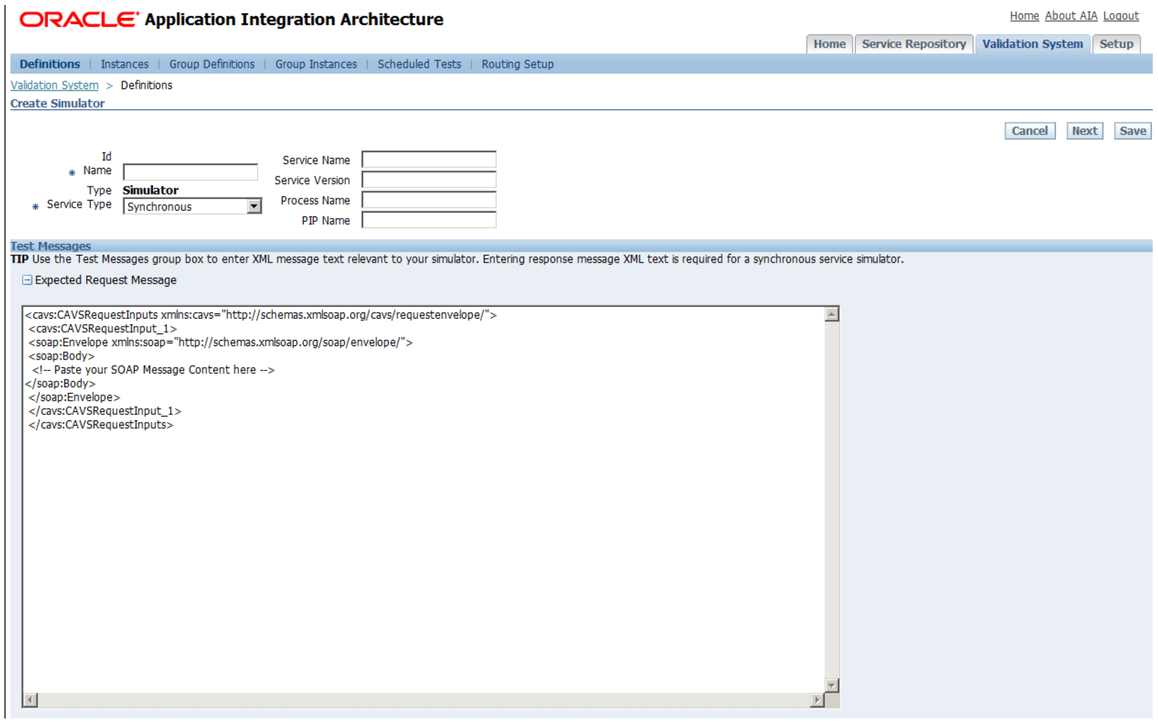
Pages Used to Create and Modify Simulator Definitions

Page Name	Navigation	Usage
Create Simulator	Access the Oracle AIA Console. Select the Validation System tab. Click the Definitions link. Click **Create Simulator**.	Create simulator definitions.
Modify Simulator Definition	• Click a Definition ID link for a simulator definition in the Search Results group box on the Definitions page. • Enter required values on the Create Simulator page and click **Next**.	Modify and manage an existing simulator definition.
View Simulator Definition	• Select ***Lock*** in the Action drop-down list box on the Modify Simulator Definition page and click **Go**. • Click the Definition Id link for a locked simulator definition on the Definitions page.	Access a read-only view of the simulator definition.

Page Name	Navigation	Usage
Search Definitions – Test	Click **Assign** in the Test Definitions grid on the Modify Simulator Definitions page.	Search for test definitions to which you want to assign a simulator definition. Search for test definitions to assign to a simulator definition. This assignment facilitates linking of simulator instances to test instances.

Creating a Simulator Definition

Access the Create Simulator page.


ORACLE Application Integration Architecture
Home About AIA Logout
Home | Service Repository | Validation System | Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup
Validation System > Definitions
Create Simulator
Cancel | Next | Save

Id		Service Name	
* Name		Service Version	
Type	Simulator	Process Name	
* Service Type	Synchronous	PIP Name	

Test Messages

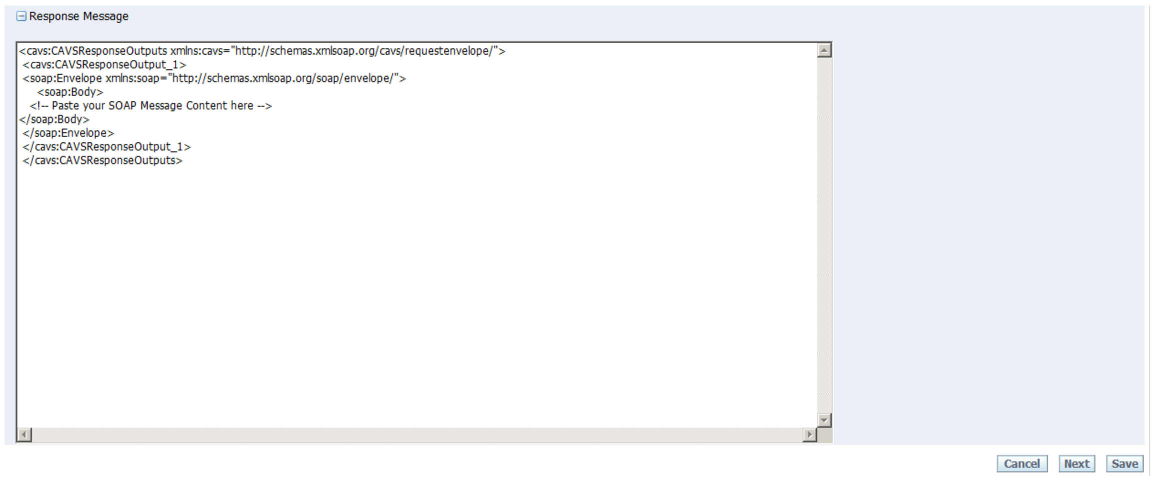
TIP Use the Test Messages group box to enter XML message text relevant to your simulator. Entering response message XML text is required for a synchronous service simulator.

Expected Request Message

```
<cavs:CAVSRequestInputs xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
 <cavs:CAVSRequestInput_1>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Body>
  <!-- Paste your SOAP Message Content here -->
</soap:Body>
 </soap:Envelope>
 </cavs:CAVSRequestInput_1>
 </cavs:CAVSRequestInputs>
```

Create Simulator page (1 of 2)

Copyright © 2009, Oracle. All rights reserved.



```
<cavs:CAVSResponseOutputs xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
 <cavs:CAVSResponseOutput_1>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
  <!-- Paste your SOAP Message Content here -->
</soap:Body>
 </soap:Envelope>
 </cavs:CAVSResponseOutput_1>
 </cavs:CAVSResponseOutputs>
```

Create Simulator page (2 of 2)

Id	Upon saving the simulator definition, a unique key identifier is assigned to the simulator definition.
Name	Enter the descriptive name you want to use for the simulator definition.
Type	Displays the type of definition you have chosen to create. On the Create Simulator page, this value will always be set to ***Simulator.***
Service Type	Select the business service pattern of the web service the simulator definition is simulating. ***Synchronous*** (request-and-reply) ***Notify*** (asynchronous request-only) ***Asynchronous two way***
Service Name	Enter the name of the web service that you want to simulate using the simulator definition.
Service Version	Enter the version of the web service you want simulate using the simulator definition.
Process Name	Enter the name of the process that includes the web service that you want to simulate using the simulator definition.
PIP Name (Process Integration Pack name)	Enter the name of the PIP that includes the web service that you want to simulate using the simulator definition.

Test Messages

Use the **Test Messages** group box to generate XPath values based on provided request XML message text. By default, SOAP envelope XML text is provided in these fields. You can paste XML text within this default SOAP envelope, or paste your own XML text already enclosed in an envelope into these fields.

For more information about how to create simulator request and response messages that hold multiple sets of test data in a single definition, see Providing Multiple Request and Response Message Sets in a Single Simulator Definition.
For more information about how to create simulator request and response messages that support chatty service conversations, see Creating a Simulator Definition that Supports Chatty Services.

Expected Request Message

Entering expected request message XML text facilitates the generation of XPath values that are used to match a received request with this simulator's expected request, as well as to validate values in this received request message. That is, the XPath values you supply provide a signature for the simulator definition that the simulator service attempts to match with arriving request actions. In addition to enabling the simulator service to match a test request with a simulator definition, the XPath criteria you provide can also serve to validate data sent in the test request.

If a simulator has been directly designated for use in the AIAConfigurationProperties.xml using the Routing Configurations page, the simulator definition will be identified directly. However, once the simulator has been identified, there may be multiple requests within it. If so, the XPath key field values provide an efficient search method for request matching.

For more information about the Routing Configurations page, see Chapter 15: Defining CAVS Routing Setup IDs.

You can enter expected request message XML text on this page and click **Generate Xpath** on the Modify Simulator Definition page to generate XPath values used to validate the actual request sent by the test definition. You may also choose to manually enter or modify the XPath values directly on the Modify Simulator Definition page. You do not need to enter request message XML if you are manually entering XPath values directly on the Modify Simulator Definition page.

You may choose to copy and paste messages from the BPEL Console, instead of manually entering them.

For more information, see Obtaining Message XML Text from a BPEL Process.

Response Message

Entering response message XML text for a simulator definition is required when the **Service Type** field value is set to ***Synchronous*** or ***Asynchronous two way.*** Enter the XML text of the response message that you want to use for the simulator definition. This response message should mimic the actual response message that would be sent by the service that the simulator definition is simulating.

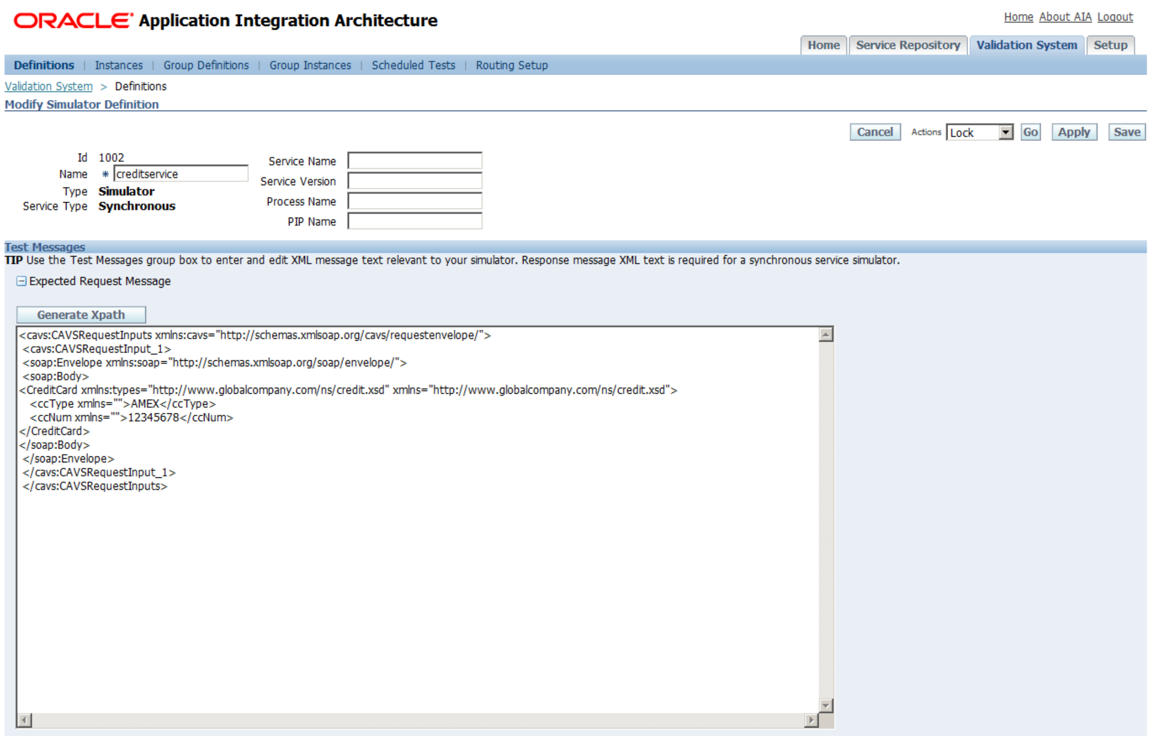
This text box is hidden when the **Service Type field** value is

Copyright © 2009, Oracle. All rights reserved.

	set to ***Notify.*** In this case, a response message is not a simulator requirement. You may choose to copy and paste messages from the BPEL Console, instead of manually entering them. **For more information**, see Obtaining Message XML Text from a BPEL Process. You may choose to generate response message XML text for simulators with the **Service Type** value set to ***Asynchronous two way*** by clicking the Get Response Message button on the Modify Simulator Definition page. **For more information**, see Modifying a Simulator Definition.
Cancel	Click to discard any updates you have made and return to the Definitions page.
Next	Click to save entries on the Create Simulator page and go to the Modify Simulator Definition page, where you can generate XPaths and further edit and manage the simulator definition.
Save	Click to save entries on the Create Simulator page and return to the Definitions page.

Modifying a Simulator Definition

Access the Modify Simulator Definition page.



ORACLE Application Integration Architecture

Home About AIA Logout

Home | Service Repository | Validation System | Setup

Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup

Validation System > Definitions

Modify Simulator Definition

Cancel Actions Lock Go Apply Save

Id	1002	Service Name	
Name	* creditservice	Service Version	
Type	**Simulator**	Process Name	
Service Type	**Synchronous**	PIP Name	

Test Messages

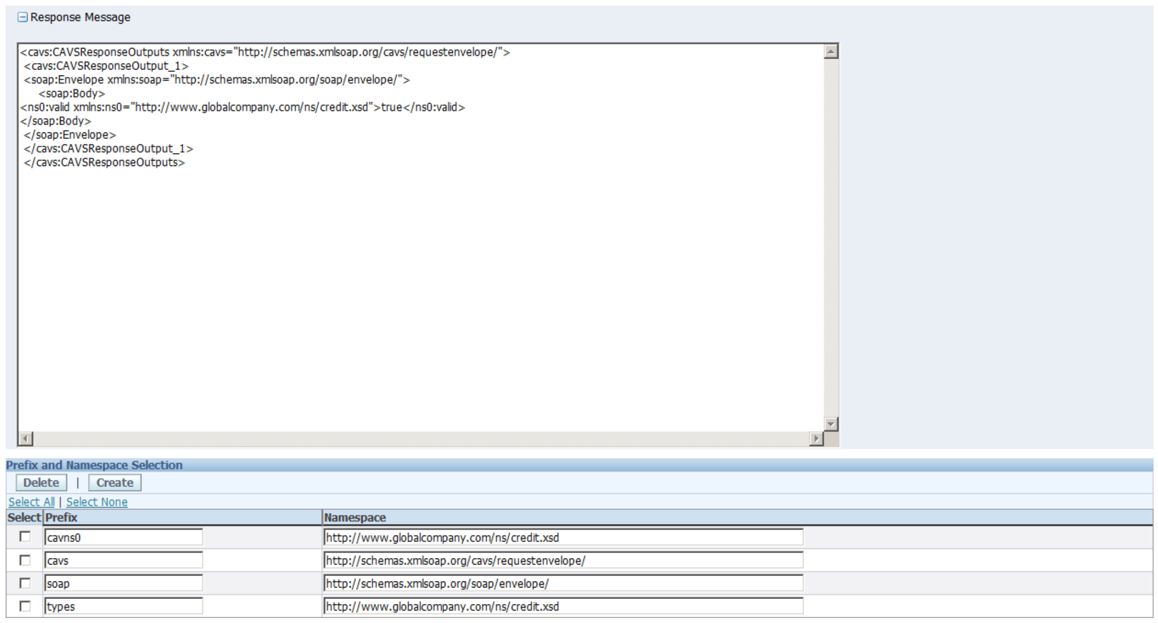
TIP Use the Test Messages group box to enter and edit XML message text relevant to your simulator. Response message XML text is required for a synchronous service simulator.

Expected Request Message

Generate Xpath

```
<cavs:CAVSRequestInputs xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
 <cavs:CAVSRequestInput_1>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Body>
<CreditCard xmlns:types="http://www.globalcompany.com/ns/credit.xsd" xmlns="http://www.globalcompany.com/ns/credit.xsd">
   <ccType xmlns="">AMEX</ccType>
   <ccNum xmlns="">12345678</ccNum>
</CreditCard>
</soap:Body>
 </soap:Envelope>
 </cavs:CAVSRequestInput_1>
 </cavs:CAVSRequestInputs>
```

Modify Simulator Definition page (1 of 3)



Response Message

```
<cavs:CAVSResponseOutputs xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
 <cavs:CAVSResponseOutput_1>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
<ns0:valid xmlns:ns0="http://www.globalcompany.com/ns/credit.xsd">true</ns0:valid>
</soap:Body>
 </soap:Envelope>
 </cavs:CAVSResponseOutput_1>
 </cavs:CAVSResponseOutputs>
```

Prefix and Namespace Selection

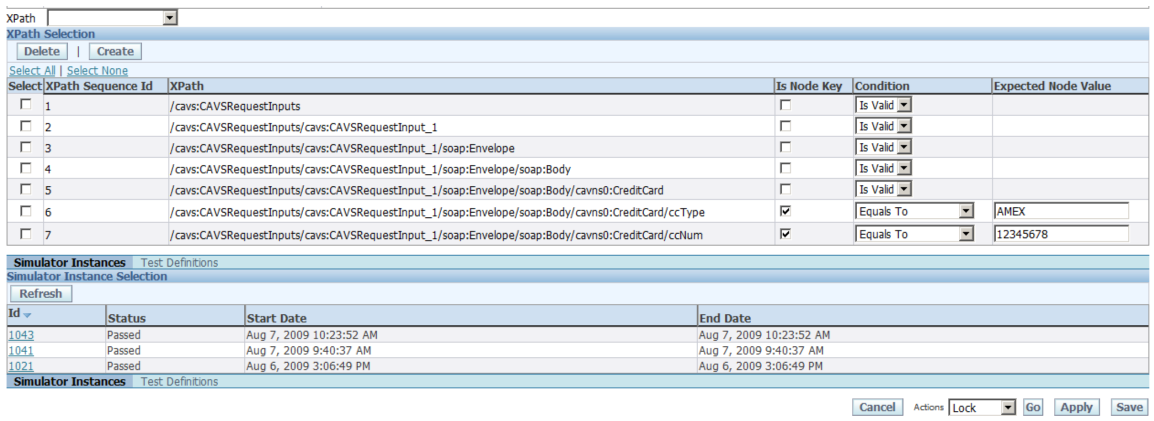
Delete | Create

Select All | Select None

Select	Prefix	Namespace
☐	cavns0	http://www.globalcompany.com/ns/credit.xsd
☐	cavs	http://schemas.xmlsoap.org/cavs/requestenvelope/
☐	soap	http://schemas.xmlsoap.org/soap/envelope/
☐	types	http://www.globalcompany.com/ns/credit.xsd

Modify Simulator Definition page (2 of 3)

Copyright © 2009, Oracle. All rights reserved.



XPath

XPath Selection

Select	XPath Sequence Id	XPath	Is Node Key	Condition	Expected Node Value
☐	1	/cavs:CAVSRequestInputs	☐	Is Valid	
☐	2	/cavs:CAVSRequestInputs/cavs:CAVSRequestInput_1	☐	Is Valid	
☐	3	/cavs:CAVSRequestInputs/cavs:CAVSRequestInput_1/soap:Envelope	☐	Is Valid	
☐	4	/cavs:CAVSRequestInputs/cavs:CAVSRequestInput_1/soap:Envelope/soap:Body	☐	Is Valid	
☐	5	/cavs:CAVSRequestInputs/cavs:CAVSRequestInput_1/soap:Envelope/soap:Body/cavns0:CreditCard	☐	Is Valid	
☐	6	/cavs:CAVSRequestInputs/cavs:CAVSRequestInput_1/soap:Envelope/soap:Body/cavns0:CreditCard/ccType	☑	Equals To	AMEX
☐	7	/cavs:CAVSRequestInputs/cavs:CAVSRequestInput_1/soap:Envelope/soap:Body/cavns0:CreditCard/ccNum	☑	Equals To	12345678

Simulator Instance Selection

Id	Status	Start Date	End Date
1043	Passed	Aug 7, 2009 10:23:52 AM	Aug 7, 2009 10:23:52 AM
1041	Passed	Aug 7, 2009 9:40:37 AM	Aug 7, 2009 9:40:37 AM
1021	Passed	Aug 6, 2009 3:06:49 PM	Aug 6, 2009 3:06:49 PM

Actions: Lock

Modify Simulator Definition page (3 of 3)

The page displays values you defined for the simulator definition. You can modify the values in editable fields.

Callback URL	If you are creating a simulator with a **Service Type** of ***Asynchronous two way,*** enter the URL of the web service that should be called back by the simulator.
Get Operations	Displays for simulators with a **Service Type** value set to ***Asynchronous two way***. Click to display the list of operations supported by the WSDL associated with the **Call Back URL** value you provided. Supported operations display in the Select WSDL Operations window. Select the operation that you want to simulate using the simulator definition. The selected operation displays in the **SOAP Action** field.
SOAP Action	If you are creating a simulator with a **Service Type** of ***Asynchronous two way,*** enter the operation of the callback URL.
Get Response Message	Click to generate a response message for the operation specified in the **SOAP Action** field.
Delay (msec)	If you are creating a simulator with a **Service Type** of ***Asynchronous two way,*** enter the number of milliseconds that you want the simulator definition to wait before issuing the call back service invocation. **Note**: If you are using this simulator along with an asynchronous two-way test definition, ensure that the **Delay (msec)** value you provide is less than the Timeout (msec) value defined for any test definition **For more information** about the Timeout (msec) field, see Modifying a Test Definition.
Cancel	Click to discard any updates you have made and return to the

	Definitions page.
Actions	Select the action you want to take with the simulator definition. ***Lock:*** Select and click **Go** to lock the simulator definition and view the simulator definition on the View Simulator Definition page. A locked definition cannot be edited. ***Duplicate:*** Select and click **Go** to duplicate the simulator definition. The duplicate definition is created using the exact values of the original, with the exception of being given a unique Id value.
Go	Make a selection in the **Action** drop-down list box and click **Go** to execute the action.
Apply	Click to apply and save any changes you have made to values on the page.
Save	Click to save entries on the page and go to the Definitions page.

For more information, see Searching for and Working with Test and Simulator Definitions.

Test Messages

The **Test Message** group box displays values defined on the Create Simulator page.

Note. If you click the **Get Response Message** button that displays for asynchronous two-way simulators on this page, any message text you have already entered in the **Response Message** text box will be overwritten.

For more information about the elements in the **Test Messages** group box, see Creating a Simulator Definition.

Prefix and Namespace Selection

Use the **Prefix and Namespace Selection** grid to define namespace data that will be used in the XPath values defined in the XPath Selection grid.

Delete	Select the Select check box for one or more namespace rows and click **Delete** to execute the deletion. This button only appears when namespace rows are present.
Create	Click to manually add and populate a namespace row.
Prefix	Prefix that should be used for the namespace.
Namespace	Namespace to be used in the XPath data for the simulator definition.

Copyright © 2009, Oracle. All rights reserved.

XPath Selection

Use the **XPath Selection** grid to work with XPath values that are used to match the simulator definition with arriving requests. XPath values can also be used to validate data send in the test request. The values in this grid use the namespace values set in the **Prefix and Namespace Selection** grid.

Note: If you are entering XPath values manually, it is important to maintain correlations with the values entered in the **Prefix and Namespace Selection** grid. Each XPath node must have a prefix that has been defined in the **Prefix and Namespace Selection** grid, unless it is an XPath expression.

Delete	Select the **Select** check box for one or more XPath rows and click **Delete** to execute the deletion. This button only appears when XPath rows are present.
Create	Click to add and manually populate an XPath row.
Xpath Sequence Id	Indicates the sequence of the XPath expressions. This value is required. This value is read-only when it has been generated using **Generate Xpath**.
Xpath	XPath value used to help match the simulator definition with arriving requests. These values can include XPath nodes and expressions. This value is read-only when it has been generated using the **Generate Xpath** button.
Is Node Key	Select if the XPath node is a key value to be used in matching arriving test requests with the simulator.
Condition	***Is Valid:*** The value provided in the XPath field is valid and no **Expected Node Value** is supplied. ***Equals To:*** The value provided in the XPath field is valid and an **Expected Node Value** is supplied. ***Not Equal To*** ***Less Than*** ***Greater Than*** ***Less Than Equal*** ***Greater Than Equal*** ***Not Null***
Expected Node Value	The value that the simulator expects to receive from the service that invokes it. When the simulator is actually executed, this value is compared with the actual value based on the validation condition selected in the **Condition** field When you use the **Generate Xpath** button to generate XPath data, this value may be populated, but can be modified as necessary. The **Condition** field value is used to qualify this value.

Simulator Instance Selection

Click the **Simulator Instances** link to display the **Simulator Instance Selection** grid, which displays information about simulator instances generated using the simulator definition.

Refresh	Click to refresh the Modify Simulator Definition page.
Id	Click to display the selected instance ID on the Simulator Instances Detail page. **For more information** about the Simulator Instances Detail page, see Viewing Simulator Instance Details.
Status	Displays the status of the simulator instance generated by the simulator definition. ***Initiated:*** The simulator instance has been initiated. ***Ended:*** This status is only applicable to simulator instances that do not involve validations. Indicates that the instance has ended. ***Faulted:*** The simulator instance could not execute properly due to exceptions or faults. ***Failed:*** The simulator instance did not pass validation. ***Passed:*** The simulator instance passed validation.
Start Date	Displays the date and time at which the simulator instance was initiated.
End Date	Displays the date and time at which the simulator instance ended.

Test Definition Selection

Click the **Test Definitions** link to display the **Test Definition Selection** grid, which displays information about test definitions associated with the simulator definition.

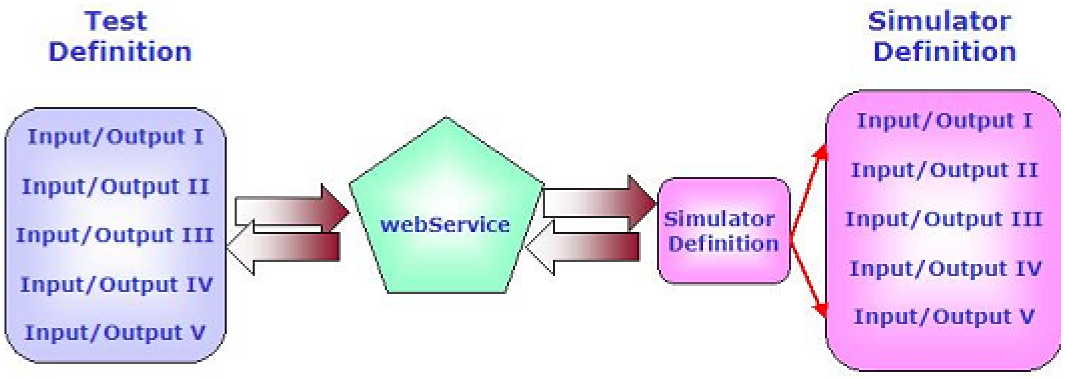
Delete	Select the **Select** check box for one or more test definition rows that you want to delete and click **Delete** to execute the deletion.
Assign	Click to access the Search Definitions - Test page, where you can search for a test definition to which you want to assign the simulator definition.
Refresh	Click to refresh the Modify Simulator Definition page.

Providing Multiple Request and Response Message Sets in a Single Simulator Definition

You can create a simulator definition that contains multiple pairs of request and response message data. This means that simulator definitions only need to be created per usage requirements, not per test data requirements.

Copyright © 2009, Oracle. All rights reserved.

For example, if you want to simulate a service against five sets of test data, you can create a single simulator definition to simulate the service and include in it all five sets of test data with which you can the service to operate. This is as opposed to creating five separate simulator definitions, one per combination of service and set of test data.



Providing multiple request and response message sets in a single simulator definition

When a simulator definition that includes multiple test data sets is invoked, the appropriate data set is matched for use based on key attributes identified in the request. At this point, the request validation and response provision can occur. Since we would typically use such definitions to handle several sets of data, it is recommended that you choose the same key values for every set of data.

Request Message Format

Use the following format to include multiple sets of request data in the simulator definition.

The `CAVSRequestInputs` and `CAVSRequestInput_1` envelope are autogenerated upon the input of the endpoint URL value on the test definition. Use copy and paste commands to create more sets; `CAVSRequestInput_2` and `CAVSRequestInput_3`, for example.

```
<cavs:CAVSRequestInputs
xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
<cavs:CAVSRequestInput_1>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body xmlns:ns1="http://xmlns.oracle.com/SimpleProcess">
   <ns1:SimpleProcessProcessRequest>
…
   </ns1:SimpleProcessProcessRequest>
   </soap:Body>
</soap:Envelope>
```

```
</cavs:CAVSRequestInput_1>

<cavs:CAVSRequestInput_2>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body xmlns:ns1="http://xmlns.oracle.com/SimpleProcess">
   <ns1:SimpleProcessProcessRequest>
   …
   </ns1:SimpleProcessProcessRequest>
   </soap:Body>
</soap:Envelope>
</cavs:CAVSRequestInput_2>

</cavs:CAVSRequestInputs>
```

Response Message Format

Use the following format to include multiple sets of response data in the simulator definition.

```
<cavs:CAVSResponseOutput_1>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body xmlns:ns1="http://xmlns.oracle.com/SimpleProcess">
   <ns1:SimpleProcessProcessResponse>
       …
   </ns1:SimpleProcessProcessResponse>
   </soap:Body>
</soap:Envelope>
</cavs:CAVSResponseOutput_1>

<cavs:CAVSResponseOutput_2>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body xmlns:ns1="http://xmlns.oracle.com/SimpleProcess">
   <ns1:SimpleProcessProcessResponse>
       …
   </ns1:SimpleProcessProcessResponse>
   </soap:Body>
</soap:Envelope>
</cavs:CAVSResponseOutput_2>
</cavs:CAVSResponseOutputs>
```

Envelope text is prepopulated. Enter actual message content within appropriate tags provided within the envelopes.

After entering request and response data sets and clicking the Generate Xpath button on the Modify Simulator Definition page, the XPath Selection grid provides access to available XPath values and enables you to select the XPaths that must be treated as key nodes.

For more information about the Modify Simulator Definition page, see Modifying a Simulator Definition.

If your testing scenario includes test definitions, you can likewise create test definitions that contain multiple request and response message sets that work with the sets defined in your simulator definition.

Copyright © 2009, Oracle. All rights reserved.

For more information, see Providing Multiple Request and Response Message Sets in a Single Test Definition.

Creating a Simulator Definition that Supports Chatty Services

You can create a simulator definition that can simulate multiple services, each with a different schema.

In general, we recommend that you create simulators that simulate a single specific service. However, in the case of chatty conversations, for the ease of maintenance, you may choose to simulate all callouts of an application business connector (ABC) service using a single simulator definition.

Using this method, you have the advantage of using one simulator for a particular ABC service, regardless of the number of callouts that need to be made. This method also provides ease of maintenance because linked callouts can all be viewed and modified in one place.

For example, in some integration scenarios, participating applications do not provide services at the same level of granularity as operations in enterprise business services (EBSs). In these scenarios, a requester ABC service may need to adopt patterns such as message enrichment, splitting, and aggregation and disaggregation as required by an EBS. Likewise, a provider ABC service may need to adopt patterns as required by participating application services.

These ABC services, which are typically implemented using BPEL process, call out to several services. To test this “chatty” ABC service using CAVS, there will likely be a need to replace the services that the ABC service calls out to with several simulators. It will also be required that these multiple request/response simulators be correlated, so that they accurately emulate the transaction of the same entity.

When this type of simulator is called, CAVS initiates the following general flow:

1. Selects simulator definition.
2. Validates the first request message based on the selected simulator.
3. Returns the appropriate response message, if the selected simulator is a two-way simulator.
4. Repeats steps 2 and 3 until the chatty service conversation is complete.

Request Message Format

Use the following format to create a simulator definition that supports chatty service conversations. This format provides the ability to specify a set of request and response messages, along with success criteria for each of them. This format is the same as that used for multiple requests and responses in a simulator definition. However, in this case, the schemas for each set will be different.

```
<cavs:CAVSRequestInputs
xmlns:cavs="http://schemas.xmlsoap.org/cavs/requestenvelope/">
<cavs:CAVSRequestInput_1>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body xmlns:ns1="http://xmlns.oracle.com/Service1">
   <ns1:Service1Request>
…
```

```
      </ns1: Service1Request>
      </soap:Body>
   </soap:Envelope>
   </cavs:CAVSRequestInput_1>

   <cavs:CAVSRequestInput_2>
   <soap:Envelope
   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
       <soap:Body xmlns:ns2="http://xmlns.oracle.com/ Service2">
      <ns2: Service2Request>
   …
      </ns2: Service2Request>
      </soap:Body>
   </soap:Envelope>
   </cavs:CAVSRequestInput_2>
```

Once you have provided request and response messages, click Generate Xpath on the Modify Simulator Definition page to generate XPath values. Modify the generated XPath values, if necessary.

For more information about the Modify Simulator Definition page, see Modifying a Simulator Definition.

When this type of simulator is called, separate simulator instances are created for each request and response pair. The evaluation of actual response versus expected response is handled per instance created for the same simulator definition.

Sending Dynamic Responses in a Simulator Response

CAVS simulator definitions are actually predefined request and response message sets. In some cases, you may not know the values for all the fields in the request message. Additionally, you may want to send these unknown dynamic values in a response to the service that called the simulator.

For example, consider the enterprise business message (EBM) ID. This value is normally generated on the fly by Oracle Application Integration Architecture (AIA) services. If you create a simulator that talks to this Oracle AIA service, you do not have a way to validate the value in the EBM ID field of the request message because the value is dynamically generated.

You may to choose to avoid validations of this value by setting the CAVS XPath validation for the EBM ID field to ***isValid.*** However, you may have a requirement in which you need to send this dynamic value back in a particular field of the simulator response. To meet this requirement, you can let the simulator pick the particular field (such as EBM ID) in the request and send it back as a field in the response.

To send a dynamic response in a simulator response:

1. Map a field from the request message and add it to the response message.

   These are two valid formats you can use:

Copyright © 2009, Oracle. All rights reserved.

```
#@#XPATH.{copy the XPath from request msg
Ex./soap:Envelop/soap:Body..}#@#

#@#SYSTEM.{SYSDATE}#@#
```

2. Before sending the response, the simulator will pick up this ID from the generated XPath, substitute the actual value, and send it in the response.

   The strings referenced here form a part of the response message. To know what the request message XPath values are, use the output that was generated by clicking the Generate Xpath button.

   For example, let's say that the request SOAP message has the following nodes:

```
<corecom:PersonName>
  <corecom:FirstName>CAVS</corecom:FirstName>
  <corecom:MiddleName>FP</corecom:MiddleName>
  <corecom:FamilyName>AIA</corecom:FamilyName>
  <corecom:CreationDateTime></corecom:CreationDateTime>
</corecom:PersonName>
```

   You would define your response SOAP message as follows:

```
<corecom:PersonName>
<corecom:FirstName>#@#XPATH.{/soap:Envelope/soap:Body/corecom:CreateC
ustomerPa
rtyListEBM/ebo:DataArea/ebo:CreateCustomerPartyList/corecomx:Contact/
corecomx:
PersonName/corecomx:FamilyName}#@#2dot1</corecom:FirstName>
<corecom:MiddleName>#@#XPATH.{/soap:Envelope/soap:Body/corecom:Create
CustomerP
artyListEBM/ebo:DataArea/ebo:CreateCustomerPartyList/corecomx:Contact
/corecomx
:PersonName/corecomx:MiddleName}#@#</corecom:MiddleName>
<corecom:FamilyName>#@#XPATH.{/soap:Envelope/soap:Body/corecom:Create
CustomerP
artyListEBM/ebo:DataArea/ebo:CreateCustomerPartyList/corecomx:Contact
/corecomx
:PersonName/corecomx:FirstName}#@#</corecom:FamilyName>
<corecom:CreationDateTime>#@#SYSTEM.{SYSDATE}#@#</corecom:CreationDat
eTime>
</corecom:PersonName>
```

   In this case, the response would be modified by the CAVS engine by copying values from the request as follows:

```
<corecom:PersonName>
  <corecom:FirstName>AIA2dot1</corecom:FirstName>
  <corecom:MiddleName>FP</corecom:MiddleName>
  <corecom:FamilyName>CAVS</corecom:FamilyName>
<corecom:CreationDateTime>2008-05-
12T15:26:43+05:30</corecom:CreationDateTime>
</corecom:PersonName>
```

   Note that `2dot1` is a static string that is always appended to the FamilyName value.

Chapter 13: Searching for Test and Simulator Definitions

This chapter discusses how to search for and work with definitions in the Composite Application Validation System (CAVS) user interface (UI).

Searching for and Working with Test and Simulator Definitions

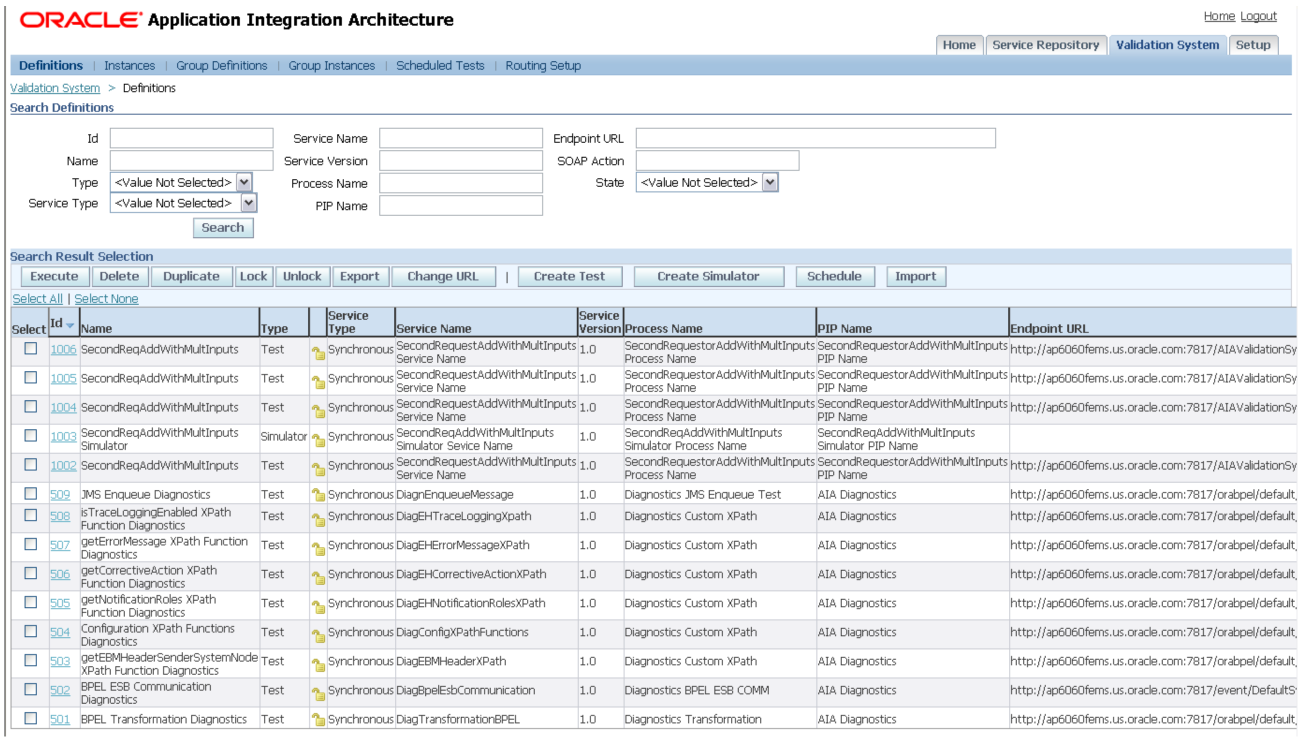
This section discusses how to search for and work with test and simulator definitions.

Page Used to Search for and Work with Definitions

Page Name	Navigation	Usage
Definitions	Access the Oracle AIA Console. Select the Validation System tab. Click the Definitions link.	Search for, execute, migrate, and manage existing test and simulator definitions. Access pages you can use to create and modify test and simulator definitions.

Searching for and Working with Test and Simulator Definitions

Access the Definitions page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup
Validation System > Definitions
Search Definitions
Search
Search Result Selection
Execute | Delete | Duplicate | Lock | Unlock | Export | Change URL | Create Test | Create Simulator | Schedule | Import
Select All | Select None

Id	
Name	
Type	<Value Not Selected>
Service Type	<Value Not Selected>
Service Name	
Service Version	
Process Name	
PIP Name	
Endpoint URL	
SOAP Action	
State	<Value Not Selected>

Select	Id	Name	Type		Service Type	Service Name	Service Version	Process Name	PIP Name	Endpoint URL
☐	1006	SecondReqAddWithMultInputs	Test	🔓	Synchronous	SecondRequestAddWithMultInputs Service Name	1.0	SecondRequestorAddWithMultInputs Process Name	SecondRequestorAddWithMultInputs PIP Name	http://ap6060fems.us.oracle.com:7817/AIAValidationSy
☐	1005	SecondReqAddWithMultInputs	Test	🔓	Synchronous	SecondRequestAddWithMultInputs Service Name	1.0	SecondRequestorAddWithMultInputs Process Name	SecondRequestorAddWithMultInputs PIP Name	http://ap6060fems.us.oracle.com:7817/AIAValidationSy
☐	1004	SecondReqAddWithMultInputs	Test	🔓	Synchronous	SecondRequestAddWithMultInputs Service Name	1.0	SecondRequestorAddWithMultInputs Process Name	SecondRequestorAddWithMultInputs PIP Name	http://ap6060fems.us.oracle.com:7817/AIAValidationSy
☐	1003	SecondReqAddWithMultInputs Simulator	Simulator	🔓	Synchronous	SecondReqAddWithMultInputs Simulator Sevice Name	1.0	SecondReqAddWithMultInputs Simulator Process Name	SecondReqAddWithMultInputs Simulator PIP Name	
☐	1002	SecondReqAddWithMultInputs	Test	🔓	Synchronous	SecondRequestAddWithMultInputs Service Name	1.0	SecondRequestorAddWithMultInputs Process Name	SecondRequestorAddWithMultInputs PIP Name	http://ap6060fems.us.oracle.com:7817/AIAValidationSy
☐	509	JMS Enqueue Diagnostics	Test	🔓	Synchronous	DiagnEnqueueMessage	1.0	Diagnostics JMS Enqueue Test	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default,
☐	508	isTraceLoggingEnabled XPath Function Diagnostics	Test	🔓	Synchronous	DiagEHTraceLoggingXpath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default,
☐	507	getErrorMessage XPath Function Diagnostics	Test	🔓	Synchronous	DiagEHErrorMessageXPath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default,
☐	506	getCorrectiveAction XPath Function Diagnostics	Test	🔓	Synchronous	DiagEHCorrectiveActionXPath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default,
☐	505	getNotificationRoles XPath Function Diagnostics	Test	🔓	Synchronous	DiagEHNotificationRolesXPath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default,
☐	504	Configuration XPath Functions Diagnostics	Test	🔓	Synchronous	DiagConfigXPathFunctions	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default,
☐	503	getEBMHeaderSenderSystemNode XPath Function Diagnostics	Test	🔓	Synchronous	DiagEBMHeaderXPath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default,
☐	502	BPEL ESB Communication Diagnostics	Test	🔓	Synchronous	DiagBpelEsbCommunication	1.0	Diagnostics BPEL ESB COMM	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/event/DefaultS
☐	501	BPEL Transformation Diagnostics	Test	🔓	Synchronous	DiagTransformationBPEL	1.0	Diagnostics Transformation	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default,

Definitions page

Search Definitions

Use the **Search Definitions** group box to enter search criteria to find the test or simulator definition you are searching for.

Id	Enter the unique key identifier assigned to the test or simulator definition.
Name	Enter the descriptive name assigned to the test or simulator definition.
Type	Select the type of definition. ***<Value Not Selected>:*** Select to display all definition types. ***Test*** ***Simulator***
Service Type	Select the business service pattern of the web service for which the definition was created. ***<Value Not Selected>:*** Select to display definitions for all service types. ***Synchronous*** ***Notify*** ***Asynchronous two way***
Service Name	Enter the name of the web service for which the definition was created.

Copyright © 2009, Oracle. All rights reserved.

Service Version	Enter the version of the service for which the definition was created. This is the web service whose URL is provided in the **Endpoint URL** field.
Process Name	Enter the name of the process that includes the web service for which the definition was created.
PIP Name (Process Integration Pack name)	Enter the name of the Process Integration Pack that includes the web service for which the definition was created.
Endpoint URL	Enter the URL of the web service for which the definition was created.
SOAP Action	Enter the operation called by the web service for which the definition was created.
State	Select the state of the definition. ***<Value Not Selected>:*** Select to display definitions in all states. ***Locked*** ***Unlocked***
Search	Click to execute a search for definitions using the search criteria entered in the **Search Definitions** group box.

Search Result Selection

Use the **Search Result Selection** grid to work with definitions returned in your search results. Upon accessing this page, the grid displays all definitions.

Execute	Select the **Select** check box for one or more test definitions that you want to run and click **Execute** to execute the test definition. When a test definition has successfully executed, you can view details of the test instance generated by the test execution on the Test Instance Details page. **For more information** about the Test Instance Details page, see Viewing Test Instance Details. Simulator definitions cannot be executed.
Delete	Select the **Select** check box for one or more definitions that you want to delete and click **Delete** to execute the deletion.
Duplicate	Select the **Select** check box for one or more definitions that you want to duplicate and click **Duplicate** to execute the duplication. The duplicate definition is created using the exact values of the original, with the exception of being assigned a unique ID value.
Lock	Select the **Select** check box for one or more definitions that you want to lock and click **Lock** to lock the definitions. A definition with its **State** value set to ***Locked*** cannot be edited.

Unlock	Select the **Select** check box for one or more definitions that you want to unlock and click **Unlock** to unlock the definitions. An unlocked definition can be edited. A definition with its **State** value set to ***Unlocked*** is editable.
Export	**For more information** about exporting definitions and instances, see Exporting and Importing Definitions and Instances.
Change URL	Select the **Select** check box for one or more test definitions for which you want to change the endpoint URL value. Click **Change URL** to launch a pop-up window in which you can enter the new endpoint URL value that you want to use for the selected test definitions.
Create Test	Click to access the Create Test page, where you can create a test definition. **For more information** about the Create Test page, see Creating a Test Definition.
Create Simulator	Click to access the Create Simulator page, where you can create a simulator definition. **For more information** about the Create Simulator page, see Creating a Simulator Definition.
Schedule	Select the **Select** check box for one or more definitions for which you want to schedule a run time. Click **Schedule** to access the Scheduler page, where you can define the run time for selected definitions. **For more information** about scheduling tests, see Working with Scheduled Tests.
Import	**For more information** about importing test definitions, see Exporting and Importing Definitions and Instances.
Id	Click for an unlocked test definition to access the Modify Test Definition page. Click for a locked test definition to access the View Test Definition page. Click for an unlocked simulator definition to access the Modify Simulator Definition page. Click for a locked simulator definition to access the View Simulator Definition page.

Copyright © 2009, Oracle. All rights reserved.

For more information, see Creating and Modifying Test Definitions.
For more information, see Creating and Modifying Simulator Definitions.

Chapter 14: Working with Group Definitions

This chapter discusses how to create and modify group definitions.

Creating Group Definitions

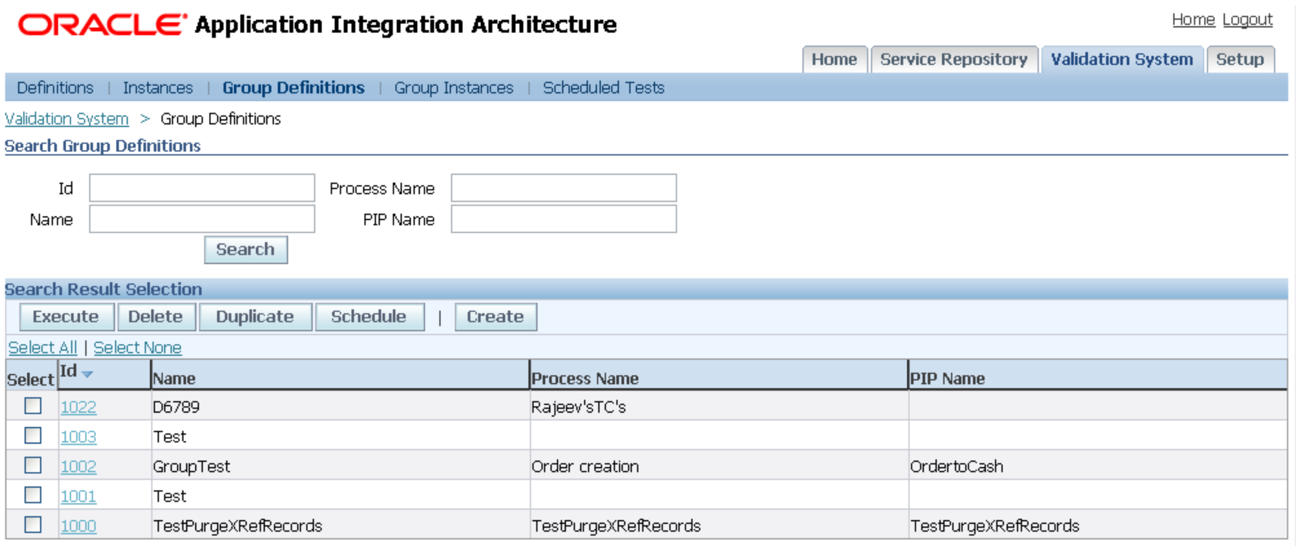
A group definition is a configuration of a single execution of one or more test definitions in a single-threaded batch.

Pages Used to Work with Group Definitions

Page Name	Navigation	Usage
Group Definitions	Access the Oracle AIA Console. Select the Validation System tab. Click the Group Definitions link.	Search for, execute, and manage existing group definitions. Access a page you can use to create and modify group definitions.
Scheduler	Access the Oracle AIA Console. Select the Validation System tab. Click the Group Definitions link. Select one or more definitions and click Scheduler.	Schedule run times for selected definitions.
Group Definition Detail	Click Create Group Definition on the Group Definitions page.	Create and modify a group definition that combines one or more tests and executes them in a single-threaded batch sequence.
Search Definitions - Test	Click the Assign button on the Group Definition Detail page.	Search for test definitions that you want to assign to a group definition.

Working with Group Definitions

Access the Group Definitions page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests
Validation System > Group Definitions
Search Group Definitions
Id
Name
Process Name
PIP Name
Search
Search Result Selection
Execute | Delete | Duplicate | Schedule | Create
Select All | Select None

Select	Id	Name	Process Name	PIP Name
☐	1022	D6789	Rajeev'sTC's	
☐	1003	Test		
☐	1002	GroupTest	Order creation	OrdertoCash
☐	1001	Test		
☐	1000	TestPurgeXRefRecords	TestPurgeXRefRecords	TestPurgeXRefRecords

Group Definitions page

Search Group Definitions

Use the **Search Group Definitions** group box to enter search criteria to find the group definition you are searching for.

Id	Enter the unique key identifier assigned to the group definition.
Name	Enter the descriptive name assigned to the group definition.
Process Name	Enter the name of the process associated with the group definition.
Search	Click to execute a search for group definitions using the search criteria entered in the Search Group Definitions group box.

Search Result Selection

Use the **Search Result Selection** grid to work with group definitions returned in your search results. Upon accessing this page, the grid is populated by all group definitions.

Execute

Select the **Select** check box for one or more group definitions that you want to run and click **Execute** to execute the group definition. When a group definition has successfully executed, you can view details of the group instance on the Group Instances Detail page.

For more information about the Group Instances Detail page, see Working with Group Instances.

Delete

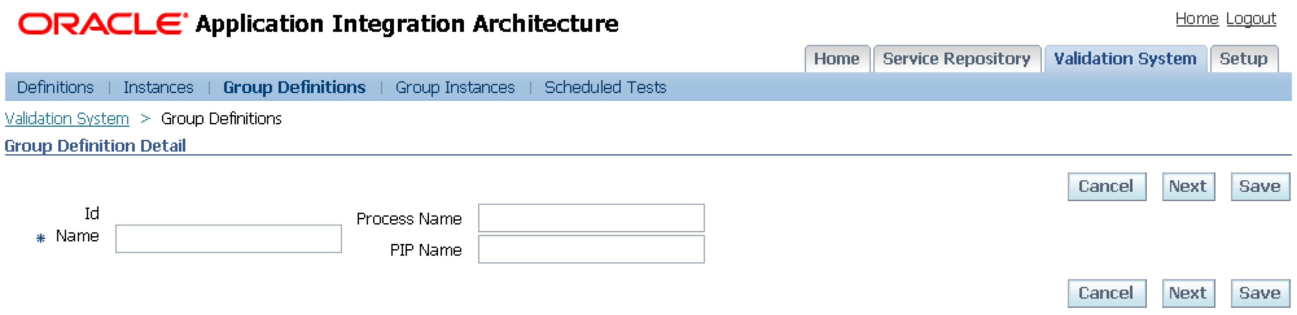
Select the **Select** check box for one or more group definitions that you want to delete and click **Delete** to execute the deletion.

Copyright © 2009, Oracle. All rights reserved.

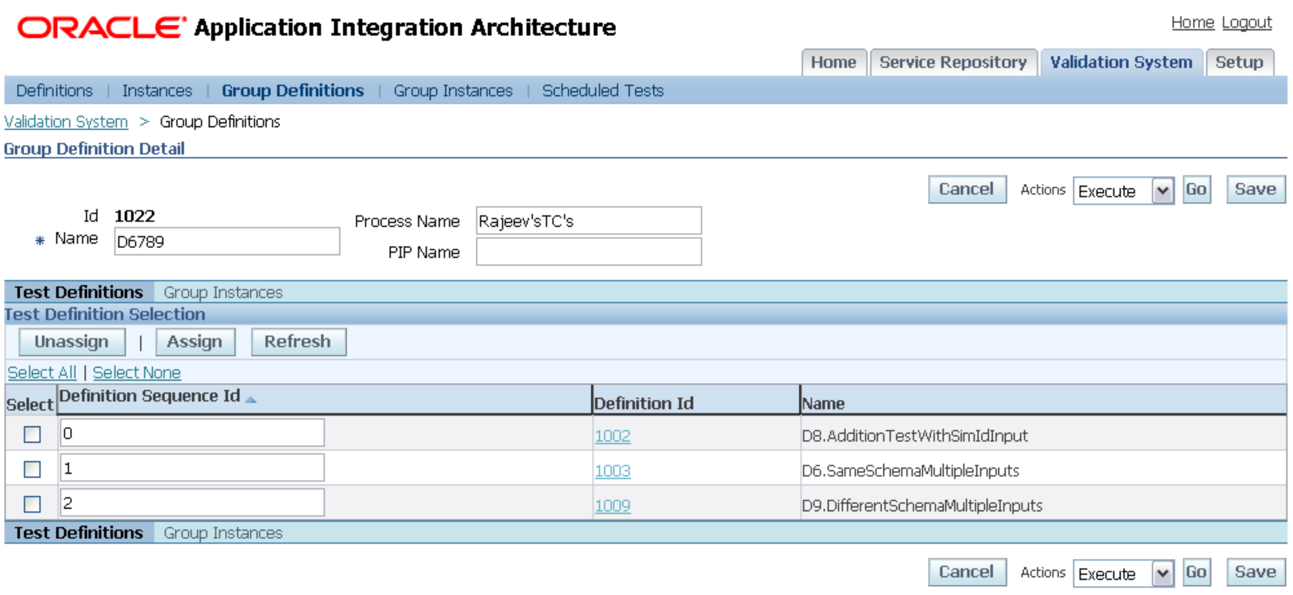
Duplicate	Select the **Select** check box for one or more group definitions that you want to duplicate and click **Duplicate** to execute the duplication. The duplicate group definition is created using the exact values of the original, with the exception of being given a unique Id value.
Schedule	Select the **Select** check box for one or more definitions for which you want to schedule a run time. Click **Schedule** to access the Scheduler page, where you can define the run time for selected definitions. **For more information** about scheduling tests, see Working with Scheduled Tests.
Create	Click to access the Group Definition Detail page, where you can create a group definition
Id	Click to access the Group Definition Detail page.

Creating and Modifying a Group Definition

Access the Group Definition Detail page.



Group Definition Detail page (new group definition)


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests
Validation System > Group Definitions
Group Definition Detail
Cancel Actions Execute Go Save
Test Definitions | Group Instances
Test Definition Selection
Unassign | Assign Refresh
Select All | Select None

Id 1022
* Name D6789
Process Name Rajeev'sTC's
PIP Name

Select	Definition Sequence Id	Definition Id	Name
☐	0	1002	D8.AdditionTestWithSimIdInput
☐	1	1003	D6.SameSchemaMultipleInputs
☐	2	1009	D9.DifferentSchemaMultipleInputs

Group Definition Detail page (existing definition)

Id	Upon saving a new group definition, a unique key identifier is assigned to the group definition. For an existing group definition, displays the unique key identifier assigned to the group definition.
Name	For a new group definition, enter a descriptive name for the group definition. For an existing group definition, displays the descriptive name assigned to the group definition.
Process Name	For a new group definition, enter the name of the process you want to associate with the group definition. For an existing group definition, displays the process name associated with the group definition. This value is editable.
PIP Name (process integration pack)	For a new group definition, enter the name of the PIP you want to associate with the group definition. For an existing group definition, displays the PIP associated with the group definition. This value is editable.
Cancel	Click to discard updates to the page and return to the Group Definitions page.
Actions	Select the action you want to take with the group definition. ***Execute:*** Select and click **Go** to execute the group definition. When a group definition has successfully executed, you can view details of the test instance on the Group Instances page. ***Duplicate:*** Select and click **Go** to duplicate the group definition. The duplicate definition is created using the exact values of the original, with the exception of being given a unique Group Definition Id value.

Copyright © 2009, Oracle. All rights reserved.

Go	Make a selection in the **Action** drop-down list box and click **Go** to execute the action.
Apply	Click to apply and save any changes you have made to values on the page.
Next	For a new group definition, click to save entries and display further group definition details on the Group Definition Detail page. This button does not appear for existing group definitions.
Save	Click to save entries on the Group Definition Detail page and return to the Group Definitions page.

Test Definition Selection

Click the **Test Definitions** link to access the **Test Definition Selection** grid, where you can associate test definitions with the group definition.

Unassign	Select the **Select** check box for one or more test definition rows that you want to disassociate from the group definition. Click **Unassign** to execute the disassociation.
Assign	Click to access the Search Definitions - Test page, where you can search for a test definition that you want to assign to the simulator definition.
Refresh	Click to refresh the Group Definition Detail page.
Definition Sequence Id	Displays the sequence in which the test definition is initiated by the group definition.
Definition Id	Click for an unlocked test definition to access the Modify Test Definition page. Click for a locked test definition to access the View Test Definition page.

For more information, see Creating and Modifying Simulator Definitions.

Group Instance Selection

Click the **Group Instances** link to display the **Group Instance Selection** grid, which displays information about group instances generated by the group definition.

Refresh	Click to refresh the Group Definition Detail page.
Id	Click to access the Group Instances Detail page.
Start Date	Displays the date and time at which the group instance was initiated.

Chapter 15: Defining CAVS Routing Setup IDs

This chapter provides an overview of Composite Application Validation System (CAVS) routing setup IDs and discusses how to define CAVS routing setup IDs.

Understanding CAVS Routing Setup IDs

CAVS routing setups are used in the following two scenarios.

- CAVS test definitions call services that in turn, call CAVS simulators.
- Actual applications and services call CAVS simulators instead of calling subsequent actual services.

CAVS routing setup IDs are used to route the service calls to the CAVS simulators. Use the pages covered in this chapter to set up CAVS routing setup IDs before executing tests. These CAVS routing setup IDs are stored as RouteToCAVS properties in the AIAConfigurationProperties.xml file in <aia.home>/config/. This file is read during runtime to determine whether routing needs to be made to a CAVS simulator or to an actual system.

For example, you could create the following three routing setup IDs for the scenarios:

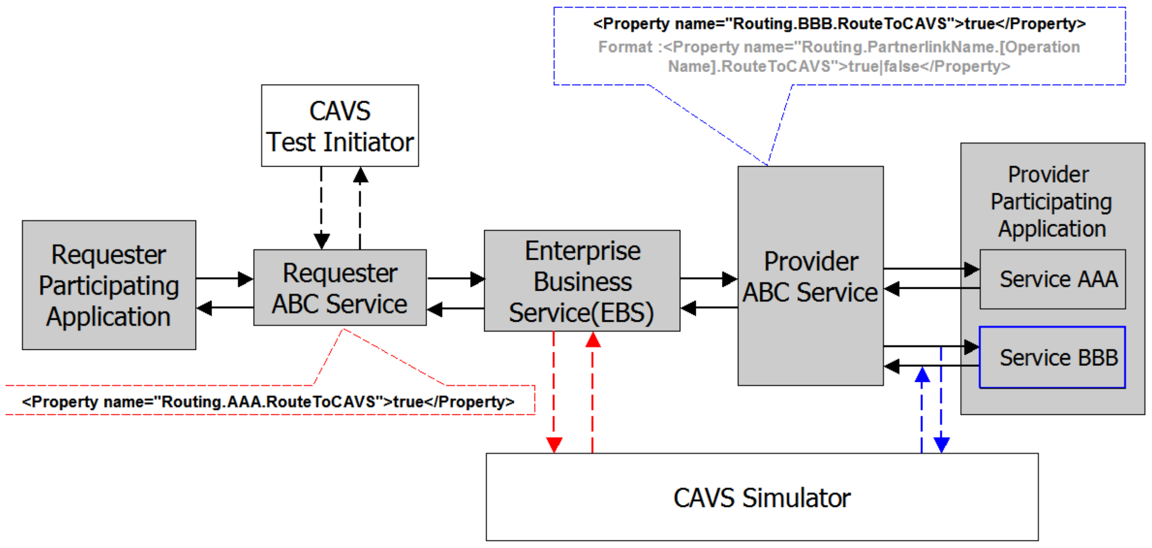
1. To test the requester application business connector (ABC) service or when the provider ABC service is not available, you would want the requester ABC service to call a simulator instead of actual Oracle AIA services. For this scenario, create a routing setup ID to set the RouteToCAVS property to TRUE on the requester ABC service. This will ensure that the message is routed to the CAVS simulator, as indicated in red.

   Note: An actual participating application or test definition can be used to invoke the requester ABC service.

2. To test the provider ABC service or when the provider application is not available, you would want the provider ABC service to call a simulator instead of an actual provider application service. For this scenario, create a routing setup ID to set the RouteToCAVS property to TRUE on the provider ABC service. This will ensure that the message is routed to the CAVS simulator, as indicated in blue.

   Note: If there is more than one callout from the provider ABC service, the CAVS user can have fine-grained control over the routing by setting the routing at the PartnerLink level (and optionally at the operation level). This is indicated in the diagram.

3. To test the requester ABC service and the provider ABC service together, you would create a routing setup ID to set the RouteToCAVS property to FALSE on the requester ABC service so that it can go on to call the provider ABC service and TRUE on the provider ABC service.



<Property name="Routing.BBB.RouteToCAVS">true</Property>
Format :<Property name="Routing.PartnerlinkName.[Operation Name].RouteToCAVS">true|false</Property>
CAVS Test Initiator
Requester Participating Application
Requester ABC Service
Enterprise Business Service(EBS)
Provider ABC Service
Provider Participating Application
Service AAA
Service BBB
<Property name="Routing.AAA.RouteToCAVS">true</Property>
CAVS Simulator

Sample scenarios for using CAVS routing setup IDs

This diagram illustrates the need for different routing setup IDs to test each of these three scenarios. When creating test definitions that will be used to initiate these test scenarios, CAVS allows you to associate the test definition with a specific routing setup ID. This routing setup ID determines the configuration that is required and automatically applies it before executing the test.

For example, if these three test scenarios are grouped into a single test group for execution, each test requires a different routing setup. In this case, you would create three routing setup Ids, 1001, 1002, and 1003, for example. Each routing setup ID is required by one of the scenarios. You assign routing setup ID 1001 to the test definition for scenario 1, 1002 to the test definition for scenario 2, and so forth. When these three test definitions are executed as a part of the test group, the CAVS system automatically applies routing setup IDs 1001, 1002, and 1003 when executing the appropriate test definition. This eliminates the need to manually modify routing configurations between test scenario executions.

If, for example, you did not associate routing setup ID 1002 with the test definition for scenario 2, the test definition for scenario 2 would use routing setup ID 1001, because it was the last applied routing setup ID.

Routing setup IDs are also useful when it comes to scheduled tests. For example, you may have tests scheduled to execute throughout the day and night. It is not practical to expect a user to be present to edit routing configurations at all times before scheduled tests execute.

For more information about assigning a routing setup ID to a test definition, see Creating a Test Definition.

Another option for applying routings is to directly modify them on the Configuration page.

For more information about the Configuration page, see Chapter 7: Loading Oracle AIA Configuration File Updates.

Copyright © 2009, Oracle. All rights reserved.

Defining CAVS Routing Setup IDs

This section discusses how to:

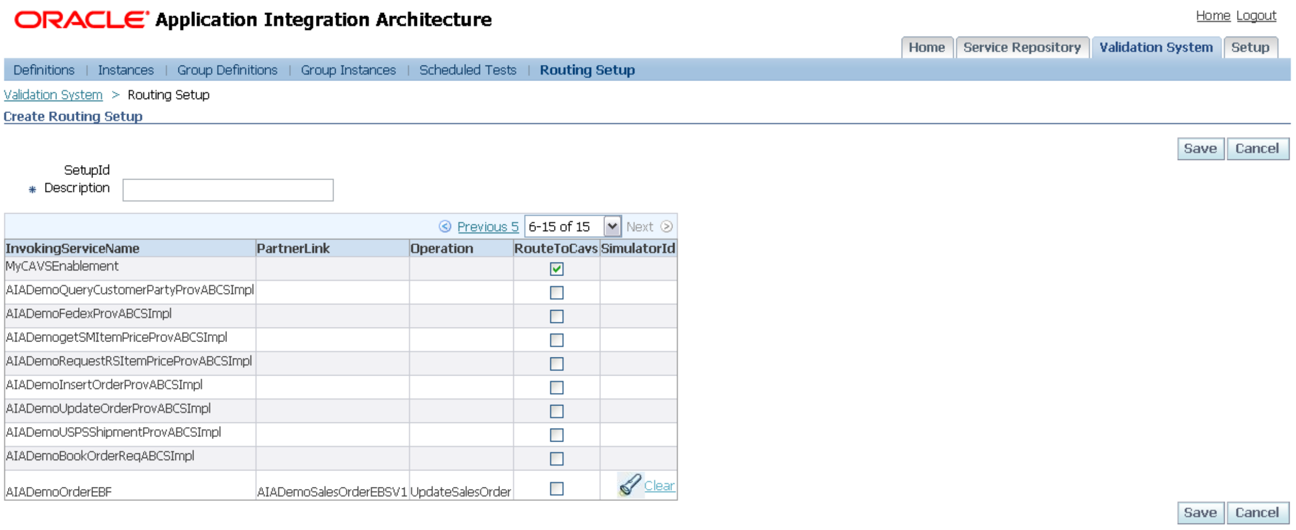
- Create CAVS routing setup IDs.
- Search for CAVS routing setup IDs.
- Modify CAVS routing setup ID.

Pages Used to Define CAVS Routing Setup IDs

Page Name	Navigation	Usage
Create Routing Setup	Access the Oracle AIA Console. Select the Validation System tab. Click the Routing Setup link. Click Create.	Create a new CAVS routing.
Simulator ID Selection	Access the Create Routing Setup page. Click the SimulatorId button.	Select the simulator definition that you want an invoking service to route to.
Routing Setup	Access the Oracle AIA Console. Select the Validation System tab. Click the Routing Setup link. Click a SetupId link.	Modify existing routing setup IDs.
Search Routing Setups	Access the Oracle AIA Console. Select the Validation System tab. Click the Routing Setup link.	Search for an existing CAVS routing setup ID, or access functionality to create and delete routings.
Configurations	Access the Search Routing Setups page. Click View Routing.	Access a read-only view of all current routing setup IDs.
Configuration	Access the Oracle AIA Console. Select the Setup tab. Click the Configuration link.	Quickly set up a routing without having to create a routing setup. This is particularly useful when you are only interested in using simulators without test definitions. **For more information**, see Chapter 7: Loading Oracle AIA Configuration File Updates.

Creating CAVS Routing Setup IDs

Access the Create Routing Setup page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup
Validation System > Routing Setup
Save Cancel
Previous 5 | 6-15 of 15 | Next

Create Routing Setup

SetupId

* Description

InvokingServiceName	PartnerLink	Operation	RouteToCavs	SimulatorId
MyCAVSEnablement			☑	
AIADemoQueryCustomerPartyProvABCSImpl			☐	
AIADemoFedexProvABCSImpl			☐	
AIADemogetSMItemPriceProvABCSImpl			☐	
AIADemoRequestRSItemPriceProvABCSImpl			☐	
AIADemoInsertOrderProvABCSImpl			☐	
AIADemoUpdateOrderProvABCSImpl			☐	
AIADemoUSPSShipmentProvABCSImpl			☐	
AIADemoBookOrderReqABCSImpl			☐	
AIADemoOrderEBF	AIADemoSalesOrderEBSV1	UpdateSalesOrder	☐	Clear

Create Routing Setup page

Upon access, this page displays routing information for all services with a RoutetoCAVS property defined in the AIAConfigurationProperties.xml file in <aia.home>/config/.

Use this page to perform a one-time setup of routing setup IDs that you can later associate with test definitions using the SetupId field on the Create Test page. By making this association, the required routing setup will be automatically applied during the execution of the test definition.

For more information about the SetupId field, see Creating a Test Definition.

Data saved on this page is stored in a CAVS table, rather than in the AIAConfigurationProperties.xml.

For more information about how to quickly define a routing configuration that is stored in AIAConfigurationProperties.xml, see Chapter 7: Loading Oracle AIA Configuration File Updates.

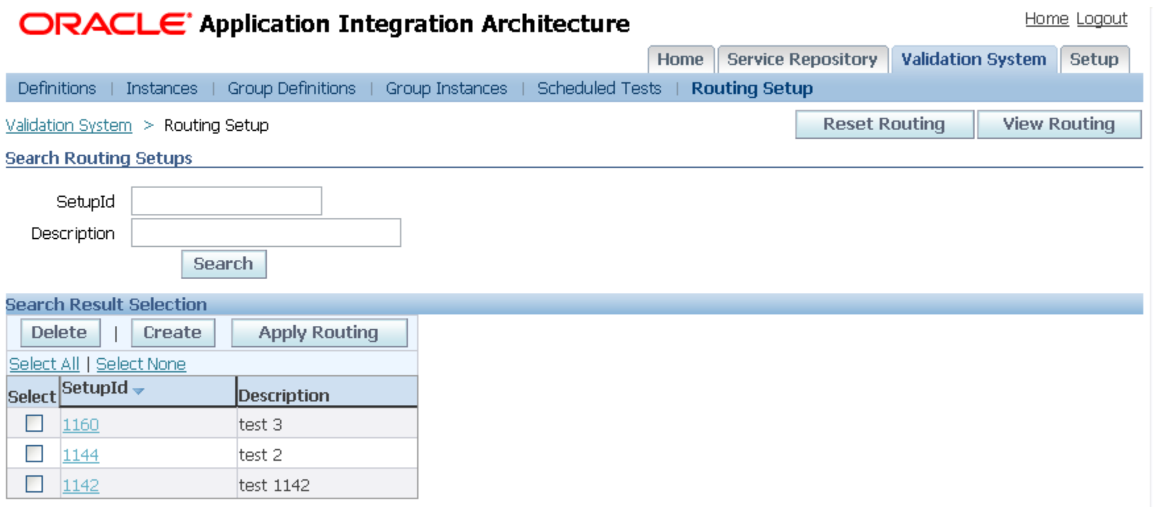
SetupId	Upon saving, a sequentially generated ID is assigned to the routing setup ID.
Description	Enter a description of the routing setup ID you are creating.
InvokingServiceName	Lists all services defined in the AIAConfigurationProperties.xml file in <aia.home>/config/.
PartnerLink	The PartnerLink that is invoked by the service that you want to route to the CAVS simulator.
Operation	The operation of the PartnerLink that you want to route to the CAVS simulator. Displays a value only when multiple operations on the service are invoked using the same PartnerLink, typically when calling an enterprise business service.

Copyright © 2009, Oracle. All rights reserved.

RouteToCavs	Select to indicate that the invoking service should route to the selected CAVS simulator.
SimulatorId	Click the button to access the Simulator ID Selection page, where you can select the simulator definition that you want an invoking service to route to.
	Upon access, the page displays all available CAVS simulator definition IDs. Select the option for the simulator definition ID to which you want to route an invoking service.

Searching for CAVS Routing Setup IDs

Access the Search Routing Setups page.



Select	SetupId	Description
☐	1160	test 3
☐	1144	test 2
☐	1142	test 1142

Search Routing Setups page

Reset Routing	Click to set all routing configurations to ***FALSE.*** This means that all routings to simulators (RoutetoCAVS property settings) in the AIAConfigurationProperties.xml file will be set to ***FALSE,*** whether you have defined them through the Routing Setup pages or directly in the file.
View Routing	Click to access the Configuration page, where you can access a read-only view of the last applied, or active, routing setup ID.

Search Routing Setups

Use the **Search Routing Setups** group box to enter search criteria to find the routing setup IDs you are searching for.

SetupId	Enter the ID assigned to the routing setup ID you are searching for.
Description	Enter description text used for the routing setup ID you are searching for.
Search	Click to execute a search for routing setup IDs using the search criteria entered in the **Search Routing Setups** group box.

Search Result Selection

Delete	Select the **Select** check box for one or more routing setup IDs that you want to delete and click **Delete** to execute the deletion.
Create	Click to access the Create Routing Setup page, where you can create a routing setup ID.

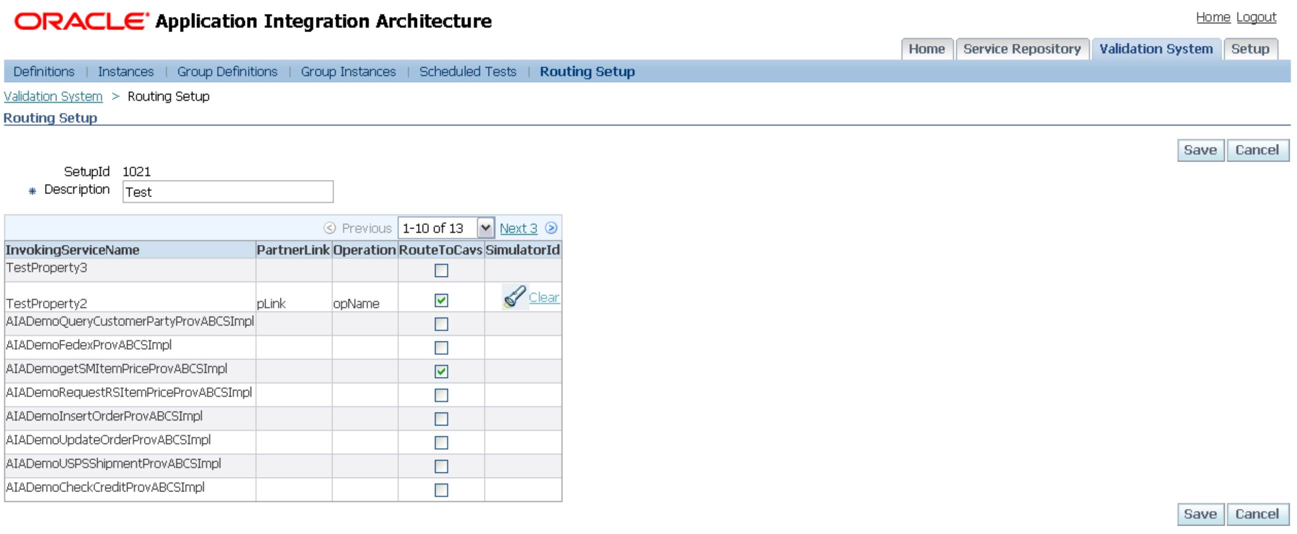
For more information, see Creating CAVS Routing Setup IDs.

Apply Routing	Once you have created a new routing setup ID, you may apply it to populate the AIAConfigurationProperties.xml file. To do this, select the **Select** check box for a single routing setup ID and click **Apply Routing**. If you apply the routing setup ID to the AIAConfigurationProperties.xml file, it becomes a routing configuration that is applied in all executions of the associated invoking service, not just when the routing setup ID is referenced on a test definition.
SetupId	Click to access the Routing Setup page, where you can modify an existing routing setup ID.

For more information the Routing Setup page, see Modifying Routing Setup IDs.

Modifying Routing Setup IDs

Access the Routing Setup page.


Application Integration Architecture
Home Logout
Home
Service Repository
Validation System
Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup
Validation System > Routing Setup
Routing Setup
Save Cancel
Previous 1-10 of 13 Next 3
Clear
Save Cancel

SetupId 1021
* Description Test

InvokingServiceName	PartnerLink	Operation	RouteToCavs	SimulatorId
TestProperty3			☐	
TestProperty2	pLink	opName	✓	Clear
AIADemoQueryCustomerPartyProvABCSImpl			☐	
AIADemoFedexProvABCSImpl			☐	
AIADemogetSMItemPriceProvABCSImpl			✓	
AIADemoRequestRSItemPriceProvABCSImpl			☐	
AIADemoInsertOrderProvABCSImpl			☐	
AIADemoUpdateOrderProvABCSImpl			☐	
AIADemoUSPSShipmentProvABCSImpl			☐	
AIADemoCheckCreditProvABCSImpl			☐	

Routing Setup page

Copyright © 2009, Oracle. All rights reserved.

Use this page modify existing routing setups. Data saved on this page is stored in a CAVS table, rather than in the AIAConfigurationProperties.xml. If you want to apply the data to the AIAConfigurationProperties.xml file, you must click Apply Routing for the routing setup ID on the Search Routing Setups page.

For more information about the Apply Routings button, see Searching for CAVS Routing Setup IDs.

SetupId	Displays the ID you assigned to routing setup ID on the Create Routing Setup page.
Description	If applicable, edit the routing setup ID description.
Invoking Service Name	This is the service after which the service routing to CAVS should happen.
PartnerLink	The PartnerLink that is invoked by the service that you want to route to the CAVS simulator.
Operation	The operation of the PartnerLink that you want to route to the CAVS simulator. Displays a value only when multiple operations on the service are invoked using the same PartnerLink, typically when calling an enterprise business service.
RouteToCavs	Select to indicate that the invoking service should route to the selected CAVS simulator.
SimulatorId	Click the button to access the Simulator ID Selection page, where you can select the simulator definition that you want an invoking service to route to. If a simulator is associated with a row, the **Clear** link displays. Click to clear the association.

Chapter 16: Scheduling Tests

This chapter discusses how to work with scheduled tests.

Working with Scheduled Tests

In this section, we discuss how to:

- Schedule tests
- Search for scheduled tests.
- View scheduled test results.

Tests can be scheduled on the Definitions page and Group Definitions page.

For more information about the Definitions page, see Searching for and Working with Test and Simulator Definitions.
For more information about the Group Definitions page, see Working with Group Definitions.

Once scheduled, Composite Application Validation System (CAVS) provides pages you can use to access further information about the scheduled tests.

Note: The Scheduled Tests feature is not functional when running Oracle Application Integration Architecture (AIA) Foundation Pack on WebLogic Server.

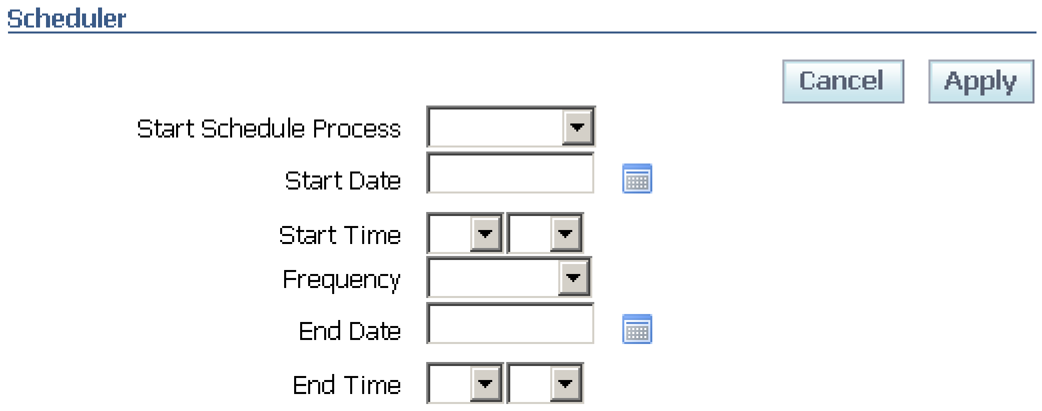
Pages Used to Work with Scheduled Tests

Page Name	Navigation	Usage
Scheduler	• Access the Oracle AIA Console. Select the Validation System tab. Click the Definitions link. Select one or more definitions and click Scheduler. • Access the Oracle AIA Console. Select the Validation System tab. Click the Group Definitions link. Select one or more definitions and click Scheduler.	Schedules run times for selected definitions.
Scheduled Tests	Access the Oracle AIA Console. Select the Validation System tab. Click	Search for scheduled tests.

Page Name	Navigation	Usage
	the Scheduled Tests link.	
Test Execution Summary	Click a View Result link on the Scheduled Tests page.	View details about the execution of a scheduled test group.

Scheduling Tests

Access the Scheduler page.



Start Schedule Process	
Start Date	
Start Time	
Frequency	
End Date	
End Time	

Scheduler page

On the Definitions page or Group Definitions page, select the **Select** check box for one or more definitions for which you want to schedule a run time. Click **Schedule** to access the Scheduler page, where you can define the run time and frequency for selected definitions.

For more information about the Definitions page, see Searching for and Working with Test and Simulator Definitions.
For more information about the Group Definitions page, see Working with Group Definitions.

This functionality enables a CAVS user to execute tests at specific times and intervals. For example, you can schedule tests to be run to coincide with new build dates.

Start schedule process	Select when you want the test(s) to commence. ***immediate:*** Select to run the test(s) immediately. Specify **Frequency** and **End Date** values. ***later:*** Select to run the test(s) at a later time. Specify **Start Date**, **Time**, **Frequency**, and **End Date** values.
Start Date	Specify the date on which you want the test(s) to commence.
Start Time	Specify the time at which you want the test(s) to commence.
Frequency	Specify how often you want the test(s) to run. ***ONLY ONCE*** ***HOURLY***

Copyright © 2009, Oracle. All rights reserved.

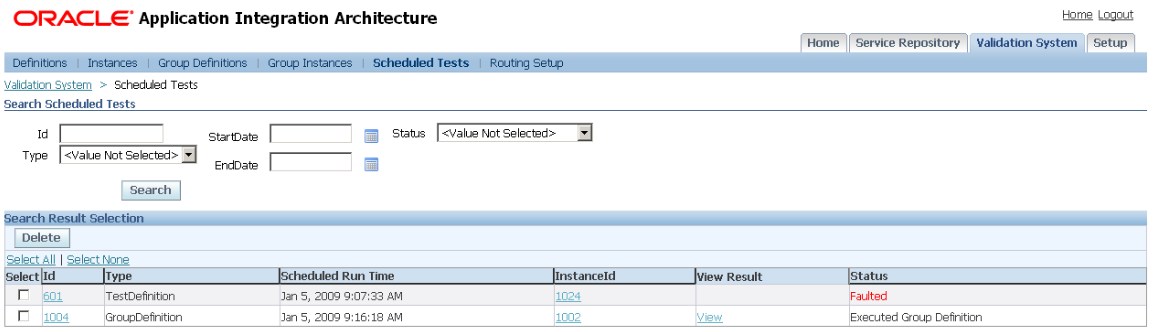
	DAILY
	WEEKLY
End Date	Select the date on which you want the test(s) to stop running.
End Time	Specify the time at which you want the test(s) to stop running.
Apply	Click to schedule the selected test(s).

Once tests have been scheduled, you can use the Scheduled Tests page to view details about scheduled test.

For more information, see Searching for Scheduled Tests.

Searching for Scheduled Tests

Access the Scheduled Tests page.


ORACLE Application Integration Architecture

Home Logout

Home | Service Repository | Validation System | Setup

Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup

Validation System > Scheduled Tests

Search Scheduled Tests

Id:

Type: <Value Not Selected>

StartDate:

EndDate:

Status: <Value Not Selected>

Search

Search Result Selection

Delete

Select All | Select None

Select	Id	Type	Scheduled Run Time	InstanceId	View Result	Status
☐	601	TestDefinition	Jan 5, 2009 9:07:33 AM	1024		Faulted
☐	1004	GroupDefinition	Jan 5, 2009 9:16:18 AM	1002	View	Executed Group Definition

Scheduled Tests page

Search Scheduled Tests

Use the **Search Scheduled Tests** group box to enter search criteria to find the scheduled definition you are searching for.

Id	Enter the unique key identifier assigned to the test or group definition.
Type	Select the type of definition.
	<Value Not Select>: Select to display all definition types.
	TestDefinition
	GroupDefinition
StartDate	Select the date on which the definition was scheduled to commence.
EndDate	Select the date on which the definition was scheduled to stop running.
Status	***<Value Not Selected>:*** Select to display all statuses.

	Passed
	Failed
	Faulted
	Delayed: Set for asynchronous two-way test instances. Indicates that the test has been executed, but is awaiting a response.
	Ended: Set for asynchronous (notify) test instances for which there is never success criteria. Indicates that the test instance has been executed.
	Pending: Indicates the test instance has not been picked up for execution yet because it is future-dated.
	Executed Group Definition
Search	Click to execute a search for scheduled definitions using the search criteria entered in the **Search Scheduled Tests** group box.

Search Result Selection

Delete	Select the **Select** option for one or more scheduled tests and click **Delete** to delete the scheduled test.
Id	Click a test definition ID link to access the Modify Test Definition page.
	For more information, see Modifying a Test Definition.
	Click a group definition ID link to access the Group Definition Detail page.
	For more information, see Creating and Modifying a Group Definition.
Scheduled Run Time	Date and time at which the test is scheduled to run.
InstanceId	Click for a scheduled test definition that has been executed to access the Test Instances Detail page.
	For more information, see Viewing Test Instance Details.
	Click for a scheduled group definition that has been executed to access the Group Instances Detail page.
	For more information, see Viewing Group Instance Details.
View Result	If the execution of the scheduled test involved a single test, this field is blank. If the execution of the scheduled test

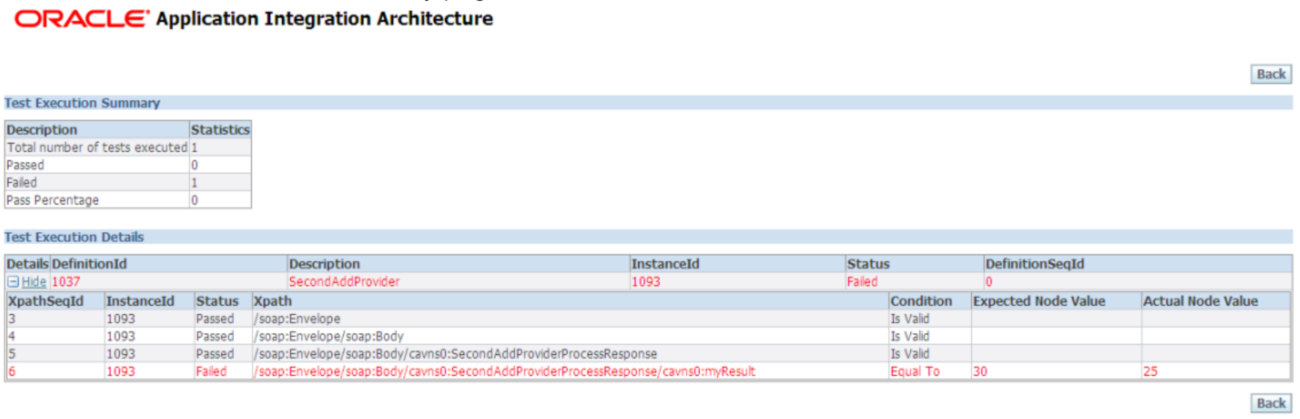
Copyright © 2009, Oracle. All rights reserved.

involved a group definition, click the **View** link to access result details on the Test Execution Summary page.

For more information, see Viewing Scheduled Test Group Results.

Viewing Scheduled Test Group Results

Access the Test Execution Summary page.


ORACLE Application Integration Architecture

Test Execution Summary

Description	Statistics
Total number of tests executed	1
Passed	0
Failed	1
Pass Percentage	0

Test Execution Details

Details	DefinitionId	Description	InstanceId	Status	DefinitionSeqId
Hide	1037	SecondAddProvider	1093	Failed	0

XpathSeqId	InstanceId	Status	Xpath	Condition	Expected Node Value	Actual Node Value
3	1093	Passed	/soap:Envelope	Is Valid		
4	1093	Passed	/soap:Envelope/soap:Body	Is Valid		
5	1093	Passed	/soap:Envelope/soap:Body/cavns0:SecondAddProviderProcessResponse	Is Valid		
6	1093	Failed	/soap:Envelope/soap:Body/cavns0:SecondAddProviderProcessResponse/cavns0:myResult	Equal To	30	25

Test Execution Summary page

Description

Total Number of Tests Executed: Displays the total number of tests executed as a part of the scheduled test group.

Passed: Displays the number of tests that passed when executed as a part of the scheduled test group.

Failed: Displays the number of tests that failed when executed as a part of the scheduled test group.

Pass Percentage: Displays the percentage of tests that passed when executed as a part of the scheduled test group.

Test Execution Details

Details

Click the **Show** link to display details about the execution of the test definition that was run as a part of the scheduled test group. Click the **Hide** link to hide details about the execution of the test definition.

Details include the results of any response message validations defined for the test in the XPath Selection grid on the Modify Test Definitions page.

For more information about the Modify Test Definitions page, see Modifying a Test Definition.

DefinitionId

Displays the ID of the test definition that was executed.

InstanceId	Displays the ID of the instance that was created by the execution of the test definition.
Status	Displays the status of the test instance.
DefinitionSeqId	Displays the sequence in which the test definition was initiated in the scheduled test group.

Copyright © 2009, Oracle. All rights reserved.

Chapter 17: Working with Test and Simulator Instances

This chapter discusses how to work with test and simulator instances.

Working with Instances

A test instance captures the details of the execution of a test definition. A simulator instance captures the details of a simulator definition's behavior during the execution of a test definition with which it is associated.

This section discusses how to:

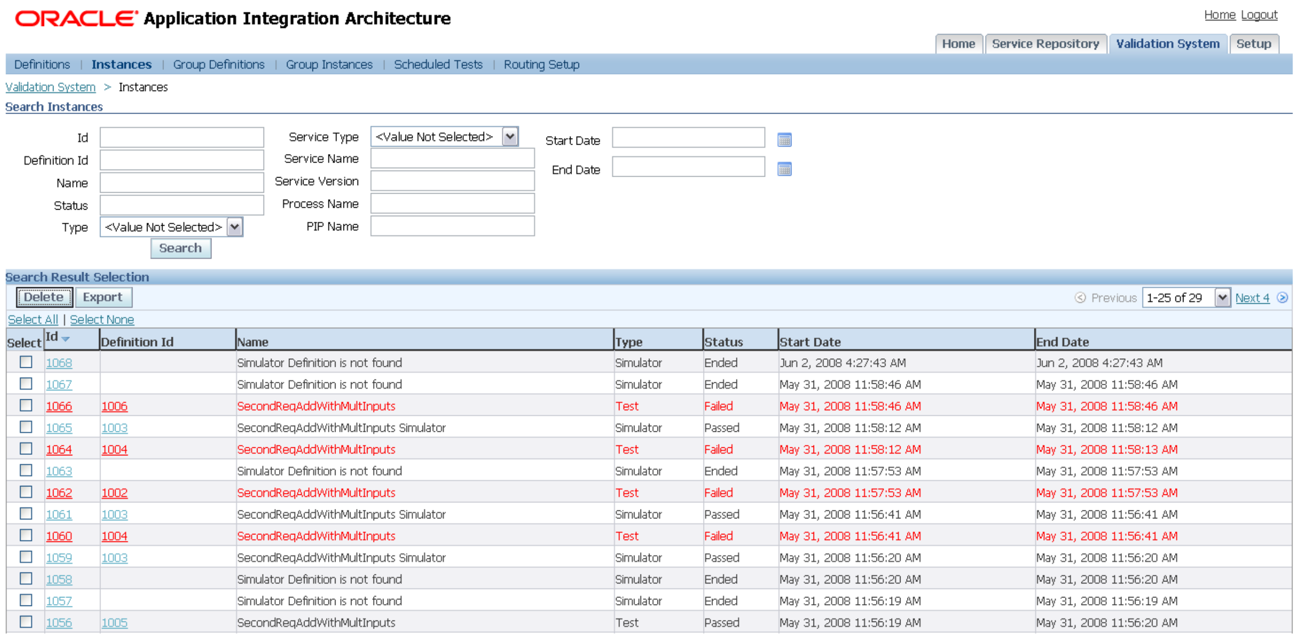
- Work with test and simulator instances.
- View test instance details.
- View simulator instance details.

Pages Used to Work with Instances

Page Name	Navigation	Usage
Instances	Access the Oracle AIA Console. Select the Validation System tab. Click the Instances link.	Search for test and simulator instances. Access pages you can use to view test and simulator instance details.
Test Instances Detail	• Select Execute in the Actions drop-down list box and click Go on the Modify Test Definition page. • Click an Instance ID link on the Instances page. • Click an Instance ID link in the Test Instances group box on the Modify Test Definition page.	View the details of a test instance.
Simulator Instances Detail	Click the Instance Id link for a simulator instance on the Instances page.	View the details of a simulator instance.

Working with Test and Simulator Instances

Access the Instances page.



ORACLE Application Integration Architecture

Home Logout

Home | Service Repository | Validation System | Setup

Definitions | **Instances** | Group Definitions | Group Instances | Scheduled Tests | Routing Setup

Validation System > Instances

Search Instances

Id		Service Type	<Value Not Selected>	Start Date	
Definition Id		Service Name		End Date	
Name		Service Version			
Status		Process Name			
Type	<Value Not Selected>	PIP Name			

Search

Search Result Selection

Delete Export — Previous 1-25 of 29 Next 4

Select All | Select None

Select	Id	Definition Id	Name	Type	Status	Start Date	End Date
☐	1068		Simulator Definition is not found	Simulator	Ended	Jun 2, 2008 4:27:43 AM	Jun 2, 2008 4:27:43 AM
☐	1067		Simulator Definition is not found	Simulator	Ended	May 31, 2008 11:58:46 AM	May 31, 2008 11:58:46 AM
☐	1066	1006	SecondReqAddWithMultInputs	Test	Failed	May 31, 2008 11:58:46 AM	May 31, 2008 11:58:46 AM
☐	1065	1003	SecondReqAddWithMultInputs Simulator	Simulator	Passed	May 31, 2008 11:58:12 AM	May 31, 2008 11:58:12 AM
☐	1064	1004	SecondReqAddWithMultInputs	Test	Failed	May 31, 2008 11:58:12 AM	May 31, 2008 11:58:13 AM
☐	1063		Simulator Definition is not found	Simulator	Ended	May 31, 2008 11:57:53 AM	May 31, 2008 11:57:53 AM
☐	1062	1002	SecondReqAddWithMultInputs	Test	Failed	May 31, 2008 11:57:53 AM	May 31, 2008 11:57:53 AM
☐	1061	1003	SecondReqAddWithMultInputs Simulator	Simulator	Passed	May 31, 2008 11:56:41 AM	May 31, 2008 11:56:41 AM
☐	1060	1004	SecondReqAddWithMultInputs	Test	Failed	May 31, 2008 11:56:41 AM	May 31, 2008 11:56:41 AM
☐	1059	1003	SecondReqAddWithMultInputs Simulator	Simulator	Passed	May 31, 2008 11:56:20 AM	May 31, 2008 11:56:20 AM
☐	1058		Simulator Definition is not found	Simulator	Ended	May 31, 2008 11:56:20 AM	May 31, 2008 11:56:20 AM
☐	1057		Simulator Definition is not found	Simulator	Ended	May 31, 2008 11:56:19 AM	May 31, 2008 11:56:19 AM
☐	1056	1005	SecondReqAddWithMultInputs	Test	Passed	May 31, 2008 11:56:19 AM	May 31, 2008 11:56:20 AM

Instances page

Search Instances

Use the **Search Instances** group box to enter search criteria to locate the instance you are searching for.

Id	Enter the unique key assigned to the instance.
Definition Id	Enter a definition ID associated with the definition that generated the instance.
Name	Enter the descriptive name given to the definition that generated the instance.
Status	Enter the status of the instance.
	Ended: This status is only applicable to instances that do not involve validations. Indicates that the instance has ended.
	Faulted: The instance could not execute properly due to exceptions or faults.
	Failed: The instance did not pass validation.
	Passed: The instance passed validation.
	Delayed: For an asynchronous two-way test instance, indicates that the test instance is still active and waiting for an asynchronous reply.
Type	Select the type of instance.
	<Value Not Selected>

Copyright © 2009, Oracle. All rights reserved.

	Test
	Simulator
Service Type	Select the business service pattern of the web service associated with the instance. For example, if you are searching for a test instance, this is the business service pattern of the web service tested by the test definition that generated the test instance. If you are searching for a simulator instance, this is the business service pattern of the web service simulated by the simulator definition that generated the simulator instance. ***<Value Not Selected>*** ***Synchronous*** ***Notify*** ***Asynchronous two way***
Service Name	Enter the name of the web service associated with the definition that created the instance.
Service Version	Enter the version of the web service associated with the definition that created the instance.
Process Name	Enter the name of the process associated with the definition that created the instance.
PIP Name (Process Integration Pack name)	Enter the name of the Process Integration Pack associated with the definition that created the instance.
Start Date	Enter a start date and time that you want to use as search criteria. The search will look for all instances that were created on and after the given date and time.
End Date	Enter an end date and time that you want to use as search criteria. The search will look for all instances that were created before and on the given date and time.
Search	Click to execute a search for instances using the search criteria entered in the Search Instances group box.

Search Result Selection

Use the **Search Result Selection** grid to work with instances returned in your search results. Upon accessing this page, the grid is populated by all instances.

Export	**For more information** about exporting instances, see Exporting and Importing Definitions and Instances.
Delete	Select the **Select** check box for one or more instances that you want to delete and click the **Delete** button to execute the deletion.
Id	Click for a test instance to access the Test Instance Detail page.

Definition Id

Click for a simulator instance to access the Simulator Instance Detail page.

For a test instance, click to access details about the test definition that generated the test instance. An unlocked test definition displays on the Modify Test Definition page. A locked test definition displays on the View Test Definition page.

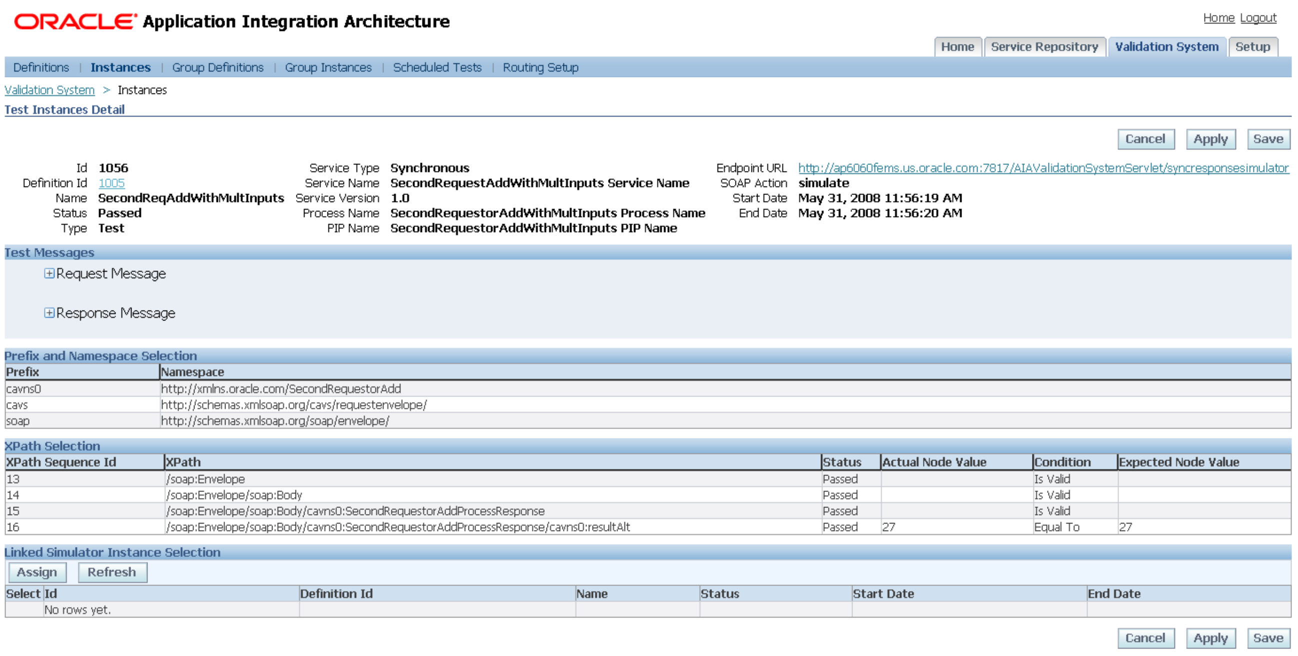
For more information, see Creating and Modifying Test Definitions.

For a simulator instance, click to access details about the simulator definition that generated the simulator instance. An unlocked simulator definition displays on the Modify Simulator Definition page. A locked test definition displays on the View Simulator Definition page.

For more information, see Creating and Modifying Simulator Definitions.

Viewing Test Instance Details

Access the Test Instances Detail page.


ORACLE Application Integration Architecture
Home Logout
Home
Service Repository
Validation System
Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup
Validation System > Instances
Cancel
Apply
Save
Request Message
Response Message
Assign
Refresh
Cancel
Apply
Save

Test Instances Detail

Id	1056	Service Type	Synchronous	Endpoint URL	http://ap6060fems.us.oracle.com:7817/AIAValidationSystemServlet/syncresponsesimulator
Definition Id	1005	Service Name	SecondRequestAddWithMultInputs Service Name	SOAP Action	simulate
Name	SecondReqAddWithMultInputs	Service Version	1.0	Start Date	May 31, 2008 11:56:19 AM
Status	Passed	Process Name	SecondRequestorAddWithMultInputs Process Name	End Date	May 31, 2008 11:56:20 AM
Type	Test	PIP Name	SecondRequestorAddWithMultInputs PIP Name		

Test Messages

Prefix and Namespace Selection

Prefix	Namespace
cavns0	http://xmlns.oracle.com/SecondRequestorAdd
cavs	http://schemas.xmlsoap.org/cavs/requestenvelope/
soap	http://schemas.xmlsoap.org/soap/envelope/

XPath Selection

XPath Sequence Id	XPath	Status	Actual Node Value	Condition	Expected Node Value
13	/soap:Envelope	Passed		Is Valid	
14	/soap:Envelope/soap:Body	Passed		Is Valid	
15	/soap:Envelope/soap:Body/cavns0:SecondRequestorAddProcessResponse	Passed		Is Valid	
16	/soap:Envelope/soap:Body/cavns0:SecondRequestorAddProcessResponse/cavns0:resultAlt	Passed	27	Equal To	27

Linked Simulator Instance Selection

Select	Id	Definition Id	Name	Status	Start Date	End Date
	No rows yet.					

Test Instances Detail page

Cancel

Click to discard any updates to the page and return to the Instances page.

Copyright © 2009, Oracle. All rights reserved.

Apply	Click to apply and save any updates you have made to the page.
Save	Click to save any updates you have made to the page and go to the Instances page.
Id	Displays the unique ID assigned to the instance.
Definition Id	Displays the ID of the test definition that generated the test instance. Click for an unlocked test definition to access the Modify Test Definition page. Click for a locked test definition to access the View Test Definition page. **For more information**, see Creating and Modifying Test Definitions.
Name	Displays the descriptive name associated with the test definition that generated the instance.
Status	Displays the status of the test instance. ***Ended:*** This status is only applicable to test instances that do not involve validations. Indicates that the instance has ended. ***Faulted:*** The test instance could not execute properly due to exceptions or faults. ***Failed:*** The test instance did not pass validation. ***Passed:*** The instance passed validation. ***Delayed:*** For an asynchronous two-way test instance, indicates that the test instance is still active and waiting for an asynchronous reply.
Type	Displays the type of definition that generated the test instance. On the Test Instances Detail page, this value will always be ***Test.***
Service Type	Displays the business service pattern of the web service tested by the test definition that generated the test instance. ***Synchronous*** ***Notify*** ***Asynchronous two way***
Service Name	Displays the name of the web service tested by the test definition that created the instance.
Service Version	Displays the version of the web service tested by the test definition that created the instance.
Process Name	Displays the name of the process associated with the test definition that created the instance.
PIP Name (Process	Displays the name of the Process Integration Pack associated

Integration Pack name)	with the test definition that created the instance.
Endpoint URL	Displays the URL of the web service tested by the test definition that created the instance.
SOAP Action	Displays the operation called by the web service tested by the test definition that created the instance.
Start Date	Displays the date and time at which the test instance was initiated.
End Date	Displays the date and time at which the test instance ended.

Test Messages

Use the **Test Messages** group box to view the request and response XML messages associated with the test definition that generated the instance.

Request Message	Displays request message XML defined for the test definition that generated the test instance. **For more information** about the **Request Message** field, see Creating and Modifying Test Definitions.
Response Message	Displays response message XML defined for the test definition that generated the test instance. **For more information** about the **Response Message** field, see Creating and Modifying Test Definitions.

Prefix and Namespace Selection

Displays namespace data created for the test definition that generated the test instance. This namespace data is used in the XPath values defined in the **XPath Selection** grid.

For more information about the **Prefix and Namespace Selection** grid, see Creating and Modifying Test Definitions.

XPath Selection

Displays XPath data created for the test definition that generated the test instance. The values in this grid use the namespace values set in the **Prefix and Namespace Selection** grid.

For more information about the **XPath Selection** grid, see Creating and Modifying Test Definitions.

Linked Simulator instance Selection

Use the **Linked Simulator instance Selection** grid to work with associations between test instances and simulator instances.

Copyright © 2009, Oracle. All rights reserved.

If no correlation logic has been defined between the test definition and the simulator definition, the test and simulator instances will not always be reconcilable, especially when the same web service is invoked multiple times during a very short time period, as in during performance testing.

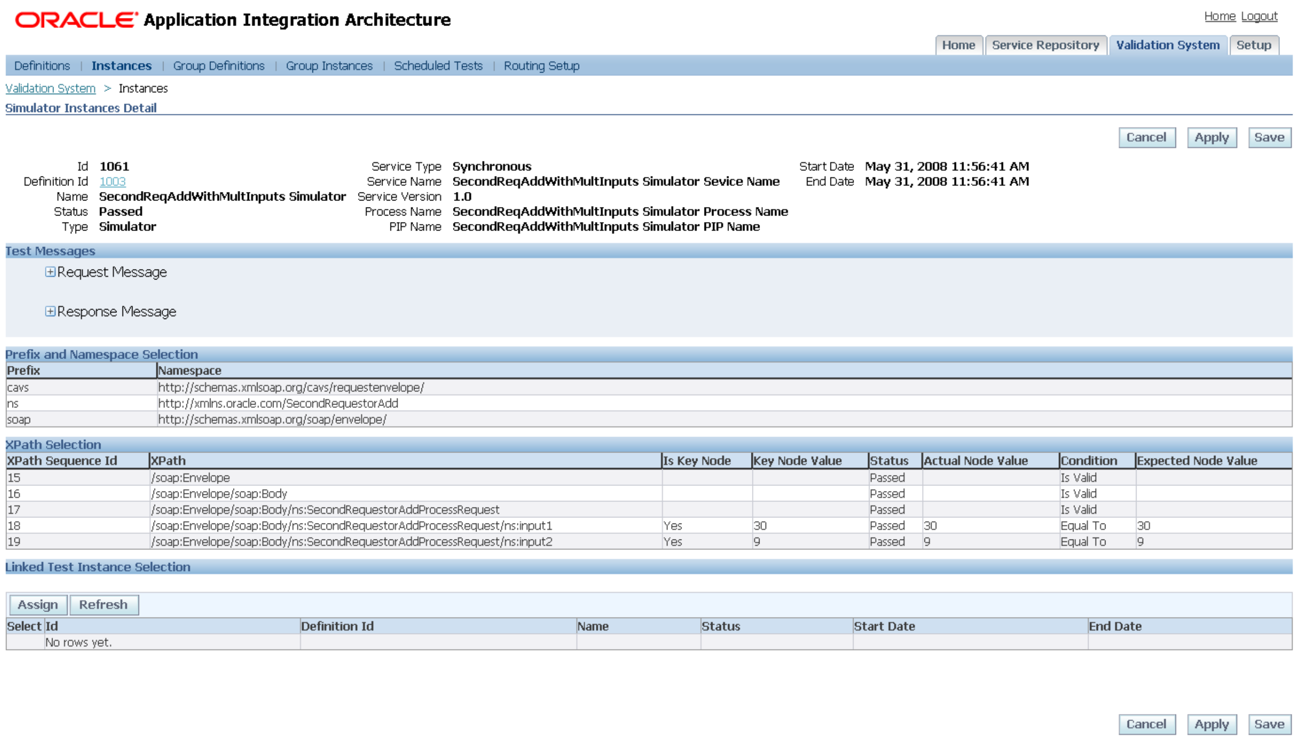
However, if a simulator definition is associated with a test definition, any test instances generated by the test definition will automatically reflect associations to simulator instances generated by associated simulator definitions.

You can manually adjust these associations in this grid area.

Unassign	Select the Select check box for one or more simulator instance rows that you want to disassociate with the test instance. Click the Unassign button to execute the disassociation.
Assign	Click to access the Search Instances - Simulator page, where you can search for a simulator instance that you want to manually associate with the test instance. Once you have associated a simulator instance with the test instance using the Search Instances - Simulator page, the Test Instances Detail page displays the selected simulator instance.
Refresh	Click to refresh the Test Instances Detail page.
Id	Click to access the Simulator Instances Detail page.
Definition ID	Click to view details about the test definition that generated the test instance. An unlocked test definitions display on the Modify Test Definition page. A locked test definition displays on the View Test Definition page. **For more information**, see Creating and Modifying Test Definitions.

Viewing Simulator Instance Details

Access the Simulator Instances Detail page.


ORACLE Application Integration Architecture
Home Logout
Home
Service Repository
Validation System
Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup
Validation System > Instances
Cancel
Apply
Save
Assign
Refresh

Simulator Instances Detail

Id	1061	Service Type	Synchronous	Start Date	May 31, 2008 11:56:41 AM
Definition Id	1003	Service Name	SecondReqAddWithMultInputs Simulator Sevice Name	End Date	May 31, 2008 11:56:41 AM
Name	SecondReqAddWithMultInputs Simulator	Service Version	1.0		
Status	Passed	Process Name	SecondReqAddWithMultInputs Simulator Process Name		
Type	Simulator	PIP Name	SecondReqAddWithMultInputs Simulator PIP Name		

Test Messages

⊞Request Message

⊞Response Message

Prefix and Namespace Selection

Prefix	Namespace
cavs	http://schemas.xmlsoap.org/cavs/requestenvelope/
ns	http://xmlns.oracle.com/SecondRequestorAdd
soap	http://schemas.xmlsoap.org/soap/envelope/

XPath Selection

XPath Sequence Id	XPath	Is Key Node	Key Node Value	Status	Actual Node Value	Condition	Expected Node Value
15	/soap:Envelope			Passed		Is Valid	
16	/soap:Envelope/soap:Body			Passed		Is Valid	
17	/soap:Envelope/soap:Body/ns:SecondRequestorAddProcessRequest			Passed		Is Valid	
18	/soap:Envelope/soap:Body/ns:SecondRequestorAddProcessRequest/ns:input1	Yes	30	Passed	30	Equal To	30
19	/soap:Envelope/soap:Body/ns:SecondRequestorAddProcessRequest/ns:input2	Yes	9	Passed	9	Equal To	9

Linked Test Instance Selection

Select	Id	Definition Id	Name	Status	Start Date	End Date
	No rows yet.					

Simulator Instances Detail page

Cancel	Click to discard any updates to the page and return to the Instances page.
Apply	Click to apply and save any updates you have made to the page.
Save	Click to save any updates you have made to the page and go to the Instances page.
Id	Displays the unique ID assigned to the instance.
Definition ID	Click for an unlocked simulator definition to access the Modify Simulator Definition page. Click for a locked simulator definition to access the View Simulator Definition page.
Name	Displays the descriptive name associated with the simulator definition that generated the instance.
Status	Displays the status of the simulator instance. ***Initiated:*** The simulator instance has been initiated. ***Ended:*** This status is only applicable to simulator instances that do not involve validations. Indicates that the instance has ended. ***Faulted:*** The simulator instance could not execute properly due to exceptions or faults. ***Failed:*** The simulator instance did not pass validation. ***Passed:*** The simulator instance passed validation.

Copyright © 2009, Oracle. All rights reserved.

Type	Displays the type of definition that generated the simulator instance. On the Simulator Instances Detail page, this value will always be ***Simulator.***
Service Type	Displays the business service pattern of the web service simulated by the simulator definition that generated the instance. ***Synchronous*** ***Notify*** ***Asynchronous two way***
Service Name	Displays the name of the web service simulated by the simulator definition that created the instance.
Service Version	Displays the version of the web service simulated by the simulator definition that created the instance.
Process Name	Displays the name of the process associated with the simulator definition that created the instance.
PIP Name (Process Integration Pack name)	Displays the name of the Process Integration Pack associated with the simulator definition that created the instance.
Start Date	Displays the date and time at which the simulator instance was initiated.
End Date	Displays the date and time at which the simulator instance ended.

Test Messages

Use the **Test Messages** group box to view the request and response XML messages associated with the simulator definition that generated the instance.

Request Message	Displays request message XML defined for the simulator definition that generated the instance. **For more information** about the **Request Message** field, see Creating and Modifying Simulator Definitions.
Response Message	Displays response message XML defined for the simulator definition that generated the instance. **For more information** about the **Response Message** field, see Creating and Modifying Simulator Definitions.

Prefix and Namespace Selection

Displays namespace data created for the simulator definition that generated the simulator instance. This namespace data is used in the XPath values defined in the **XPath Selection** grid.

For more information about the **Prefix and Namespace Selection** grid, see Creating and Modifying Simulator Definitions.

XPath Selection

Displays XPath data created for the simulator definition that generated the instance. The values in this grid use the namespace values set in the **Prefix and Namespace Selection** grid.

For more information about the **XPath Selection** grid, see Creating and Modifying Simulator Definitions.

Linked Test Instance Selection

Displays the test instance with which the simulator instance is associated. This is a one-to-one association.

If no correlation logic has been defined between the test definition and the simulator definition, the test and simulator instances will not always be reconcilable, especially when the same web service is invoked multiple times during a very short time period, as in during performance testing.

However, if a simulator definition is associated with a test definition, any test instances generated by the test definition will automatically reflect associations to simulator instances generated by associated simulator definitions.

You can adjust the association between the simulator instance and a test instance using the controls on this page.

Unassign	Select the **Select** option for the test instance ID that you want to disassociate from the simulator instance and click the **Unassign** button to execute the disassociation.
Assign	Click to access the Search Instances - Test page, where you can search for a test instance that you want to manually associate with the simulator instance. Once you have associated a test instance with the simulator instance using the Search Instances - Test page, the Simulator Instances Detail page displays the selected test instance.
Refresh	Click to refresh the Simulator Instances Detail page.
Id	Click to display the selected test instance on the Test Instances Detail page
Definition Id	Displays the ID of the test definition that generated the test instance. Click for an unlocked test definition to access the Modify Test Definition page. Click for a locked test definition to access the View Test Definition page. **For more information**, see Creating and Modifying Test

Copyright © 2009, Oracle. All rights reserved.

Definitions.

Chapter 18: Working with Group Instances

This chapter discusses how to work with group instances.

Working with Group Instances

A group instance captures the details of the execution of a group definition.

This section discusses how to:

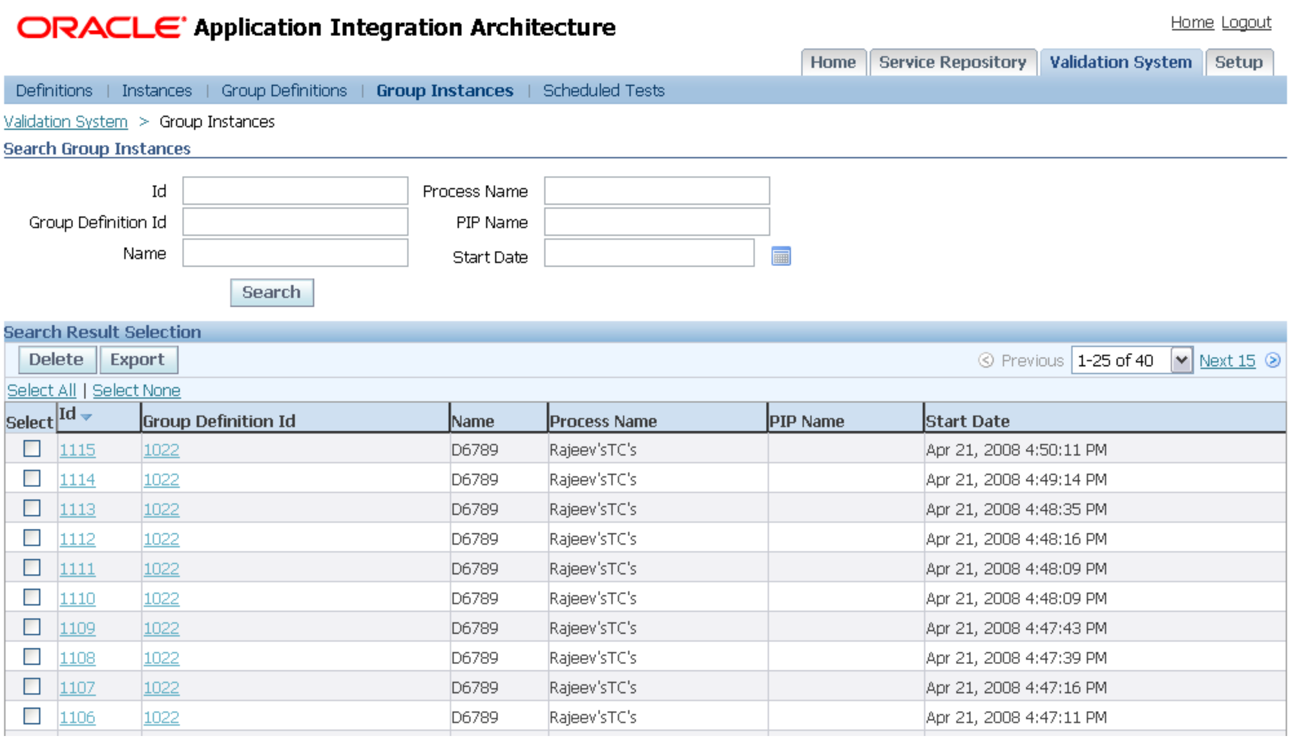
- View group instances.
- View group instance details.

Pages Used to View Group Instances

Page Name	Navigation	Usage
Group Instances	Access the Oracle AIA Console. Select the Validation System tab. Click the Group Instances link.	Search for group instances. Access a page you can use to view group instance details.
Group Instances Detail	Click a Group Instance ID link on the Group Instances page.	View the details of a group instance.

Viewing Group Instances

Access the Group Instances page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests
Validation System > Group Instances
Search Group Instances
Id
Group Definition Id
Name
Process Name
PIP Name
Start Date
Search
Search Result Selection
Delete | Export
Previous 1-25 of 40 Next 15
Select All | Select None

Select	Id	Group Definition Id	Name	Process Name	PIP Name	Start Date
☐	1115	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:50:11 PM
☐	1114	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:49:14 PM
☐	1113	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:48:35 PM
☐	1112	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:48:16 PM
☐	1111	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:48:09 PM
☐	1110	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:48:09 PM
☐	1109	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:47:43 PM
☐	1108	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:47:39 PM
☐	1107	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:47:16 PM
☐	1106	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:47:11 PM

Group Instances page

Search Group Instances

Use the **Search Group Instances** group box to enter search criteria to find the group instance you are searching for.

Id	Enter the unique key identifier assigned to the group instance.
Group Definition Id	Enter the unique key ID assigned to the group definition that generated the instance.
Name	Enter a descriptive name assigned to the group definition.
Process Name	Enter the name of the process associated with the group definition that generated the instance.
PIP Name (process integration pack)	Enter the name of the PIP associated with the group definition that generated the instance.
Start Date	Enter a start date and time that you want to use as search criteria. The search will look for all group instances that were created on and after the given date and time.
Search	Click to execute a search for group instances using the search criteria entered in the **Search Group Instances** group box.

Search Result Selection

Use the **Search Result Selection** grid to work with group instances returned in your search results. Upon accessing this page, the grid is populated by all group instances.

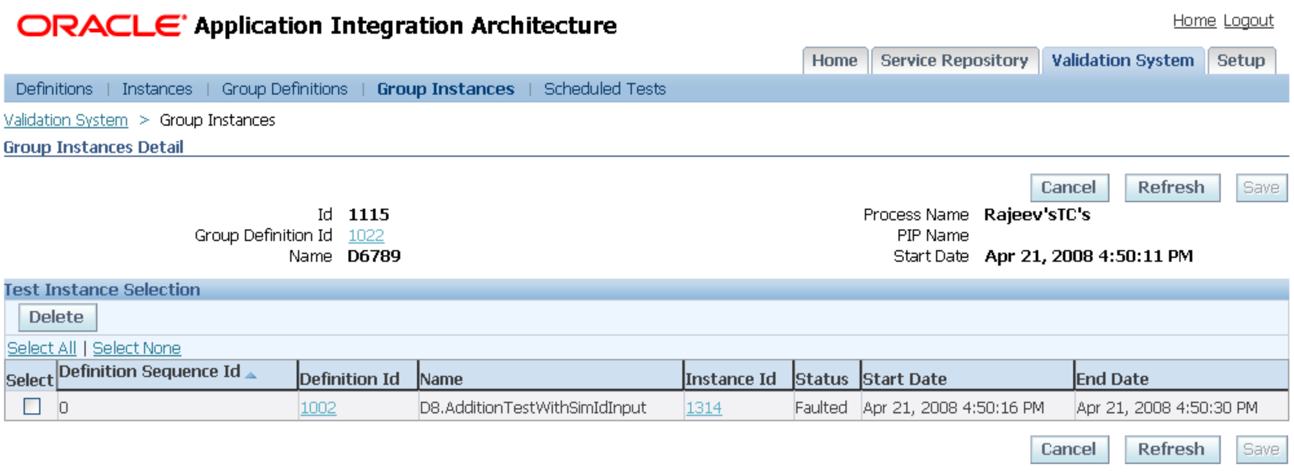
Delete	Select the **Select** check box for one or more group instances that you want to delete and click the **Delete** button to execute

Copyright © 2009, Oracle. All rights reserved.

	the deletion.
Export	**For more information** exporting group instances, see Exporting and Importing Definitions and Instances.
Id	Click to access the Group Instances Detail page.
Group Definition Id	Click to access the Group Definition Detail page. **For more information** about the Group Definition Detail page, see Working with Group Definitions.

Viewing Group Instance Details

Access the Group Instances Detail page.



Id	1115
Group Definition Id	1022
Name	D6789
Process Name	Rajeev'sTC's
PIP Name	
Start Date	Apr 21, 2008 4:50:11 PM

Select	Definition Sequence Id	Definition Id	Name	Instance Id	Status	Start Date	End Date
☐	0	1002	D8.AdditionTestWithSimIdInput	1314	Faulted	Apr 21, 2008 4:50:16 PM	Apr 21, 2008 4:50:30 PM

Group Instances Detail page

Id	Displays the unique key identifier assigned to the group instance.
Group Definition Id	Click to access the Group Definition Detail page.
Name	Displays the descriptive name assigned to the group definition.
Process Name	Displays the name of the process associated with the group definition that generated the instance.
PIP Name (process integration pack)	Enter the name of the PIP associated with the group definition that generated the instance.
Start Date	Displays the date and time at which the group instance was initiated.

Test instance Selection

Use the **Test Instance Selection** grid to access information about test instances associated with the group instance.

Delete	Select the **Select** check box for one or more test instance rows that you want to delete and click the **Delete** button to execute the deletion.
Definition Sequence Id	Indicates the sequence in which the test definitions were initiated by the group definition that generated the group instance.
Definition Id	Click to access the Modify Test Definition page. **For more information** about the Modify Test Definition page, see Creating and Modifying Test Definitions.
Instance Id	Click to access the Test Instances Detail page. **For more information** about the Test Instances Detail page, see Viewing Test Instance Details.
Status	Displays the status of the test instance in the group instance. ***Initiated:*** The test instance has been initiated. ***Ended:*** This status is only applicable to test instances that do not involve validations. Indicates that the instance has ended. ***Faulted:*** The test instance could not execute properly due to exceptions or faults. ***Failed:*** The test instance did not pass validation. ***Passed:*** The instance passed validation.
Start Date	Displays the date and time at which the test instance was initiated.
End Time	Displays the date and time at which the test instance ended.

Copyright © 2009, Oracle. All rights reserved.

Chapter 19: Exporting and Importing CAVS Definitions and Instances

This chapter discusses the export and import of Composite Application Validation System (CAVS) definitions and instances.

Exporting and Importing Definitions and Instances

This section discusses how to:

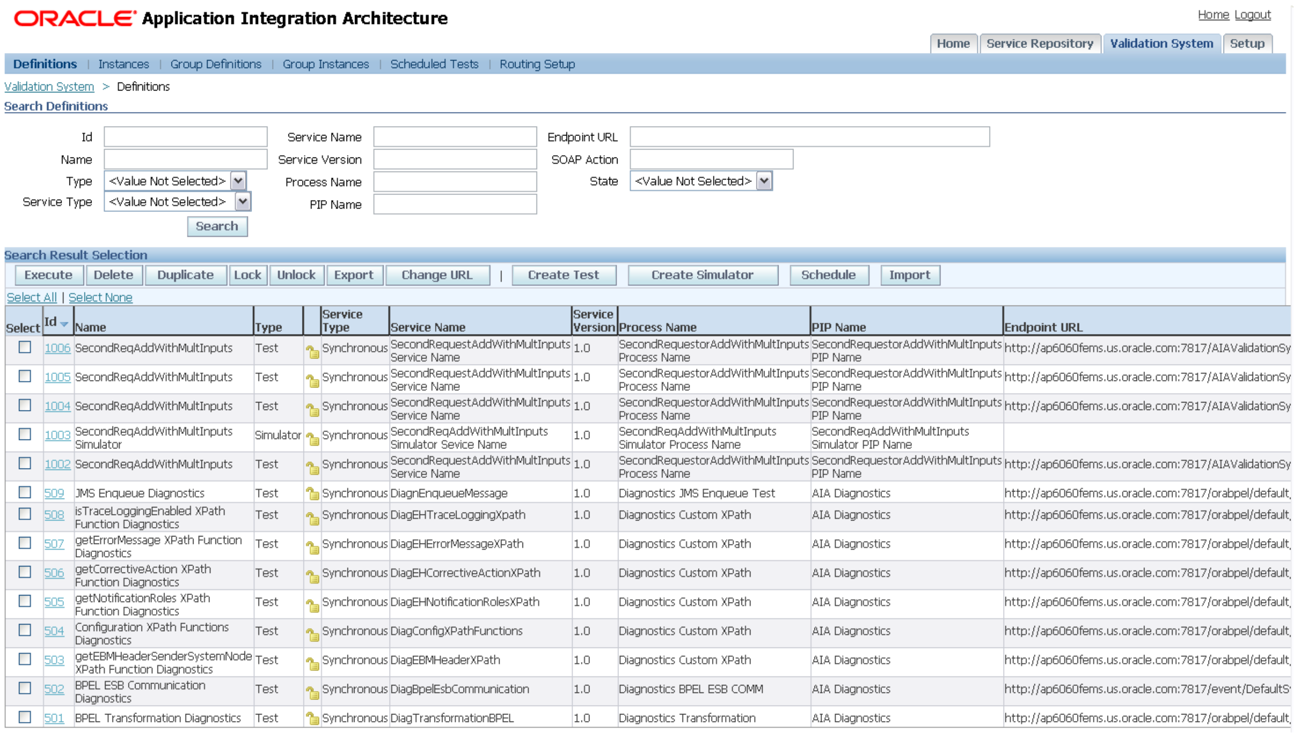
- Export and import definitions.
- Export test and simulator instances.
- Export group instances.

Pages Used to Export and Import Definitions and Instances

Page Name	Navigation	Usage
Definitions	Access the Oracle AIA Console. Select the Validation System tab. Click the Definitions link.	Search for, execute, migrate, and manage existing test and simulator definitions. Access pages you can use to create and modify test and simulator definitions.
Instances	Access the Oracle AIA Console. Select the Validation System tab. Click the Instances link.	Search for test and simulator instances. Access pages you can use to view test and simulator instance details.
Group Instances	Access the Oracle AIA Console. Select the Validation System tab. Click the Group Instances link.	Search for group instances. Access a page you can use to view group instance details.

Exporting and Importing Definitions

Access the Definitions page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests | Routing Setup
Validation System > Definitions
Search Definitions
Search
Search Result Selection
Execute | Delete | Duplicate | Lock | Unlock | Export | Change URL | Create Test | Create Simulator | Schedule | Import
Select All | Select None

Id		Service Name		Endpoint URL	
Name		Service Version		SOAP Action	
Type	<Value Not Selected>	Process Name		State	<Value Not Selected>
Service Type	<Value Not Selected>	PIP Name			

Select	Id	Name	Type		Service Type	Service Name	Service Version	Process Name	PIP Name	Endpoint URL
☐	1006	SecondReqAddWithMultInputs	Test	🔓	Synchronous	SecondRequestAddWithMultInputs Service Name	1.0	SecondRequestorAddWithMultInputs Process Name	SecondRequestorAddWithMultInputs PIP Name	http://ap6060fems.us.oracle.com:7817/AIAValidationSy
☐	1005	SecondReqAddWithMultInputs	Test	🔓	Synchronous	SecondRequestAddWithMultInputs Service Name	1.0	SecondRequestorAddWithMultInputs Process Name	SecondRequestorAddWithMultInputs PIP Name	http://ap6060fems.us.oracle.com:7817/AIAValidationSy
☐	1004	SecondReqAddWithMultInputs	Test	🔓	Synchronous	SecondRequestAddWithMultInputs Service Name	1.0	SecondRequestorAddWithMultInputs Process Name	SecondRequestorAddWithMultInputs PIP Name	http://ap6060fems.us.oracle.com:7817/AIAValidationSy
☐	1003	SecondReqAddWithMultInputs Simulator	Simulator	🔓	Synchronous	SecondReqAddWithMultInputs Simulator Sevice Name	1.0	SecondReqAddWithMultInputs Simulator Process Name	SecondReqAddWithMultInputs Simulator PIP Name	
☐	1002	SecondReqAddWithMultInputs	Test	🔓	Synchronous	SecondRequestAddWithMultInputs Service Name	1.0	SecondRequestorAddWithMultInputs Process Name	SecondRequestorAddWithMultInputs PIP Name	http://ap6060fems.us.oracle.com:7817/AIAValidationSy
☐	509	JMS Enqueue Diagnostics	Test	🔓	Synchronous	DiagnEnqueueMessage	1.0	Diagnostics JMS Enqueue Test	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default
☐	508	isTraceLoggingEnabled XPath Function Diagnostics	Test	🔓	Synchronous	DiagEHTraceLoggingXpath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default
☐	507	getErrorMessage XPath Function Diagnostics	Test	🔓	Synchronous	DiagEHErrorMessageXPath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default
☐	506	getCorrectiveAction XPath Function Diagnostics	Test	🔓	Synchronous	DiagEHCorrectiveActionXPath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default
☐	505	getNotificationRoles XPath Function Diagnostics	Test	🔓	Synchronous	DiagEHNotificationRolesXPath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default
☐	504	Configuration XPath Functions Diagnostics	Test	🔓	Synchronous	DiagConfigXPathFunctions	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default
☐	503	getEBMHeaderSenderSystemNode XPath Function Diagnostics	Test	🔓	Synchronous	DiagEBMHeaderXPath	1.0	Diagnostics Custom XPath	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default
☐	502	BPEL ESB Communication Diagnostics	Test	🔓	Synchronous	DiagBpelEsbCommunication	1.0	Diagnostics BPEL ESB COMM	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/event/DefaultS
☐	501	BPEL Transformation Diagnostics	Test	🔓	Synchronous	DiagTransformationBPEL	1.0	Diagnostics Transformation	AIA Diagnostics	http://ap6060fems.us.oracle.com:7817/orabpel/default

Definitions page

Export

Use the **Export** and **Import** buttons on this page to migrate test definitions, simulator definitions, and any associated group definitions in XML flat-file format between instances running on the same version of Foundation Pack.

Examples of uses for this export and import functionality include:

- QA may want to certify a set of definitions that have been run in one build in other builds.
- Support analysts and customers may want to exchange definition files.
- You may want to verify validity of new environments.

Select the **Select** check box for one or more definitions and click the **Export** button to initiate the export. The following options display:

Export selected Definition(s) only

Export selected Definition(s) and associated Group Definition(s)

Export selected Definition(s), associated GroupDefinition(s) and Test Definition(s) that belong to

Copyright © 2009, Oracle. All rights reserved.

the associated GroupDefinition(s) but are not selected

Select an option and click the **Proceed** button to create and save the definitions to a location on your local system. The default file name for the exported definition(s) is ***Definitions.xml.***

If a test definition that you are exporting is associated with a routing setup ID, the routing setup information will also be exported.

If that routing setup is associated with one or more simulator definitions, which were provided when the Route To CAVS option was set to true, then these simulator definitions will also be exported.

For more information about the structure of the Definitions.xml file created by the CAVS export definition feature, see Appendix A: XML Structures of Exportable CAVS Definitions and Instances.

Import

Use the **Import** button to upload a test, simulator, or group definition in the XML flat-file format generated by CAVS export functionality. You can generate these files by clicking the **Export** button on this page. The definition file to be uploaded must be accessible by the local system being used to perform the upload.

Click the **Import** button and browse for the file you want to upload. The CAVS validates the structure of the file being uploaded. If the structure is invalid, an error will be raised.

If a test definition that you are importing is associated with a routing setup ID, the routing setup information will also be imported.

If that routing setup is associated with one or more simulator definitions, which were provided when the Route To CAVS option was set to true, then these simulator definitions will also be imported.

For more information about the valid structure of the Definitions.xml file created by the CAVS export definition feature, see Appendix A: XML Structures of Exportable CAVS Definitions and Instances.

Imported definitions will still reference endpoint URLs pointing to tested web services in the source system. You must update imported definition endpoint URL values to point to tested web services in the target system. The CAVS enables you to update these URLs directly on the following pages:

Use the **Change URL** button on this page.

Update the Endpoint URL field value on the Modify Test

Definitions page.

For more information about the Endpoint URL field, see Creating and Modifying Test Definitions.

Because the sequential definition IDs assigned in the source system may not be valid in the target system, new sequential definition IDs will be assigned by the target system. As a result, associations between definitions will be severed in the target system and will need to be reestablished.

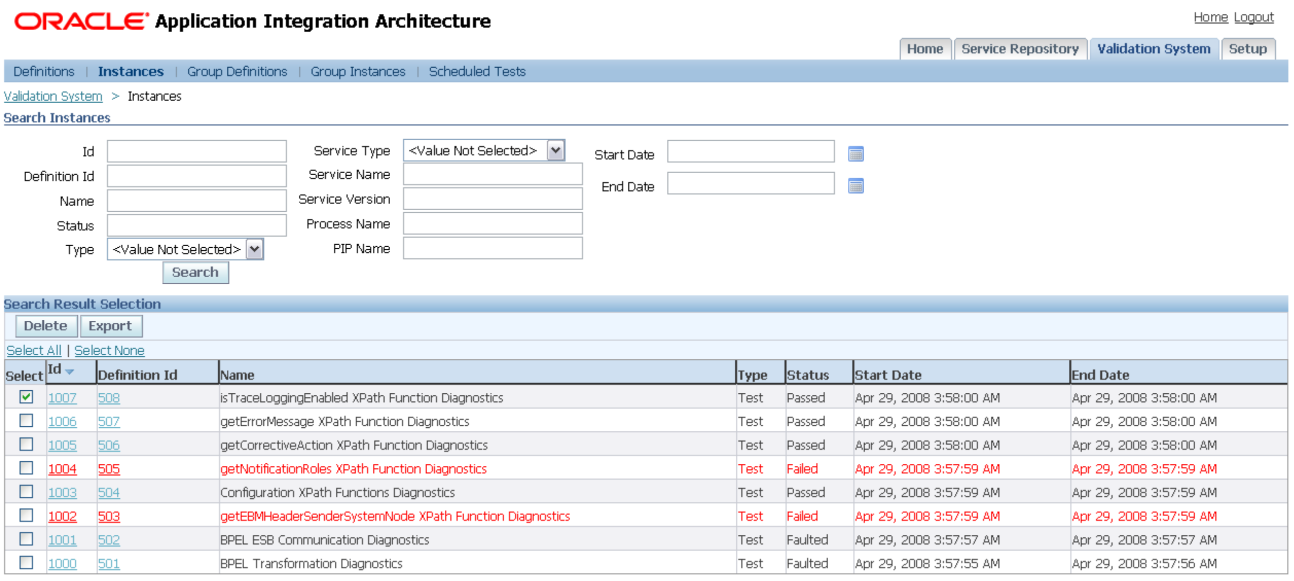
Because test, simulator, and group instance details that may be associated with definitions in the source system are not valid in the target system, they will not be imported.

If the same definition is uploaded multiple times, multiple duplicate definitions will be created in the target system.

For more information about the Definitions page, see Searching for and Working with Test and Simulator Definitions.

Exporting Test and Simulator Instances

Access the Instances page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests
Validation System > Instances
Search Instances

Id	
Definition Id	
Name	
Status	
Type	<Value Not Selected>
Service Type	<Value Not Selected>
Service Name	
Service Version	
Process Name	
PIP Name	
Start Date	
End Date	

Search
Search Result Selection
Delete Export
Select All | Select None

Select	Id	Definition Id	Name	Type	Status	Start Date	End Date
✓	1007	508	isTraceLoggingEnabled XPath Function Diagnostics	Test	Passed	Apr 29, 2008 3:58:00 AM	Apr 29, 2008 3:58:00 AM
☐	1006	507	getErrorMessage XPath Function Diagnostics	Test	Passed	Apr 29, 2008 3:58:00 AM	Apr 29, 2008 3:58:00 AM
☐	1005	506	getCorrectiveAction XPath Function Diagnostics	Test	Passed	Apr 29, 2008 3:58:00 AM	Apr 29, 2008 3:58:00 AM
☐	1004	505	getNotificationRoles XPath Function Diagnostics	Test	Failed	Apr 29, 2008 3:57:59 AM	Apr 29, 2008 3:57:59 AM
☐	1003	504	Configuration XPath Functions Diagnostics	Test	Passed	Apr 29, 2008 3:57:59 AM	Apr 29, 2008 3:57:59 AM
☐	1002	503	getEBMHeaderSenderSystemNode XPath Function Diagnostics	Test	Failed	Apr 29, 2008 3:57:59 AM	Apr 29, 2008 3:57:59 AM
☐	1001	502	BPEL ESB Communication Diagnostics	Test	Faulted	Apr 29, 2008 3:57:57 AM	Apr 29, 2008 3:57:57 AM
☐	1000	501	BPEL Transformation Diagnostics	Test	Faulted	Apr 29, 2008 3:57:55 AM	Apr 29, 2008 3:57:56 AM

Instances page

Export	Use to export instances in XML format. You can use XML-based reporting tools to generate reports of test and simulator executions using these XML files. Select the **Select** check box for one or more instances and click the **Export** button to initiate the export.

Copyright © 2009, Oracle. All rights reserved.

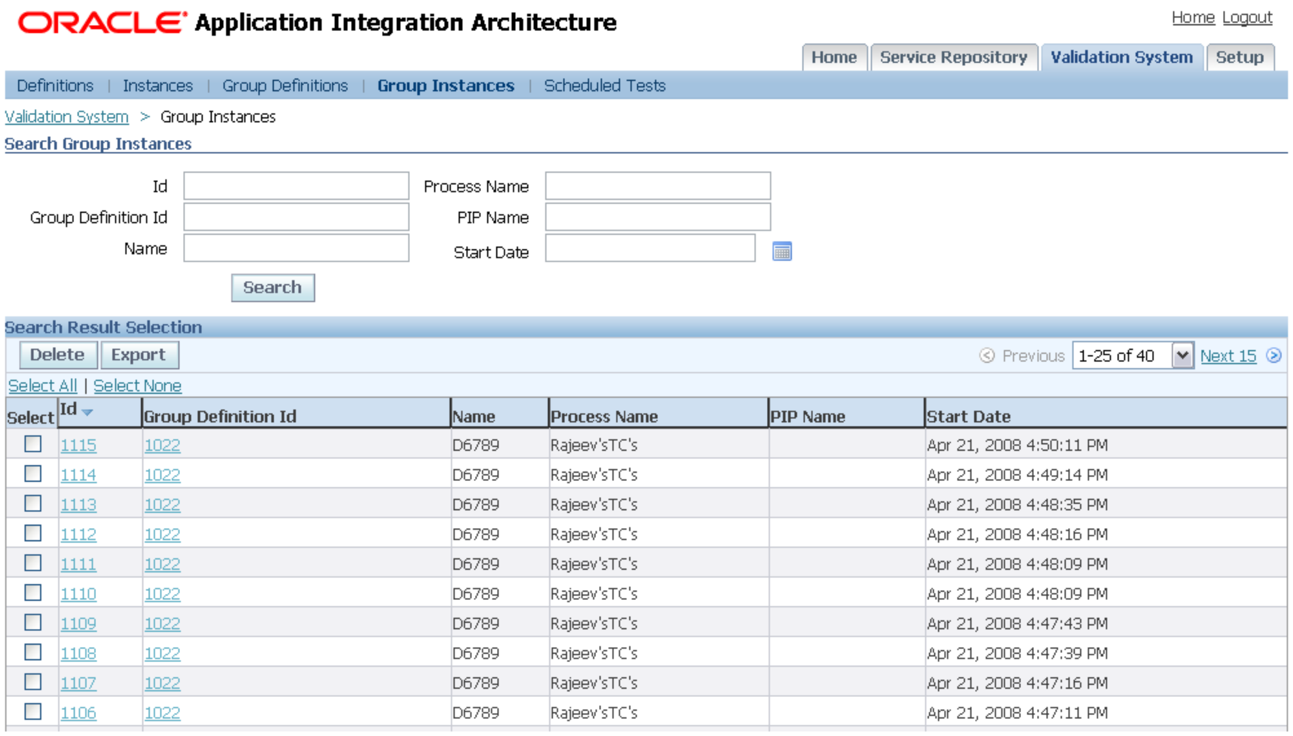
Click Save to create and save the definitions to a location on your local system. The default file name for the exported definition(s) is ***Instances.xml.***

For more information about the structure of the Definitions.xml file created by the CAVS export instance feature, see Appendix A: XML Structures of Exportable CAVS Definitions and Instances.

For more information about the Instances page, see Working with Test and Simulator Instances.

Exporting Group Instances

Access the Group Instances page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Definitions | Instances | Group Definitions | Group Instances | Scheduled Tests
Validation System > Group Instances
Search Group Instances
Id
Group Definition Id
Name
Process Name
PIP Name
Start Date
Search
Search Result Selection
Delete Export
Previous 1-25 of 40 Next 15
Select All | Select None

Select	Id	Group Definition Id	Name	Process Name	PIP Name	Start Date
☐	1115	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:50:11 PM
☐	1114	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:49:14 PM
☐	1113	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:48:35 PM
☐	1112	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:48:16 PM
☐	1111	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:48:09 PM
☐	1110	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:48:09 PM
☐	1109	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:47:43 PM
☐	1108	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:47:39 PM
☐	1107	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:47:16 PM
☐	1106	1022	D6789	Rajeev'sTC's		Apr 21, 2008 4:47:11 PM

Group Instances page

Export

Select the **Select** check box for one or more group instances that you want to export and click the **Export** button to execute the download.

For more information about the structure of the Definitions.xml file created by the CAVS export definition feature, see Appendix A: XML Structures of Exportable CAVS

Definitions and Instances.

For more information about the Group Instances page, see Viewing Group Instances.

Copyright © 2009, Oracle. All rights reserved.

Part 3: Setting Up and Using Error Handling and Logging

Chapter 20: Understanding Error Handling and Logging

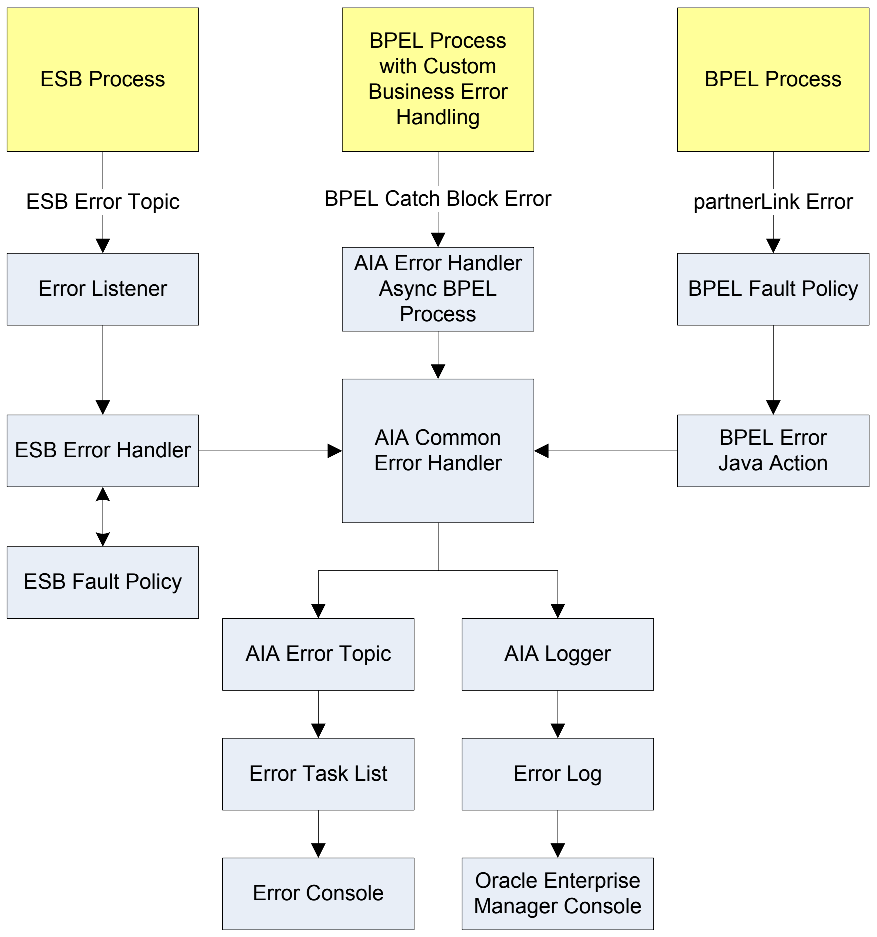
Error handling and logging Core Infrastructure Components support the needs of integration services operating in an Oracle Application Integration Architecture (AIA) ecosystem.

This chapter provides overviews of:

- The Error Handling Framework
- Trace and error logging.

Understanding the Error Handling Framework

This diagram provides a high-level overview of the Error Handling Framework.



Error Handling Framework components

The Error Handling Framework provides the following key features for integration services operating in an Oracle AIA ecosystem:

- Error logging.

  Logs error messages in a consistent schema in a non-intrusive manner.

- Error notifications.

  Issues error notifications to suitable actor roles, such as integration administrators, and FYI roles, such as customer service representatives.

- Error actions.

  Automatically acts upon the errored object. For example, provides an automated retry action.

  Error actions are configured in fault policies. For BPEL, error actions include retry, rethrow, replay, abort, Java action, and human-intervention. For Enterprise Service Bus (ESB), retry is the only error action.

Error handling is invoked in the following scenarios:

- A BPEL partner link experiences an invocation error.
- A BPEL partner link receives a fault.
- A BPEL process throws a non-partner link error in a catch block and invokes the Oracle AIA Async Error Handling process.
- An ESB service experiences an error.

Understanding Error Handling for BPEL Partner Link Errors

There are two BPEL partner link error scenarios:

1. A BPEL partner link invocation fails.
2. A BPEL partner link returns a fault.

Both of these types of errors are captured using the BPEL Fault Policy Framework. The Error Handling Framework provides a custom Java action, oracle.apps.aia.core.eh.BPELJavaAction, which can be configured as the Java action for all of the policies. By configuring fault policies to include this Java action, the Error Handling Framework will perform all necessary error logging and notifications.

For more information, see *Oracle SOA Suite 10.1.3.3.0 New Features Technote:* Fault Management Framework.

Copyright © 2009, Oracle. All rights reserved.

Understanding Error Handling for Non-Partner Link Errors in BPEL

If a BPEL process wants to throw a non-partner link error, such as a business or validation errors, the process needs to throw the error explicitly, catch it in a catch block, and invoke the Oracle AIA Async Error Handing process. The input to the Async Error Handling process is a fault message in the Oracle AIA fault message schema.

Understanding Error Handling for ESB Service Errors

If an error occurs in an ESB service, the ESB posts an error object on the ESB_ERROR topic of the ESB. The Error Handling Framework intercepts the error whenever an error object is posted on this topic and gets the details of the error from ESB. Once the framework obtains the error details, it does the necessary error logging and notifications.

ESB errors can be one of two types:

1. Retryable.
2. Non-retryable.

For more information, see *Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide,* "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Configuring ESB Processes to Adhere to Error Handling Requirements.

- If the retry action results in success, a warning message is logged to the trace logger.
- If the retry action results in failure, the Error Handling framework does the necessary error logging and notifications.

Understanding Trace and Error Logging

The Oracle AIA enables you to generate trace and error log files that provide a detailed view of services running in your Oracle AIA ecosystem. These logs can be especially informative when troubleshooting service processing issues:

- Trace

  Trace logs capture chronological recordings of a service's general activities. The trace log is created by configuring the service to make an explicit call using the trace logging custom XPath or Java API.

For more information, see *Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide,* "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Configuring Oracle AIA Processes for Trace Logging.

- Error

  Error logs capture a recording of errors that occur during a service's activities. No specific configurations are required to make BPEL and ESB services eligible for error logging. The Error Handling Framework is designed to trigger an error logging event for errors occurring in any of the Oracle AIA services, whether they are BPEL- or ESB-based. The Error Handling Framework does this logging non-intrusively.

These log files can be managed and viewed using the Oracle Enterprise Manager (EM) user interface, in much the same way that standard log files generated by various components of the Oracle SOA Suite can be handled in EM. Using EM as the user interface for the logs enables searching, sorting, and filtering of logs.

For more information, see Viewing Trace and Error Logs in EM.

Copyright © 2009, Oracle. All rights reserved.

Chapter 21: Enabling and Viewing Trace and Error Logs

This chapter discusses how to:

- Enable trace logging.
- View trace and error logs in Oracle Enterprise Manager (EM).

Enabling Trace Logging

Trace logging is enabled using configurations in the AIAConfigurationProperties.xml file located in <aia.home>/config/.

To enable trace logging for the entire system, access the AIAConfigurationProperties.xml file and set the TRACE.LOG.ENABLED property at the system level to *TRUE*.

To enable trace logging for an individual service, access the AIAConfigurationProperties.xml file and set the TRACE.LOG.ENABLED property for the service to *TRUE*.

The property set at the service level overrides the property set at the system level.

Note: Whenever the AIAConfigurationProperties.xml file is updated, the file must be reloaded for updates to be reflected in the applications or services that use the updated properties. You can perform this reload by clicking the Reload button on the Configuration page in the Oracle Application Integration Architecture (AIA) Console. Alternatively, you can perform the reload by rebooting the server.

For more information, see Chapter 7: Loading Oracle AIA Configuration File Updates.

Note: No specific configurations are required to make Oracle AIA services eligible for error logging. The Error Handling Framework is designed to trigger an error logging event for errors occurring in any of the Oracle AIA services, whether they are BPEL- or ESB-based. The Error Handling Framework does this logging non-intrusively.

Viewing Trace and Error Logs in EM

In this section, we discuss how to view trace and error logs in EM.

Pages Used to View Trace and Error Logs in EM

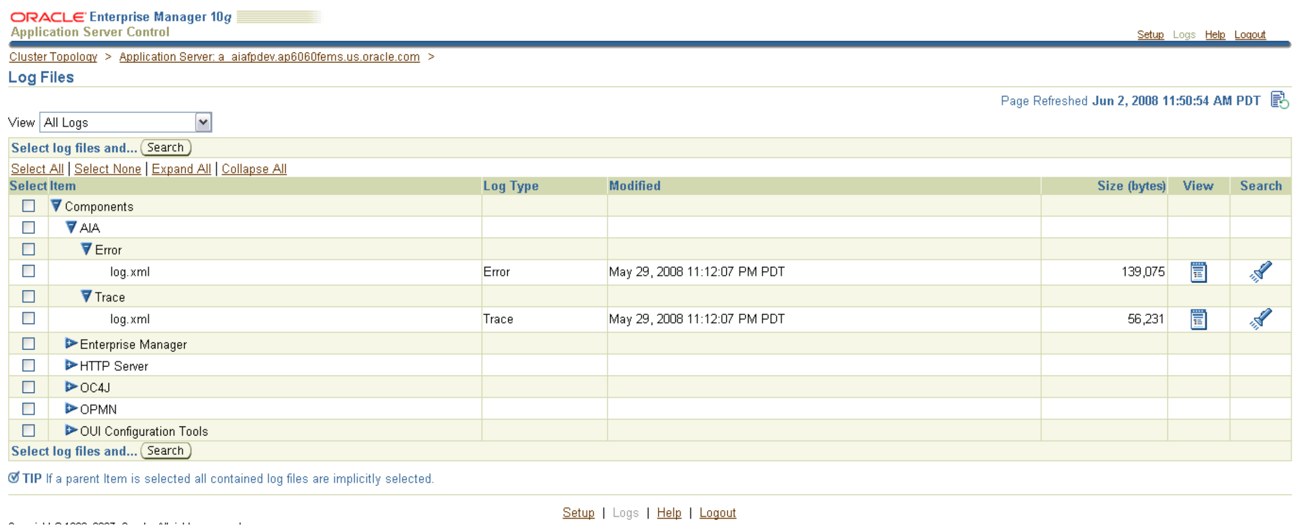
Page Name	Navigation	Usage
Log Files	1. Access the Oracle Enterprise Manager -Cluster Topology page. 2. In the Name column, click the oc4j_<*instance name*> application server link. 3. Click the Logs link located in the page header or footer. 4. On the Log Files page, in Components, AIA, expand Error to access error logs. Expand Trace to access trace logs.	Access pages that allow you to search for log messages and view log files.
Search Logs	1. Access the Oracle Enterprise Manager -Cluster Topology page. 2. In the Name column, click the oc4j_<instance name> application server link. 3. Click the Logs link located in the page header or footer. 4. On the Log Files page, in Components, AIA, expand Error to be able to search for error log messages. Expand Trace to be able to search for trace log messages. 5. Click the Search contents of this log file button for the log in which you want to search for specific log messages.	Search for specific log messages in a selected log. Use the page elements to search for specific log messages in the selected log. Search results are displayed at the bottom of the page.
View Log: <logname>	1. Access the Oracle Enterprise Manager -Cluster Topology page. 2. In the Name column, click the oc4j_<instance name> application server link. 3. Click the Logs link located in the page header or footer. 4. On the Log Files page, in	View log messages and other contents of a selected log.

Copyright © 2009, Oracle. All rights reserved.

Page Name	Navigation	Usage
	Components, AIA, expand Error to be able to view the error log. Expand Trace to be able to view the trace log. 5. Click the View this log file button for the selected log.	

Viewing a Trace or Error Log in EM

Access the Log Files page.



ORACLE Enterprise Manager 10g
Application Server Control

Setup Logs Help Logout

Cluster Topology > Application Server: a_aiafpdev.ap6060fems.us.oracle.com >

Log Files

Page Refreshed Jun 2, 2008 11:50:54 AM PDT

View All Logs

Select log files and... Search

Select All | Select None | Expand All | Collapse All

Select	Item	Log Type	Modified	Size (bytes)	View	Search
☐	▼ Components					
☐	▼ AIA					
☐	▼ Error					
☐	log.xml	Error	May 29, 2008 11:12:07 PM PDT	139,075		
☐	▼ Trace					
☐	log.xml	Trace	May 29, 2008 11:12:07 PM PDT	56,231		
☐	▶ Enterprise Manager					
☐	▶ HTTP Server					
☐	▶ OC4J					
☐	▶ OPMN					
☐	▶ OUI Configuration Tools					

Select log files and... Search

TIP If a parent Item is selected all contained log files are implicitly selected.

Setup | Logs | Help | Logout

Log Files page

To work with Oracle AIA trace logs, expand the AIA and Trace elements.

To work with Oracle AIA error logs, expand the AIA and Error elements.

Chapter 22: Using the Error Console

This chapter provides an overview of the Error Console and discusses how to use the Error Console.

Understanding the Error Console

The Oracle Worklist application is used to provide an Error Console for the Oracle Application Integration Architecture (AIA). The Error Console application is a user interface (UI) that Actor roles, such as integration administrators, and FYI roles, such as customer service representatives (CSR), can use to access details about Oracle AIA ecosystem service errors that have been assigned to them for resolution or for informational purposes only. Actor and FYI roles are notified upon the creation of an error task that has been assigned to them.

Based on their roles, users will be able to interact with the following types of tasks in the Error Console:

- Single-approver task

  Actor roles, such as integration administrators, are assigned single-approver tasks in the Error Console. Typically, this role is responsible for taking action to resolve the error and must update the error task with activity and status details. Therefore, for Actor roles, the Error Console provides an editable UI.

- FYI task

  FYI roles, such as customer service representatives, are assigned FYI tasks in the Error Console. Typically, this role only needs a view of information about the status of the errored end-to-end transaction. Therefore, for FYI roles, the Error Console provides a display-only UI. The FYI role is not responsible for taking any particular action to resolve the error.

The Error Console provides the following error details that can assist in the troubleshooting process:

- EBMID
- EBMName
- EBOName
- Verb Code
- Business Scope Reference ID
- Business Scope Reference Instance ID
- Enterprise Service Name
- Enterprise Service Operation Name
- Sender Reference ID

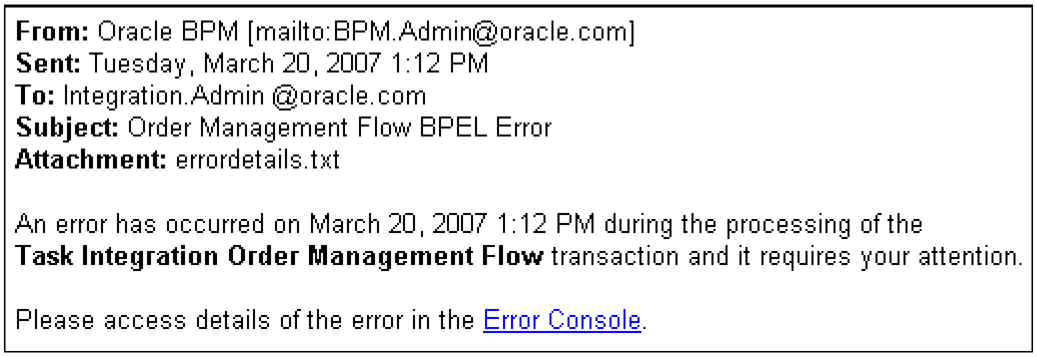
- Sender Message ID
- Sender Reference Transaction Code
- Sender Object Identification ID
- Context ID
- EBOID
- Reporting Date Time
- Corrective Action
- Fault Message Code
- Fault Message Text
- Severity
- Stack
- Faulting Service ID
- Faulting Service Implementation Code
- Faulting Service Instance ID

Errors that need to appear in the Error Console are aggregated by the AIAReadJMSNotification BPEL process. The AIAReadJMSNotification BPEL process listens to the Oracle AIA Error Topic Java message service (JMS) topic, which is populated by the Error Handling Framework.

When the AIAReadJMSNotification process receives an error using the Oracle AIA Error Topic, it picks up the fault message and calls the AIAErrorTaskAdministration process for notifying actor and FYI roles. The AIAReadJMSNotification process consults the Business Service Registry's (BSR) Error Notifications table to derive the user role that should be assigned to the error and receive error notification.

For more information, see Setting Up Error Notifications for Oracle AIA Processes.

This is the text of a sample error notification email.


From: Oracle BPM [mailto:BPM.Admin@oracle.com]
Sent: Tuesday, March 20, 2007 1:12 PM
To: Integration.Admin @oracle.com
Subject: Order Management Flow BPEL Error
Attachment: errordetails.txt

An error has occurred on March 20, 2007 1:12 PM during the processing of the **Task Integration Order Management Flow** transaction and it requires your attention.

Please access details of the error in the Error Console.

Sample error notification email

Copyright © 2009, Oracle. All rights reserved.

Using the Error Console

Once you have been notified of an Oracle AIA error task that you need to act upon to resolve, you can use the details provided by the Error Console to troubleshoot the error.

To use the Error Console, log into the Oracle Worklist application that is providing the Error Console. Your assigned tasks display on the My Tasks page. You can filter your assigned tasks using various criteria and search for assigned tasks by title, priority, and status.

Actor user roles can work on a task by acquiring the task. They can also enter comments against the task and update the task status. For example, when the error has been resolved, the user can set the task action to *COMPLETED.* Setting this value in the Actions field completes the task.

For more information about message resubmission, see Chapter 23: Using the Message Resubmission Utility.

Click an assigned task to access the Task Details page, where you can see complete task details, including task history and actions.

For more information, see *Oracle BPEL Process Manager Developer's Guide*, "Worklist Application."

Chapter 23: Using the Message Resubmission Utility

This chapter provides an overview of message resubmission and discusses how to use the Message Resubmission Utility.

Overview

To use the Message Resubmission Utility, you must implement error handling and recovery for the asynchronous message exchange pattern.

For more information, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Implementing Error Handling and Recovery for the Asynchronous Message Exchange Pattern to Ensure Guaranteed Message Delivery."

According to this implementation method, when a message cannot be delivered to a service or component in the flow of a global transaction, the message is rolled back to the appropriate source milestone. This source milestone corresponds to an Oracle Advanced Queue, JMS topic, or a Resequencer store. It is here that the message will be persisted until it can be resubmitted for delivery to the service or component.

At the same time, a BPEL PartnerLink fault is raised by the Error Handling framework and notifications regarding the fault are issued to administrators using the Error Console.

For more information about the Error Console, see Understanding the Error Console.

Once notified, the most natural course of action is for the administrator to bring up the failed service or component. Once the service or component is back up and running, the administrator can use the Message Resubmission Utility to recover the faulted message from the source milestone. The Message Resubmission Utility changes the state of the faulted message to the ***Ready*** state, enabling it to be picked up by the consumer process.

Using the Message Resubmission Utility

This section discusses how to use the Message Resubmission Utility to resubmit a faulted message.

Note. For message resubmission scenarios that involve Oracle Advanced Queue, we provide the MSG_RESUBMIT stored procedure. This procedure assumes that the message type is ***SYS.AQ$_JMS_MESSAGE***. If the message type being used is not ***SYS.AQ$_JMS_MESSAGE,*** you must change the data type for the *MSG* variable in the MSG_RESUBMIT stored procedure and then recompile the procedure. You can then use the Message Resubmission Utility for resubmission based on message ID.

To use the Message Resubmission Utility:

1. Access the Oracle AIA log file in Oracle Enterprise Manager (EM) to look up the following values for the message that requires resubmission:
   - SenderResourceTypeCode
   - SenderResourceID
   - SenderMessageID

   For more information about these values in the contexts of the Oracle AIA fault message schema, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Describing the Oracle AIA Fault Message Schema.
   For more information about viewing the Oracle AIA log in Oracle EM, see Viewing Trace and Error Logs in EM.

2. Set these values in the ResubmitBuild.properties file located in $AIA_HOME/Infrastructure/ErrorHandling/util/.
3. For Windows, execute $AIA_HOME\bin\aiaenv.bat. For Linux, source $AIA_HOME/bin/aiaenv.sh.
4. Navigate to $AIA_HOME/Infrastructure/ErrorHandling/util and execute the following ant command:

   ```
   ant -buildfile MessageResubmit.xml
   ```

   The MessageResubmit.xml script references the edited ResubmitBuild.properties file.

   Once run, the script resets the message status back to a ready state so that the transaction can resume its flow.

Copyright © 2009, Oracle. All rights reserved.

Chapter 24: Setting Up Error Notifications and Trace Logging

This chapter discusses how to:

- Set up trace log levels.
- Set up error notifications for Oracle Application Integration Architecture (AIA) services.
- Set up Error Console task roles and users.

Setting Up Trace Log Levels

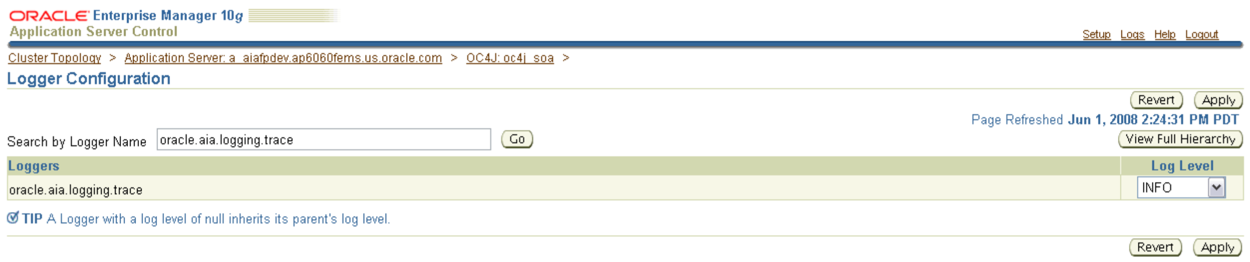
This section discusses how to configure trace log levels.

Page Used to Set Up Trace Log Levels

Page Name	Navigation	Usage
Logger Configuration	1. Access the Oracle Enterprise Manager -Cluster Topology page. In the Name column, click the oc4j_<instance name> application server link. 2. Click the Administration link. 3. Under Administration Tasks, Properties in the Go to Task column, click the Go to Task button for Logger Configuration. 4. Search for the trace log name, such as oracle.aia.logging.trace.	Set up log levels.

Setting Up Trace Log Levels

Access the Logger Configuration page.



Cluster Topology > Application Server: a_aiafpdev.ap6060fems.us.oracle.com > OC4J: oc4j_soa >

Logger Configuration

Page Refreshed Jun 1, 2008 2:24:31 PM PDT

Search by Logger Name oracle.aia.logging.trace

Loggers	Log Level
oracle.aia.logging.trace	INFO

TIP A Logger with a log level of null inherits its parent's log level.

Logger Configuration page

Use the **Search by Logger Name** field to easily locate and access log level settings for the *oracle.aia.logging.trace* log.

Use the page elements to set trace log levels.

Loggers	Displays the name of the logger.
Log Level	Select the level of logging you would like the trace logger to perform. Use the following log level descriptions to guide your selection: ***SEVERE:*** A problem requiring attention from the system administrator has occurred. ***WARNING:*** An action occurred or a condition was discovered that should be reviewed and may require action before an error occurs. ***INFO:*** A report of a normal action or event. This could be a user operation, such as "login completed" or an automatic operation such as a log file rotation. ***CONFIG:*** A configuration-related message or problem. ***FINE:*** A trace or debug message used for debugging or performance monitoring. Typically contains detailed event data. ***FINER:*** A fairly detailed trace or debug message. FINEST: A highly detailed trace or debug message.

Setting Up Error Notifications for Oracle AIA Processes

This section discusses how to define error notification details for BPEL and Enterprise Service Bus (ESB) processes operating in an Oracle AIA ecosystem.

Page Used to Set Up Error Notifications

Page Name	Navigation	Usage
Error Notification	1. Access the Oracle AIA Console.	Define and modify error notification

Copyright © 2009, Oracle. All rights reserved.

Page Name	Navigation	Usage
	2. Select the Setup tab. 3. Click the Error Notification link.	details for BPEL and ESB processes operating in an Oracle AIA ecosystem.

Configuring Error Notification Settings in ns_emails.xml

The settings you define in ns_emails.xml are used to send email error notifications to roles specified in the BSR_ERROR_NOTIFICATIONS table or to the default roles fetched from AIAConfigurationProperties.xml, as described in Setting Up Error Notification Details for Oracle AIA Processes.

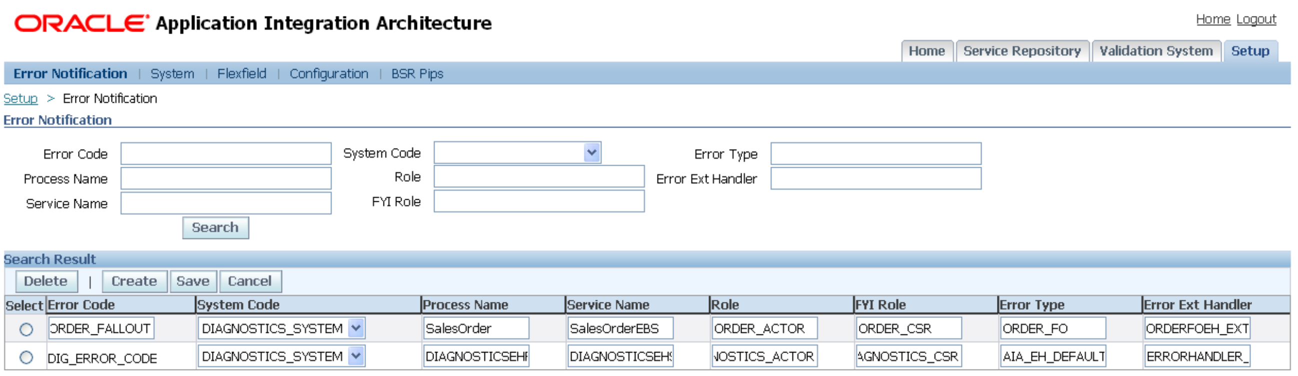
ns_emails.xml is located in <SOA_HOME>/bpel/system/services/config.

Configure the following settings in ns_emails.xml. Ensure that you restart the BPEL Process Manager after making these changes.

- In the <EmailAccounts> element, set the <NotificationMode> attribute to ***EMAIL***, instead of ***NONE***.
- In the <GeneralSettings> element, set the <FromAddress> attribute to a valid email address. For example, ***firstname.lastname@acme.com***.
- In the <OutgoingServerSettings> element, set the <STMPHost> attribute. For example, ***mail.server.com***. **Note** that <STMPPort> is already set to ***25***.
- Comment out <IncomingServerSettings> element.

Setting Up Error Notification Details for Oracle AIA Processes

Access the Error Notifications page.


ORACLE Application Integration Architecture
Home Logout
Home | Service Repository | Validation System | Setup
Error Notification | System | Flexfield | Configuration | BSR Pips
Setup > Error Notification
Error Notification
Search
Search Result
Delete | Create | Save | Cancel

Error Code		System Code		Error Type	
Process Name		Role		Error Ext Handler	
Service Name		FYI Role			

Select	Error Code	System Code	Process Name	Service Name	Role	FYI Role	Error Type	Error Ext Handler
○	ORDER_FALLOUT	DIAGNOSTICS_SYSTEM	SalesOrder	SalesOrderEBS	ORDER_ACTOR	ORDER_CSR	ORDER_FO	ORDERFOEH_EXT
○	DIG_ERROR_CODE	DIAGNOSTICS_SYSTEM	DIAGNOSTICSEHI	DIAGNOSTICSEH!	IOSTICS_ACTOR	AGNOSTICS_CSR	AIA_EH_DEFAULT	ERRORHANDLER_

Error Notifications page

The error notifications you define on the Error Notifications page are stored in the BSR ERROR NOTIFICATIONS table.

For more information, see Chapter 1: Understanding the BSR.

Note: For a given process, if no entry is found in the BSR ERROR NOTIFICATIONS table, the default roles specified in AIAConfigurationProperties.xml are used. Therefore, you are not required to populate the BSR_ERROR_NOTIFICATIONS table unless there is an explicit need.

For more information, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Configuring Oracle AIA Processes for Error Handling and Trace Logging," Performing Common Configurations for Both BPEL and ESB Processes to Adhere to Error Handling Requirements.

Error Code	Enter the error code associated with the error notification you are searching for.
Process Name	Enter the process name associated with the error notification you are searching for.
Service Name	Enter the service name associated with the error notification you are searching for.
System Code	Select the service code associated with the error notification you are searching for.
Role	Enter the actor role associated with the error notification you are searching for.
FYI Role	Enter the FYI role associated with the error notification you are searching for.
Error Type	Enter the error type associated with the error notification you are searching for.
Error Ext Handler (error extension handler)	Enter the error extension handler associated with the error notification you are searching for.
Search	Click to execute a search for error notifications based on your search criteria.

Search Result

The **Search Result** grid displays error notifications based on your search criteria.

Delete	Select a **Select** option for the error notification row you want to delete and click the **Delete** button to execute the deletion.
Create	Click to add a row to the **Search Result** grid, where you can enter details for a new error notification.
Save	Click to save all entries and updates to the page.
Cancel	Click to undo all updates made to the page after the last save.
Error Code	For a BPEL process error notification, this is the fault code. For an ESB process error notification, this is the *ESBerror* string. ESB does not have a concept of error codes.
System Code	This is the system code of the participating application.

Copyright © 2009, Oracle. All rights reserved.

Process Name	This is the business process in which the service is participating.
Service Name	For BPEL services, this is the name of the service that experiences the error for which you are defining error notification details. For example, ***SampleBPELProcess.*** For ESB services, this is the fully qualified name of the service that is erroring out. The format of a fully qualified name is *System.ServiceGroup(*).Service.* You can have *n* number of service groups between *System* and *Service*: *System.ServiceGroup1.ServiceGroup2.Service.* For example, ***default.FP_EH_Test_BPEL.FP_EH_Test_BPEL_1_0.***
Role	Specify an actor role that you want to receive notification regarding this error. This is the role that will be responsible for taking action to correct the error that generated the notification. The Oracle AIA Error Handling Framework is delivered to issue error notifications through the Oracle Worklist application. By default, the Oracle Worklist is configured to use a JAZN (Java authorization) user store, but can be configured to use other user stores, such as LDAP. Ensure that the actor role you specify here has a corresponding entry in your Oracle Worklist application's user store.
FYI Role	Specify an FYI role that you want to receive notification regarding this error. This is the role that will be notified of the error, but will not be responsible for taking any actions to correct the error that generated the notification. The Oracle AIA Error Handling Framework is delivered to issue error notifications through the Oracle Worklist application. By default, the Oracle Worklist is configured to use a JAZN (Java authorization) user store, but can be configured to use other user stores, such as LDAP. Ensure that the FYI role you specify here has a corresponding entry in your Oracle Worklist application's user store.
Error Type	The default value is AIA_EH_DEFAULT. Use this value if you want to use the Oracle AIA default error listener as the consuming component for this error notification. Enter a unique value here if you are using extended error handling functionality. **For more information** about extending error handling, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Extending Error Handling." If you want to use default and extended error handling functionality in a single error notification definition, add multiple Error Type values separated by commas. For example, ***AIA_EH_DEFAULT, ORDER_FO***, where ***AIA_EH_DEFAULT*** is the default Oracle AIA follow-through action, and ***ORDER_FO*** identifies the custom JMSCorrelationID for the extended error handling implementation. The listeners and associated actions for both

	of these error types will be executed at runtime.
Error Ext Handler (error extension handler)	The default value is ***ERRORHANDLER_EXT***. Use this value for the error notification if you are not using an extended handler and the fault message will be generated based on the default fault message schema. **For more information** about extending fault messages, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Extending Fault Messages." If you are using an extended handler to extend the fault message for this error notification, enter a unique value to identify the extension handler that will be used to enrich the fault message.

Logic Used to Determine Notification Roles for an Error

The Error Handling Framework uses runtime values and the data you enter on this page to execute the following hierarchical logic to determine the appropriate notification roles for an error.

1. If all four runtime values (SYSTEM CODE, ERROR CODE, SERVICE NAME, and PROCESS_NAME) are available and they map to an error notification entry in this table, use the specified notification roles.
2. If the ERROR CODE, SERVICE NAME, and PROCESS NAME are available and map to an error notification entry in this table, use the specified notification roles.
3. If the SERVICE_NAME and PROCESS_NAME are available and map to an error notification entry in this table, use the specified notification roles.
4. If the SERVICE_NAME is available and maps to an error notification entry in this table, use the specified notification roles.
5. If none of these values are available, the default values are fetched from the AIAConfigurationProperties.xml file.

Setting Up Error Console Task Roles and Users

There are two types of Oracle AIA Error Console task roles:

- Actor role.

  Actor roles receive notifications for and are assigned to error scenarios occurring in Oracle AIA integration flows. An example of an Actor role is an administrator. The task is editable in the Error Console and is meant to be worked on and resolved by the actor assigned to the task.

- FYIrole

  This role receives for-your-information (FYI) notifications for error scenarios occurring in Oracle AIA integration flows. An example of an FYI role is a customer service representative. The task is displayed in read-only view in the Error Console.

Actor and FYI roles are defined for each Oracle AIA process on the Error Notifications page to populate the BSR ERROR NOTIFICATIONS table. These notification roles help derive the users who should be notified by the Error Console when an error occurs in a process.

Copyright © 2009, Oracle. All rights reserved.

When an Error Console task is associated with a role based on the notification details fetched from the BSR_ERROR_NOTIFICATIONS table, it uses the Oracle Worklist default XML-based Java authorization (JAZN) identity service to match the notification roles to actual users and groups.

These associations between roles, users, and groups are configured in <SOA_Oracle_Home>\j2ee\<OC4J Instance name>\config\system-jazn-data.xml. Ensure that roles that you define in the BSR_ERROR_NOTIFICATIONS table are present in system-jazn-data.xml.

Note: While using the JAZN user store is the default behavior, you can use other user stores, such as LDAP.

Once the Error Console task has been assigned to users and groups, the notification is issued based on email addresses configured in <SOA_Oracle_Home>\bpel \system\services\config\user-properties.xml.

For more information, see *Oracle BPEL Process Manager Administrator's Guide*, "Service Configuration." **For more information**, see *Oracle BPEL Process Manager Administrator's Guide*, "Creating a Custom Identity Service Plug-in." **For more information**, see Setting Up Error Notifications for Oracle AIA Processes.

Part 4: Working with the Diagnostics Framework

Copyright © 2009, Oracle. All rights reserved.

Chapter 25: Understanding the Diagnostics Framework

This chapter provides an overview of the Diagnostics Framework.

The Diagnostics Framework

The Diagnostics Framework provides a set of diagnostic tests designed to run in an Oracle Application Integration Architecture (AIA) ecosystem. These diagnostic tests provide data that can be useful to integration developers and administrators who are maintaining an Oracle AIA ecosystem. These tests can be run independently or as a group.

The Diagnostics Framework is composed of the following key components that work together to run these diagnostic tests:

- Oracle AIA Diagnostics test scripts and the classes that implement the AIA Diagnostics interface.
- Oracle AIA Diagnostics driver.
- Oracle AIA Diagnostics configuration file.
- Oracle AIA Diagnostics interface.

Oracle AIA Diagnostic Test Scripts

Each diagnostic test has its own script. The test script invokes the Oracle AIA Diagnostics driver class, oracle.apps.aia.core.diagnostics.AIADiagnostics, with the test name as input.

Another test script, batch-diagnostics, invokes the driver class,

oracle.apps.aia.core.diagnostics.AIADiagnostics, with a keyword of *batch*. This driver class reads the AIA Diagnostics configuration file, AIADiagnosticsConfig.xml, located in *<aia.home>/diagnostics/config/*, and runs all tests that are listed in the batch section.

Oracle AIA Diagnostics Driver

The Oracle AIA Diagnostics driver performs the following tasks:

1. When invoked by a test script with the test name as input, the Oracle AIA Diagnostics driver reads the Oracle AIA Diagnostics configuration file, AIADiagnosticsConfig.xml, located in <aia.home>/diagnostics/config/ to find the section for the input test name.
2. The Oracle AIA Diagnostics driver class invokes the implementing class using parameters associated with the given test as configured in the Oracle AIA Diagnostics configuration file.

Oracle AIA Diagnostics Configuration File

The Oracle AIA Diagnostics Framework uses a single configuration file, AIADiagnosticsConfig.xml, which contains metadata, such as the implementing class to be invoked, and runtime data, such as parameters for each diagnostic test. This configuration file is located in *<aia.home>/diagnostics/config/.* One test section exists per diagnostic test. The batch section contains the name of each test that will be run in batch mode.

Following is sample text from the Oracle AIA Diagnostics configuration file, which you will work in to define individual and batch test parameters:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <AIADiagnostics xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:schemaLocation="http://xmlns.oracle.com/aia/core/diagnostics/V1
   AIADiagnosticsConfig.xsd"xmlns="http://xmlns.oracle.com/aia/core/
   diagnostics/V1">
     <Test name="SOASuiteVersion">

<class>oracle.apps.aia.core.diagnostics.tests.SOASuiteVersion</class
>
     </Test>
     <Test name="CustomDirectories">
       <class>oracle.apps.aia.core.diagnostics.tests.
        CustomDirectoriesDiagnostic</class>
     </Test>
     <Test name="AIAConfigIntegrity" >

<class>oracle.apps.aia.core.diagnostics.tests.AIAConfigIntegrity
        Diagnostic</class>
     </Test>
     <Test name="CustomizedBpel">
       <class>oracle.apps.aia.core.diagnostics.tests.
        CustomizedBpelDiagnostic</class>
       <param name="Check_Service">CHECK_ALL</param>
     </Test>
     <Test name="CustomizedEsb">
       <class>oracle.apps.aia.core.diagnostics.tests.
        CustomizedEsbDiagnostic</class>
        <param name="Check_Service">CHECK_ALL</param>
     </Test>
     <Test name="CustomizedSchema">
       <class>oracle.apps.aia.core.diagnostics.tests.
        CustomizedSchemaDiagnostic</class>
       <param name="Check_Schema">CHECK_ALL</param>
     </Test>
     <Test name="ExtendedSchema">
       <class>oracle.apps.aia.core.diagnostics.tests.
        ExtendedSchemaDiagnostic</class>
       <param name="Check_Schema">CHECK_ALL</param>
     </Test>
     <Test name="SeedData">
       <class>oracle.apps.aia.core.diagnostics.tests.
        SeedDataDiagnostic</class>
     </Test>
     <Test name="XdkTest">
```

Copyright © 2009, Oracle. All rights reserved.

```
<class>oracle.apps.aia.core.diagnostics.tests.XdkDiagnostic</class>
       <param name="Test_Definition_ID">%XDK_ID%</param>
     </Test>
     <Test name="BpelEsbCommunication">

<class>oracle.apps.aia.core.diagnostics.tests.BpelEsbCommunication
        Diagnostic</class>
       <param name="Test_Definition_ID">%ESB_BPEL_ID%</param>
     </Test>
     <Test name="CustomXPath">

<class>oracle.apps.aia.core.diagnostics.tests.CustomXPathDiagnostic<
        /class>
       <param name="Definition_IDs">%CONFIG%,%EBM%</param>
     </Test>
     <Test name="DeployedServiceVersion">

<class>oracle.apps.aia.core.diagnostics.tests.DeployedServiceVersion
        </class>
     </Test>
     <Test name="TableDefinitions">
       <class>oracle.apps.aia.core.diagnostics.tests.
        TableDefinitionsDiagnostic</class>
       <param name="ListOfTables_File_Location">$aia.home$/
        diagnostics/config/TableDefinitionsList.xml</param>
     </Test>
     <Batch>
       <TestReference name="SOASuiteVersion" />
       <TestReference name="AIAConfigIntegrity"/>
     </Batch>
</AIADiagnostics>
```

Oracle AIA Diagnostics Interface

Each diagnostic test is implemented by a Java class that needs to implement this interface, IAIADiagnostics. This interface has two methods, *run* and *runBatch*.

Chapter 26: Describing the Diagnostic Tests

This chapter describes the following diagnostic tests:

- SoaSuiteVersion
- CustomDirectories
- AIAConfigIntegrity
- CustomizedBpel
- CustomizedEsb
- CustomizedSchema
- ExtendedSchema
- SeedData
- XdkTest
- BpelEsbCommunication
- CustomXPath
- DeployedServicesVersion
- TableDefinitions
- JMSQueue

The SoaSuiteVersion Diagnostic Test

The results of this diagnostic test provide operating system details, application server details, Oracle BPEL database schema details, and a list of applied patches.

Script Names

soasuiteversion.bat for Microsoft Windows.

soasuiteversion.sh for UNIX.

Test Runtime Parameters

No runtime parameters exist for this diagnostic test.

Test Results

Test results in the result_SoaSuiteVersion.log file include the following information:

- Test name

- Test start time
- Operating system name
- Operating system version
- Operating system architecture
- Application server name
- Application server version
- Application server build date
- Database version and type
- Schema version
- List of applied patches
- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_SoaSuiteVersion.log. This log stores the test instance when the test execution status is either *Partially successful or Completely failed*. The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The CustomDirectories Diagnostic Test

The results of this diagnostic test provide a list of directories that have been added or deleted from the <*aia.home*> directory post-installation.

Script Names

customdirectories.bat for Microsoft Windows.

customdirectories.sh for UNIX.

Test Runtime Parameters

No runtime parameters exist for this diagnostic test.

Test Results

Test results in the result_CustomDirectories.log file include the following information:

- Test name
- Test start time
- The following directories have been added: <*directory path*>

Copyright © 2009, Oracle. All rights reserved.

- The following directories have been deleted: *<directory path>*
- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_CustomDirectories.log. This log stores the test instance when the test execution status is either *Partially successful or Completely failed*. The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The AIAConfigIntegrity Diagnostic Test

The results of this diagnostic test indicate whether any configuration files located in the *<aia.home>/config/* directory have changed.

Script Names

aiaconfigintegrity.bat for Microsoft Windows.

aiaconfigintegrity.sh for UNIX.

Test Runtime Parameters

No runtime parameters exist for this diagnostic test.

Test Results

Test results in the result_AiaConfigIntegrity.log file include the following information:

- Test name
- Test start time
- The following files have been changed: *<file names>.*
- The details of the individual file changes can be viewed in: *AIA_HOME\util\Diagnostics\ChangedFiles*\<testname_timestamp>\
- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_AiaConfigIntegrity.log. This log stores the test instance when the test execution status is either *Partially successful or Completely failed*. The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The CustomizedBpel Diagnostic Test

The results of this diagnostic test indicate whether any BPEL process has been customized. Only BPEL processes that have been delivered as a part of the Oracle AIA installation will be checked for customizations. Any BPEL process that has been added post-installation will not be checked for customizations.

Script Names

customizedbpel.bat for Microsoft Windows.

customizedbpel.sh for UNIX.

Test Runtime Parameters

Test runtime parameters for this test are set in the AIADiagnosticsConfig.xml file located here:

<aia.home>/diagnostics/config/.

```
<param name="Check_Service">CHECK_ALL</param>
```

The Check_Service parameter accepts one of two input values:

- CHECK ALL

  When this value is used, the test scans all BPEL processes to check whether any customizations have been made to the processes post-installation. This is the default input value.

- <BPEL Process Name(s)>

  When this value is used, the test scans the specified BPEL processes to check whether any customizations have been made to the specified BPEL processes post-installation. You can specify multiple BPEL process names, separated by commas.

Note: Any input value, other than *CHECK ALL,* that is provided for the Check_Service parameter in the configuration file is treated as a BPEL process name.

Test Results

Test results in the result_CustomizedBpel.log file include the following information:

- Test name
- Test start time
- Diagnostic test result

  Diagnostics test result values include:

  - No BPEL process has been customized.
  - <*BPEL Process Name*> might have been undeployed.
  - <*BPEL Process Name*> has been customized.

Copyright © 2009, Oracle. All rights reserved.

    - Following file(s) have been changed for this process: *<List Of Files>*.
    - The details of the individual files can be viewed in*:* *AIA_HOME\util\Diagnostics\ChangedFiles\<testname_timestamp>*

- Test end time
- Total time taken

  If any errors occurred during test execution, error details can be found in error_CustomizedBpel.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The CustomizedEsb Diagnostic Test

The results of this diagnostic test indicate whether any Enterprise Service Bus (ESB) service has been customized. Only ESB services that have been delivered as a part of the Oracle AIA installation will be checked for customizations. Any ESB service that has been added post-installation will not be checked for customizations.

Script Names

customizedesb.bat for Microsoft Windows.

customizedesb.sh for UNIX.

Test Runtime Parameters

Test runtime parameters for this test are set in the AIADiagnosticsConfig.xml file located here: *<aia.home>/diagnostics/config/.*

```
<param name="Check_Service">CHECK_ALL</param>
```

The Check_Service parameter accepts one of two input values:

- CHECK ALL

  When this value is used, the test scans all ESB services to check whether any customizations have been made to the services post-installation. This is the default input value.

- <ESB Service Name(s)>

  When this value is used, the test scans the specified EBS services to check whether any customizations have been made to the specified ESB services post-installation. You can specify multiple ESB service names, separated by commas.

Note: Any input value, other than *CHECK ALL,* that is provided for the Check_Service parameter in the configuration file will be treated as an ESB service name.

Test Results

Test results in the result_CustomizedEsb.log file include the following information:

- Test name
- Test start time
- Diagnostic test result

  Diagnostic test result values include:

  - Service's status has been changed from DISABLED → ENABLED.
  - No reference data found for the service.
  - No ESB service has been customized.
  - Service might have been unregistered.
  - Service has been modified. The last modified time for this service is: *<date and time>.*
- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_CustomizedEsb.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The CustomizedSchema Diagnostic Test

The results of this diagnostic test indicate whether any schema object has been customized. This test checks only for customizations of schema files (.xsd) in the Enterprise Object Library that do not reside in a Custom folder.

Script Names

customizedschema.bat for Microsoft Windows.

customizedschema.sh for UNIX.

Test Runtime Parameters

Test runtime parameters for this test are set in the AIADiagnosticsConfig.xml file located here: *<aia.home>/diagnostics/config/.*

```
<param name="Check_Schema">CHECK_ALL</param>
```

Copyright © 2009, Oracle. All rights reserved.

The Check_Schema parameter accepts one of two input values:

- CHECK ALL

  When this value is used, the test scans all schema files to check whether any customizations have been made to the files post-installation. This is the default input value.

- <Schema Name(s)>

  When this value is used, the test scans the specified schemas to check whether any customizations have been made to the files post-installation. You can specify multiple schema names, separated by commas. Do not include the .xsd file extension in the schema name.

Note: Any input value, other than *CHECK ALL,* that is provided for the Check_Service parameter in the configuration file will be treated as a schema name.

Test Results

Test results in the result_CustomizedSchema.log file include the following information:

- Test name
- Test start time
- Diagnostic result

  Diagnostics result values include:

  - No schema object has been customized.
  - The following schema file(s) were modified: *<List of files>.*
  - The details of the individual files can be viewed in: *<directory path>.*

- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_CustomizedSchema.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The ExtendedSchema Diagnostic Test

The results of this diagnostic test indicate whether any schema object has been extended. This test checks only for extensions of schema files (.xsd) in the Custom folders in the Enterprise Object Library.

Script Names

extendedschema.bat for Microsoft Windows.

extendedschema.sh for UNIX.

Test Runtime Parameters

Test runtime parameters for this test are set in the AIADiagnosticsConfig.xml file located here: *<aia.home>/diagnostics/config/.*

```
<param name="Check_Schema">CHECK_ALL</param>
```

The Check_Schema parameter accepts one of two input values:

- CHECK ALL

  When this value is used, the test scans all schema files to check whether any extensions have been made to the files post-installation. This is the default input value.

- <Schema Name(s)>

  When this value is used, the test scans the specified schemas to check whether any extensions have been made to the files post-installation. You can specify multiple schema names, separated by commas. Do not include the .xsd file extension in the schema name.

Note: Any input value, other than *CHECK ALL,* that is provided for the Check_Service parameter in the configuration file will be treated as a schema name.

Test Results

Test results in the result_ExtendedSchema.log file include the following information:

- Test name
- Test start time
- Diagnostic result

  Diagnostics result values include:

  - No schema object has been extended.
  - The following schema file(s) were modified: *<list of files>.*
  - The details of the individual files can be viewed in: *<directory path>.*
- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_ExtendedSchema.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

Copyright © 2009, Oracle. All rights reserved.

For more information, see Chapter 27: Running Diagnostic Tests.

The SeedData Diagnostic Test

The results of this diagnostic test indicate whether any of the domain value maps (DVMs) and cross references (XREFs) that were provided at the time of installation have been customized. The test checks only for customizations of DVM and XREF metadata.

Script Names

seeddata.bat for Microsoft Windows.

seeddata.sh for UNIX.

Test Runtime Parameters

No runtime parameters exist for this diagnostic test.

Test Results

Test results in the result_SeedData.log file include the following information:

- Test name
- Test start time
- Xref Data <*has or has not*> been modified
- DVMs <have or have not> been modified
- The following files have been changed: <*list of files*>.
- The details of the individual file changes can be viewed in *AIA_HOME\util\Diagnostics\ChangedFiles\<testname_timestamp>\.*
- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_SeedData.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The XdkTest Diagnostic Test

The results of this diagnostic test indicate whether the XDK/transformation in BPEL is working.

Note: This test leverages Composite Application Validation System (CAVS) functionality.

For more information about CAVS, see Working with the CAVS.

Script Names

xdktest.bat for Microsoft Windows.

xdktest.sh for UNIX.

Test Runtime Parameters

Test runtime parameters for this test are set in the AIADiagnosticsConfig.xml file located here:

<aia. home >/diagnostics/config/.

```
<param name="Test_Definition_ID"><test_definition_id></param>
```

Test Results

Test results in the result_XdkTest.log file include the following information:

- Test name
- Test start time
- Name of the test: TransformationDiagnostics
- Final status of the test: *Passed or Failed.*

  If the status is *Passed,* the test was successful.

  If the status is *Failed,* the base transformation is not working or the diagnostic BPEL process could not be invoked by the CAVS. Use the test instance ID that is provided in the test results to view test details in the CAVS.

- Test definition ID
- Test instance ID

  This is the test instance ID that captured a particular execution of the test definition ID. Use this test instance ID to search for detailed test results in the CAVS.

- Individual XPath details.

  Provides the status of each individual XPath based on the comparison of the expected and actual output. Also provides XPath sequence IDs and values

- Test end time

Copyright © 2009, Oracle. All rights reserved.

- Total time taken

If any errors occurred during test execution, error details can be found in error_XdkTest.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The BpelEsbCommunication Diagnostic Test

The results of this diagnostic test indicate whether the communication between ESB and BPEL is working.

Note: This test leverages CAVS functionality.

For more information about the CAVS, see Working with the CAVS.

Script Names

bpelesbcommunication.bat for Microsoft Windows.

bpelesbcommunication.sh for UNIX.

Test Runtime Parameters

Test runtime parameters for this test are set in the AIADiagnosticsConfig.xml file located here: *<aia.home>/diagnostics/config/.*

```
<param name="Test_Definition_ID"><test_definition_id></param>
```

Test Results

Test results in the result_BpelEsbCommunication.log file include the following information:

- Test name
- Test start time
- Name of the test: EsbBpelCommunicationDiagnostics
- Final status of the test: *Passed* or *Failed.*

  If the status is *Passed,* the test was successful.

  If the status is *Failed,* the base transformation is not working or the diagnostic BPEL process could not be invoked by the CAVS. Use the test instance ID that is provided in the test results to view test details in the CAVS.

- Test definition ID

- Test instance ID

  This is the test instance ID that captured a particular execution of the test definition ID. Use this test instance ID to search for detailed test results in the CAVS.

- Individual XPath details.

  Provides test statuses for all of the XPath criteria in the CAVS test definition that are used for this diagnostic test. Also provides XPath sequence IDs and values.

- Test end time
- Total time taken

Note: In the case of this test, if the *Executed Successfully* test status appears in the command prompt window, it does not necessarily mean that the base transformation was done accurately. It only suggests that the test ran completely without any errors.

If any errors occurred during test execution, error details can be found in error_BpelEsbCommunication.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The CustomXpath Diagnostic Test

This diagnostic test verifies whether Oracle AIA custom XPath functions are working correctly. The following Oracle AIA custom XPath functions are tested:

- aia:getEBMHeaderSenderSystemNode
- aia:getSystemProperty
- aia:getSystemModuleProperty
- aia:getServiceProperty
- aia:getCorrectiveAction
- aia:getErrorMessage
- aia:getNotificationRoles
- aia:isTraceLoggingEnabled

Note: This test leverages CAVS functionality.

For more information about CAVS, see Working with the CAVS.

Copyright © 2009, Oracle. All rights reserved.

Script Names

customxpath.bat for Microsoft Windows.

customxpath.sh for UNIX.

Test Runtime Parameters

Test runtime parameters for this test are set in the AIADiagnosticsConfig.xml file located here: *<aia.home>/diagnostics/config/.*

```
<param name="Definition_IDs" ><test definition ID></param>
```

You can specify multiple test definition IDs, separated by commas.

Test Results

Test results in the result_CustomXpath.log file include the following information:

- Test name
- Test start time
- Total number of tests executed

  This is the number of tests that were invoked in the CAVS.
- Name of the test
- Final status of the test: *Passed* or *Failed.*

  If the status is *Passed,* the test was successful.

  If the status is *Failed,* the base transformation is not working or the diagnostic BPEL process could not be invoked by the CAVS. Use the test instance ID that is provided in the test results to view test details in the CAVS.
- Test definition ID
- Test instance ID in CAVS

  This is the test instance ID that captured a particular execution of the test definition ID. Use this test instance ID to search for detailed test results in the CAVS.
- Individual XPath details.

  Provides test statuses for all of the XPath criteria in the CAVS test definitions used to run the selected diagnostic tests. Also provides XPath sequence IDs and values.
- Total number of tests executed successfully
- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_CustomXpath.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The DeployedServicesVersion Diagnostic Test

The results of this diagnostic test provide a list of all ESB services and BPEL processes that are deployed in the Oracle AIA system.

Note: Test results are obtained by querying the Business Service Repository (BSR) tables. Therefore, this diagnostic test will not provide expected results if the BSR is not working properly or has not been refreshed to reflect the most current data.

For more information about the BSR, see Chapter 1: Understanding the BSR.

Script Names

deployedservicesversion.bat for Microsoft Windows.

deployedservicesversion.sh for UNIX.

Test Runtime Parameters

No runtime parameters exist for this diagnostic test.

Test Results

Test results in the result_DeployedServicesVersion.log file include the following information:

- Test name
- Test start time
- <SERVICENAME>, <SERVICETYPE>, <LIFECYCLESTATUS>, <SHIPPED VERSION>, and <DEPLOYED_ VERSION> values for each process deployed in the Oracle AIA system.
- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_DeployedServicesVersion.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

Copyright © 2009, Oracle. All rights reserved.

For more information, see Chapter 27: Running Diagnostic Tests.

The TableDefinitions Diagnostic Test

The results of this diagnostic test indicate whether table definitions created in the Oracle AIA schema have been changed.

Script Names

tabledefinitions.bat for Microsoft Windows.

tabledefinitions.sh for UNIX.

Test Runtime Parameters

Test runtime parameters for this test are set in the AIADiagnosticsConfig.xml file located here: *<aia.home>/diagnostics/config/.*

```
<param
name="ListOfTables_File_Location"><aia.home/diagnostics/config/Table
DefinitionsList.xml></param>
```

The ListOfTables_File_Location value is the complete path of the XML file that contains the list of table names for which you want to verify definitions.

For your convenience, Oracle provides the <aia.home>/diagnostics/config/TableDefinitionsList.xml file, which contains all table names defined under the Oracle AIA schema. Comment out the names of tables that you do not want verified in the test. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <TableDefinitions>
    <Table name="ERROR_LOG" />
    <Table name="AIA_ERROR_QUEUE_TABLE" />
    <Table name="BSR_PROVIDER_APPLICATION" />
       < ! --
     <Table name = "BSR_PROVIDER_CONNECTOR"/>
     <Table name = "BSR_PROVIDER_EBS"/>
     <Table name = "BSR_PROVIDER_EXTERNAL"/>
     <Table name = "BSR_PROVIDER_KEYWORD"/>
        - - >
    <Table name="BSR_WSDL_DEPENDENCY" />
    <Table name="BSR_WSDL_INTERFACE" />
    <Table name="CAVS_INSTANCES" />
    <Table name="CAVS_DEFINITIONS" />
  </TableDefinitions>
```

Test Results

Test results in the result_TableDefinitions.log file include the following information:

- Test name
- Test start time

- Total number of tables to be examined
- Diagnostic result

  Diagnostics result values include one of the following:

  - Definitions for the following table(s) were changed: *<table names>.*

    The details of the individual file changes can be viewed in: AIA_HOME\util\Diagnostics\ChangedFiles \<testname_timestamp>\.

  - No Table Definition has changed for the tables listed in the file: *<file path provided in ListOfTables File Location input parameter>.*
- Total number of table definitions examined successfully
- Test end time
- Total time taken

If any errors occurred during test execution, error details can be found in error_TableDefinitions.log. This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

The JMSQueue Diagnostic Test

The results of this diagnostic test indicate whether the Enqueue and the Dequeue operations of JMS are working fine.

Note: This test leverages CAVS functionality.

For more information, see Working with the CAVS.

Script Names

jmsqueuetest.bat for Microsoft Windows.

jmsqueuetest.sh for UNIX.

Test Runtime Parameters

Test runtime parameters for this test are set in the AIADiagnosticsConfig.xml file located here: *<aia.home>/diagnostics/config/.*

```
<param name="Test_Definition_ID"><test_definition_id></param>
```

You can specify multiple test definition IDs, separated by commas.

Copyright © 2009, Oracle. All rights reserved.

Test Results

Test results in the result_JMSQueue.log file include the following information:

- Test name
- Test start time
- Name of the test: JMSQueue
- Final status of the test: *Passed* or *Failed.*

  If the status is *Passed,* the test was successful.

  If the status is *Failed,* the Enqueue operation is not working or the diagnostic BPEL process could not be invoked by the CAVS. Use the test instance ID provided in the test results to view test details in the CAVS.

- Test definition ID
- Test instance ID

  This is the test instance ID that captured a particular execution of the test definition ID. Use this test instance ID to search for detailed test results in the CAVS.

- Individual XPath details.

  Provides test statuses for all XPath criteria in the CAVS test definition used for this diagnostic test. Also provides XPath sequence IDs and values.

- Test end time
- Total time taken

Note: In the case of this test, if the *Executed Successfully* test status appears in the command prompt window, it does not necessarily mean that the base transformation was done accurately. It only suggests that the test ran completely without any errors.

If any errors occurred during test execution, error details can be found in error_JMSQueue.log. This log stores the test instance when the test execution status is either Partially successful or Completely failed. The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 27: Running Diagnostic Tests.

Chapter 27: Running Diagnostic Tests

This chapter discusses how to:

- Run a single diagnostic test.
- Run diagnostic tests in batch.

Running a Single Diagnostic Test

Use this procedure to run a single test in the Diagnostics Framework.

To run a single diagnostic test:

1. If the diagnostic test has available runtime parameters listed in its description, enter them in AIADiagnosticsConfig.xml located here: <aia.home>/diagnostics/config/.
2. Run the test script using a command prompt.

   Test scripts are located here: *<aia.home>/diagnostics/bin/*
3. The test status appears in the command prompt window:
   - Completely successful.

     Indicates that all tests have been successfully run.
   - Partially successful.

     Indicates that some tests have failed.
   - Completely failed.

     Indicates that all tests have failed.

Test results are stored in logs located here: *<aia.home>/diagnostics/logs/<test_name>/.* Test logs are stored in separate folders identified by test names that are provided in test scripts and the Oracle Application Integration Architecture (AIA) Diagnostics Configuration file.

Each test log folder will contain two files:

- result_<*test name*>.log

  This log stores the results of the test execution. Test results are appended in this log following the execution of the test, regardless of the test status.
- error_<test name>.log

This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

For more information, see Chapter 26: Describing the Diagnostic Tests.

Running Diagnostic Tests in Batch

Use this procedure to run a batch of tests in the Diagnostics Framework.

To run diagnostic tests in batch:

1. If a diagnostic test that you want to run in a batch has available runtime parameters listed in its description, enter them in AIADiagnosticsConfig.xml located here: <aia.home>/diagnostics/config/.
2. Access the Oracle AIA Diagnostics configuration file, AIADiagnosticsConfig.xml, located in *<aia.home>/diagnostics/config/ and* enter the names of the diagnostic tests that you want to run in batch in the batch section.
3. Run the batch-diagnostics test script using a command prompt. In Microsoft Windows, the script name is *batchdiagnostics.bat.* In UNIX, the script name is *batchdiagnostics.sh.*
4. Test scripts are located here: *<aia.home>/diagnostics/bin/*
5. The test status appears in the command prompt window:
   - Completely successful.

     Indicates that all tests have been successfully run.
   - Partially successful.

     Indicates that some tests have failed.
   - Completely failed.

     Indicates that all tests have failed.

Test results are stored in a log located here: *<aia.home>/diagnostics/logs/Batch/.* The test log folder will contain two files:

- result_Batch.log

  This log stores the results of the test execution. Test results are appended in this log following the execution of the test, regardless of the test status.

- error_Batch.log

  This log stores the test instance when the test execution status is either *Partially successful* or *Completely failed.* The log contains complete details of the errors that occurred during the test execution to cause the test failures.

Copyright © 2009, Oracle. All rights reserved.

For more information, see Chapter 26: Describing the Diagnostic Tests.

Chapter 28: Viewing Diagnostic Test Logs

This chapter discusses how to view diagnostic test result and error logs in Oracle Enterprise Manager (EM).

Viewable logs include the result and error logs that are generated by running individual and batch diagnostic tests that are available in the Oracle Application Integration Architecture (AIA) Diagnostics Framework.

For more information, see Chapter 26: Describing the Diagnostic Tests.

Viewing Diagnostic Test Logs in EM

This section discusses how to view diagnostic test logs in EM.

Pages Used to View Diagnostic Test Logs in EM

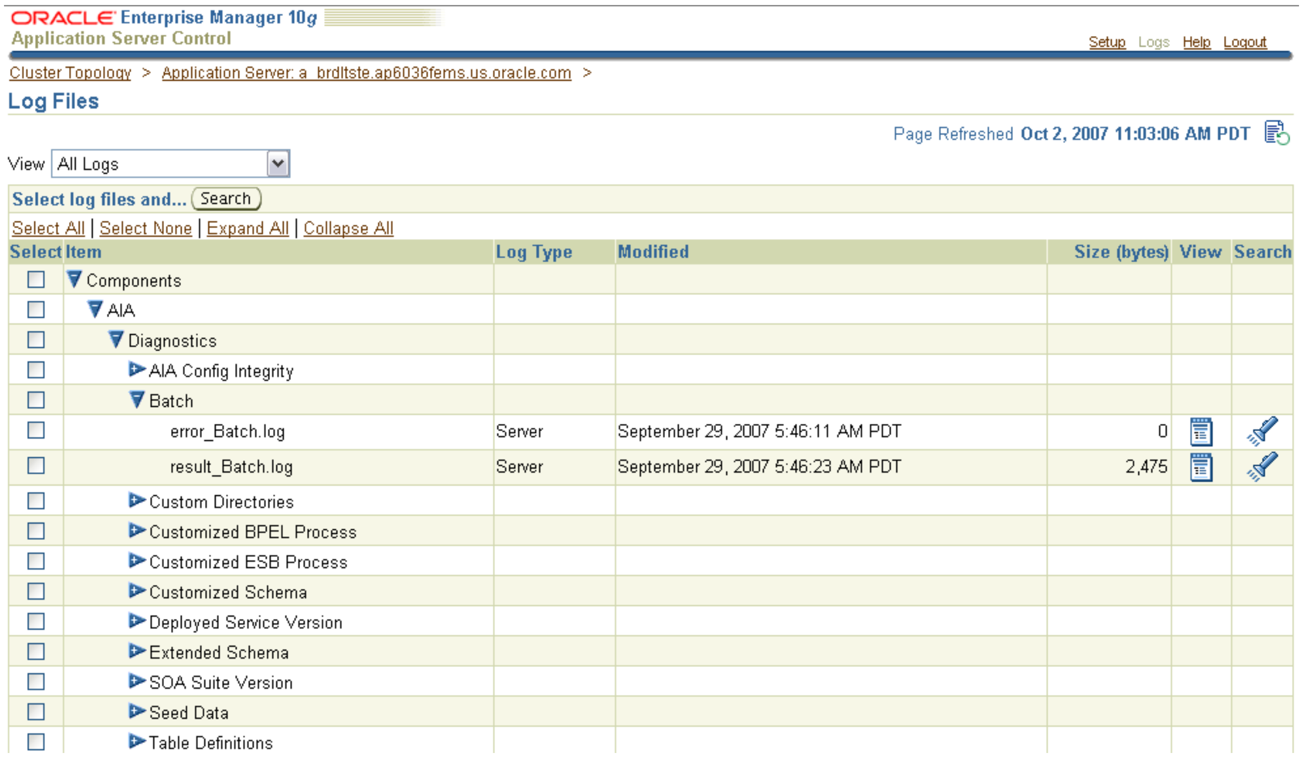
Page Name	Navigation	Usage
Log Files	1. Access the Oracle Enterprise Manager -Cluster Topology page. 2. In the Name column, click the oc4j_<instance name> application server link. 3. Click the Logs link located in the page header or footer. 4. On the Log Files page, under the Components, AIA node hierarchy, expand Diagnostics to access diagnostic test result and error logs.	Access pages that enable you to search for log messages and view log files.
Search Logs	1. Access the Oracle Enterprise Manager -Cluster Topology page. 2. In the Name column, click the oc4j_<instance name> application server link. 3. Click the Logs link located in the page header or footer. 4. On the Log Files page, under the Components, AIA node hierarchy, expand Diagnostics to be able to	Search for specific log messages in a selected diagnostic test result or error log. Use the page elements to search for specific log messages in the selected diagnostic test log. Search results are displayed at the bottom of the page.

Page Name	Navigation	Usage
	search for diagnostic test result and error log messages. 5. Click the Search contents of this log file button for the log in which you want to search for specific log messages.	
View Log: <logname>	1. Access the Oracle Enterprise Manager -Cluster Topology page. 2. In the Name column, click the oc4j_<instance name> application server link. 3. Click the Logs link located in the page header or footer. 4. On the Log Files page, under the Components, AIA node hierarchy, expand Diagnostics to be able to view the diagnostic test result or error log. 5. Click the View this log file button for the selected diagnostic test log.	View log messages and other contents of a selected diagnostic test result or error log.

Viewing Diagnostic Test Result and Error Logs in EM

Access the Log Files page.

Copyright © 2009, Oracle. All rights reserved.


ORACLE Enterprise Manager 10g
Application Server Control

Setup Logs Help Logout

Cluster Topology > Application Server: a_brdltste.ap6036fems.us.oracle.com >

Log Files

Page Refreshed Oct 2, 2007 11:03:06 AM PDT

View All Logs

Select log files and... Search

Select All | Select None | Expand All | Collapse All

Select	Item	Log Type	Modified	Size (bytes)	View	Search
☐	▼ Components					
☐	▼ AIA					
☐	▼ Diagnostics					
☐	▶ AIA Config Integrity					
☐	▼ Batch					
☐	error_Batch.log	Server	September 29, 2007 5:46:11 AM PDT	0	📄	🔦
☐	result_Batch.log	Server	September 29, 2007 5:46:23 AM PDT	2,475	📄	🔦
☐	▶ Custom Directories					
☐	▶ Customized BPEL Process					
☐	▶ Customized ESB Process					
☐	▶ Customized Schema					
☐	▶ Deployed Service Version					
☐	▶ Extended Schema					
☐	▶ SOA Suite Version					
☐	▶ Seed Data					
☐	▶ Table Definitions					

Log Files page

To work with Oracle AIA diagnostic test result and error logs, expand the AIA and Diagnostics nodes.

Chapter 29: Adding Diagnostic Tests to the Diagnostics Framework

This chapter discusses how to add your own diagnostic tests to the Diagnostics Framework.

Adding Diagnostic Tests

The Diagnostics Framework is designed to enable you to add and run your own diagnostic tests using the framework.

To add a diagnostic test to the Diagnostics Framework:

1. Create the implementing Java class for the new test.

   Ensure that the Java class implements the Oracle Application Integration Architecture (AIA) Diagnostics interface, IAIADiagnostics. This interface has two methods, *run* and *runBatch.*

   The input to both of these methods is a HashMap of the runtime parameters as configured in the Oracle AIA Diagnostics Configuration file, AIADiagnosticsConfig.xml, located in *<aia.home>/diagnostics/config/.*

2. Add the class and test runtime parameter details to AIADiagnosticsConfig.xml.

3. Create the executable script for the test and save it to this directory: <aia.home>/diagnostics/bin/.

4. Both the *run* and *runBatch* methods should return a String [ ].

   The first string in this array is written to the result log. The second string is written to the error log. The third string indicates the status of the test and contains one of the following values:

   - COMPLETE SUCCESS
   - PARTIAL SUCCESS
   - COMPLETE FAILURE

5. To enable viewing of these test log files in the Oracle Enterprise Manager (EM) Console, add the following sections to the AIADiagnostics.xml file located in <Oracle Home>/j2ee/home/applications/ascontrol/ascontrol/WEB-INF/config/registration.

```
<log path='<aia.home>/diagnostics/logs/<test name>/result_<test
name>.log'>
  <logreader type='oracle.core.ojdl.reader.ODLTextLogReader' />
    <logviewer LogType='SERVER'>
       <property name='COMPONENT_TYPE' value='AIA'/>
       <property name='ODL_FOLDER_NAME' value='Diagnostics'/>
       <property name='COMPONENT_NAME' value='<test name>'/>
       <property name='displayPath'
```

```
value='/%COMPONENT_TYPE%/%ODL_FOLDER_
        NAME%/%COMPONENT_NAME%/%LOG_NAME%'/>
    </logviewer>
 </log >
<log path='<aia.home>/diagnostics/logs/<test name>/error_<test
name>.log'>
  <logreader type='oracle.core.ojdl.reader.ODLTextLogReader'/>
    <logviewer LogType='SERVER'>
      <property name='COMPONENT_TYPE' value='AIA'/>
      <property name='ODL_FOLDER_NAME' value='Diagnostics'/>
      <property name='COMPONENT_NAME' value='<test name>'/>
      <property name='displayPath'
value='/%COMPONENT_TYPE%/%ODL_FOLDER_^>
        NAME %/% COMPONENT_NAME %/%LOG_NAME %'/>
  </logviewer>
</log >
```

Note: The value of <*aia.home*> should be complete and correct Oracle AIA Home value.

Interface Details

Following is the IAIADiagnostics interface definition:

```
package oracle.apps.aia.core.diagnostics;
import java.util.HashMap;
public interface IAIADiagnostics {
   public String [] run(HashMap params) throws Exception;
   public String [] runBatch(HashMap params) throws Exception;
}
```

Copyright © 2009, Oracle. All rights reserved.

Part 5: Working with Oracle AIA Developer Tools

Chapter 30: Working with the Artifacts Generator

This chapter provides an overview of the Oracle Application Integration Architecture (AIA) Artifacts Generator and provides detailed discussions about the features of the Artifacts Generator.

Overview

The Artifacts Generator is a development tool that can help jumpstart your Oracle AIA development by autogenerating much of the common code needed to create Oracle AIA application business connector (ABC) service implementations.

The Artifacts Generator is a command line Apache Ant tool that uses the FreeMarker template engine to generate complete and compilable ABC service implementations in BPEL.

The Artifacts Generator requires Java 1.5 and Apache Ant 1.6.5.

For information about using the Artifacts Generator, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Using the Artifacts Generator."

Understanding FreeMarker

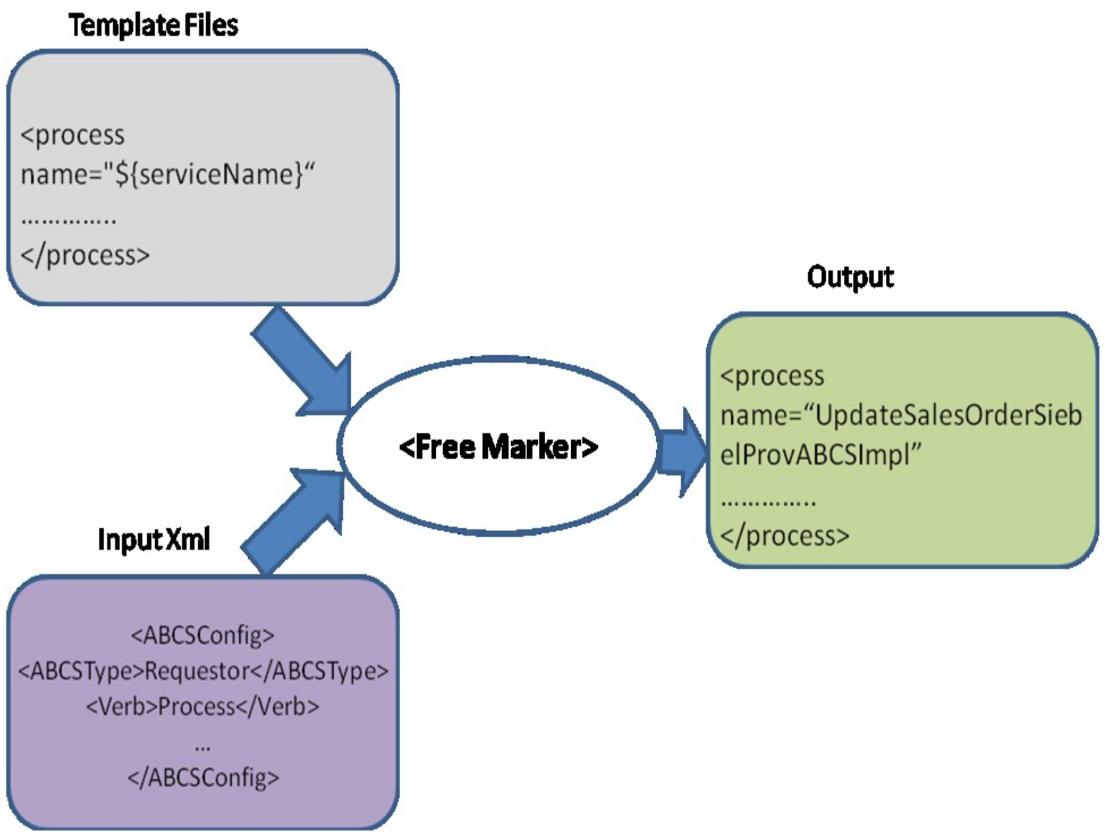
FreeMarker is an open-source tool used to generate text output, anything from HTML to autogenerated source code, based on templates. It is not an end-user application, but rather is a package of Java classes that programmers can embed in their products.

FreeMarker is designed to generate application code based on the Model View Controller (MVC) pattern. The MVC pattern for generating source code separates the work of architects (Oracle AIA service code authors) from the work of programmers, enabling each participant to focus their skills on what they do best.

For example, architects developing programming models can specify what the service code should be without creating the need for programmers to change or recompile code. Likewise, templates do not become polluted with complex program fragments. This is because using MVC, the application logic (Java programs) and service code (FreeMarker templates) are maintained separately.

This separation is useful even for projects in which the architect and programmer are the same person because it helps to keep the application clear and maintainable.

Although FreeMarker has some programming capabilities, it is not a full-blown programming language, such as PHP. Instead, Java programs prepare the data to be displayed (such as issue SQL queries) and FreeMarker generates textual pages that display the prepared data using templates. FreeMarker is not a web application framework, but is suitable as a component in a web application framework. The FreeMarker engine itself knows nothing about HTTP or servlets; it simply generates text.



FreeMarker uses templates and input xml to generate textual output

Describing Artifacts Generator Features

The Artifacts Generator generates the following artifacts to create Oracle AIA ABC service implementations in BPEL:

- Projects
- ABC service interfaces
- ABC service implementations using BPEL
- Enterprise business service (EBS) and application service invocations
- Message exchange patterns
- Logging, error handling, and fault policies
- Composite Application Validation System (CAVS) enablement and dynamic endpoint locations
- Naming conventions
- Transformations with root element and header population
- ABC service properties for the Oracle AIA configuration properties file

Copyright © 2009, Oracle. All rights reserved.

- Best-practice settings

For information about using the Artifacts Generator, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide,* "Using the Artifacts Generator."

Project Generation

The Artifacts Generator generates the complete and compilable project for an ABC service implementation.

Following is a description of the project features generated by the Artifacts Generator:

- .wsdl

  This is the web service interface for the ABC service.

- .bpel

  This is the BPEL implementation.

- Bpel.xml

  This is the deployment descriptor.

- .jpr file

  This is the BPEL project descriptor.

- Build.properties
- Build.xml
- EBM_to_Fault.xsl
- A reference .wsdl file for each target service invocation

  These are the BPEL-related PartnerLink definitions for the target service.

- Header transformation .xsl for each application business message (ABM) to enterprise business message (EBM) transformation
- Business message transformation .xsl for each ABM to EBM transformation
- Custom business message transformation .xsl for each ABM to EBM transformation
- Correlation set properties file for correlating asynchronous request-and-response messages
- File containing Oracle AIA configuration properties for the ABC service
- Fault policy file
- Abstract service interface (.wsdl file) for the ABC service extension service

ABC Service Interface Generation

The Artifacts Generator generates the abstract WSDL for the service interface needed for the ABC service implementation according to Oracle AIA programming guidelines and standards.

Following is a description of the ABC service interface features generated by the Artifacts Generator:

- .wsdl file name and definition name generated according to naming conventions
- Declaration and import of all of the necessary namespaces
- Definition of all relevant message types needed for input, as well as for output for service operations
- Definition of PortTypes
- Definition of the operations needed for service initiation and instantiation, as well as asynchronous callback, if necessary
- PartnerLink definitions for starting the process, as well as any necessary asynchronous callbacks into the process
- Generation of service annotations needed for publishing the Oracle AIA service artifact into the Business Service Repository (BSR)

For more information about the BSR, see Chapter 1: Understanding the BSR.

- Correlation properties for asynchronous request and response

ABC Service BPEL Implementation Generation

The Artifacts Generator generates the BPEL implementation file according to Oracle AIA programming guidelines and standards.

Following is a description of the ABC service implementation features generated:

- BPEL file name and process name generated according to naming conventions
- Namespace of the process generated according to naming conventions
- Namespace declaration for the required namespaces
- Service-invocation-related code for each outbound interaction
- Code needed for error handlers
- Code to CAVS-enable both the application and EBS service invocations

For more information about the CAVS, see Chapter 8: Understanding the CAVS.

- Code for all message exchange patterns

Copyright © 2009, Oracle. All rights reserved.

- Code for calling extensibility operations
- Code for calling transformation files

Enterprise Business Service and Application Service Invocation Generation

The Artifacts Generator generates the code needed to invoke an EBS or any application service. The message exchange patterns supported include:

- Synchronous request/response
- Asynchronous fire-and-forget
- Asynchronous request/response

Following is a description of the EBS and application service invocation features generated:

- The BPEL scope construct for each target service invocation
- Variables needed to hold the request, response, and fault messages pertaining to the target service
- PartnerLink definition for the target service
- CAVS code for each target service

For more information about the CAVS, see Chapter 8: Understanding the CAVS.

- The assign activity for EBS invocation
- The invoke activity for each service invocation
- The receive activity for asynchronous responses
- Extension-related code before and after service invocation
- An error handler to capture faults thrown by the target service in the case of a request/response invocation

Message Exchange Pattern Generation

The Artifacts Generator generates code for three message exchange patterns (MEPs):

- Synchronous request/response
- Asynchronous fire-and-forget
- Asynchronous request/response

Following is a description of the MEP features generated:

- Invoke activity with the inputVariable declaration for the fire-and-forget MEP

- Invoke activity with inputVariable and OutputVariable declarations for the synchronous request/response MEP
- Invoke and receive activities for the asynchronous request/response MEP. Also generates input and output variables.
- The operation in WSDL for each receive activity for the asynchronous request/response MEP
- The correlation in WSDL and BPEL for each asynchronous request/response

Logging, Error Handling, and Fault Policy Generation

The Artifacts Generator generates code for handling errors at the process level, as well as at each service invocation level. It also creates the fault policy file with default values.

For more information about error handling and logging, see Chapter 20: Understanding Error Handling and Logging.

Following is a description of the logging, error handling, and fault policy features generated:

- Fault policy file with default values
- Process-level error handlers for catching binding faults and remote faults, as well as subLanguageExecutionFault and catch-all faults
- Invocation of the Oracle AIA Error Handler Process

  The AsyncError handler process is invoked in a catch-all path
- Rethrow of the fault, in the case of synchronous request/response processes
- Logging is performed at the following stages:
  - Start and end of each process
  - Before and after each transformation
  - Before and after each target service invocation
  - Start and end of each of error handler branch
- EBM_to_Fault transformation file

CAVS Enablement and Dynamic Endpoint Location Generation

The Artifacts Generator generates CAVS code for both EBS invocation and application service invocation.

For more information about error handling and logging, see Chapter 8: Understanding the CAVS.

Copyright © 2009, Oracle. All rights reserved.

Following is a description of the CAVS enablement and dynamic endpoint location features generated:

- For EBS invocation, adds MessageProcessingInstruction section in the application business message to enterprise business message transformation
- For application service invocation, generates GetSenderSystemID.xsl, adds the assign statement, and adds Java code to the dynamicall invoke application service

Naming Convention Generation

The Artifacts Generator generates all of the files in the project using proper naming conventions.

Following is a description of the artifacts for which proper naming conventions are generated:

- Files
- Services
- Namespaces
- Namespace prefixes
- PartnerLinks
- Scopes
- BPEL variables
- Service operations
- Message types

Transformation Generation with Root Element and Header Population

The Artifacts Generator generates transformation files with root element business message transformation, header transformation, and custom message transformation.

Following is a description of the transformation file features generated:

- A separate transformation file for header population
- Autogenerates reading of configuration files
- Autogenerates population of most header properties based on naming conventions
- Namespaces for all necessary schemas
- Namespace prefixes consistent with .wsdl and .bpel implementation
- Generates standard custom elements

ABC Service Configuration Property Generation

The Artifacts Generator generates the ABC service ServiceConfiguration section of the Oracle AIA configuration properties file. All generic configuration properties related to the service are generated by the Artifacts Generator. The ABC service developer just needs to copy this section into the Oracle AIA configuration properties file to complete the properties for the ABC service.

Following is a description of the ABC service configuration properties generated:

- Service name
- Default system ID
- Property for each extension point
- Routing properties

Best-Practice Setting Generation

The Artifacts Generator generates BPEL technology best-practice settings in BPEL.xml, including .wsdl and .wsdl runtime locations.

Copyright © 2009, Oracle. All rights reserved.

Chapter 31: Working with the XSLT Mapping Analyzer

This chapter provides an overview of and discusses how to use the XSLT Mapping Analyzer.

You can download the latest version of the XSLT Mapping Analyzer and further supporting documentation from MetaLink **Note** 782351.1.

Understanding the ABC Service XSLT Mapping Analyzer

The XSLT Mapping Analyzer is a tool you can use to generate the mapping information from the application business connector (ABC) service transformation XSL into HTML.

The input of the XSLT Mapping Analyzer is the ABC service transformation XSLT. The ABC service transformation XSLT is an XSL code that maps the source enterprise business message (EBM) to application schema or from application schema to EBM.

To simplify the mapping from the source application schema to the target application schema, the XSLT Mapping Analyzer takes the XSLT file as input and extracts the mappings from the source to target elements and displays the mapping information in a table that is easier to understand.

The XSLT Mapping Analyzer output provides you with a table of input and output element mappings. Information provided in the table includes:

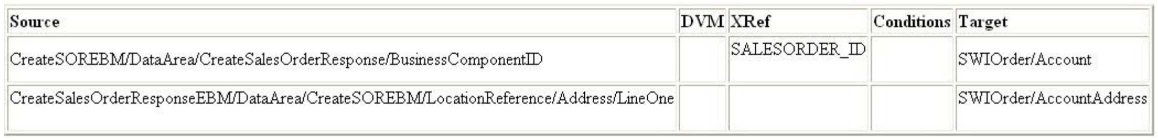
- Source enterprise business object (EBO) or application business object (ABO) element.
- Domain value mapping (DVM) table name.
- Cross-reference (Xref) table name.
- Conditions for the mapping.
- The corresponding EBO or ABO mapping.

Here's an example of an ABC service XSL mapping code:

```
<xsl:template match="/">
    <sbldata:SWIOrder>
        <sbldata:Account>
          <xsl:value-of
select='xref:lookupXRef("SALESORDER_ID","COMMON",/ns0:CreateSOREBM/n
s0:DataArea/ns0:CreateSalesOrderResponse/corecom:BusinessComponentID
,$SenderSystemInstanceCode,false())'/>
        </sbldata:Account>
        <sbldata:AccountAddress>
          <xsl:value-of
select="/ns0:CreateSalesOrderResponseEBM/ns0:DataArea/ns0:CreateSORE
BM/corecom:LocationReference/corecom:Address/corecom:LineOne"/>
        </sbldata:AccountAddress>
</sbldata:SWIOrder>
```

```
    </xsl:template>
```

Following is the corresponding XSLT Mapping Analyzer output:



Source	DVM	XRef	Conditions	Target
CreateSOREBM/DataArea/CreateSalesOrderResponse/BusinessComponentID		SALESORDER_ID		SWIOrder/Account
CreateSalesOrderResponseEBM/DataArea/CreateSOREBM/LocationReference/Address/LineOne				SWIOrder/AccountAddress

Sample output from XSLT Mapping Analyzer

The XSLT Mapping Analyzer can be used as a tool that can enable consistent and semantic data mapping integrity by introspecting the mapping structure of the ABC service transformations provided by the existing Oracle Application Integration Architecture (AIA) process integration packs (PIPs).

Interoperability between ABC services is only possible if all ABC services that need to work together use the exact same mapping, the same DVM and Xref table names, and the right translation functions. To develop a new ABC services to work with an existing PIP ABC services, this tool would help in identifying the mapping semantics in an easier to understand format than the underlying XSL. However, in some cases, the transformation complexity may still necessitate looking up the detailed code in the XSL files.

Once the mapping information is obtained, it is highly recommended that you document these independently in a mapping document or spreadsheet for the given EBO to the various applications as part of a service-oriented architecture (SOA) governance methodology.

The ABC service XSLT Mapping Analyzer uses an ANT-based command-line utility to generate the mapping information for the ABC service XSLs. You run the utility by providing the path of the ABC services XSL for which the mapping document is to be generated.

The XSLT Mapping Analyzer provides mapping information in HTML format, but also produces an intermediate XML file that can be used for further transformations, into CSV or RDF format, for example. The XML format is the master source of all mapping information and the HTML format can be fine-tuned depending upon the options you select in one of the packaging tools.

It will generate the mapping information and place the output files in the folder as specified by the user. If the user has not specified an output directory, the output files will be placed in the same folder as the input XSLs.

The mapping information provides:

- Input element and output element mapping information.
- Descriptions of the functions that are used in the mappings.
- Conditions, if any, for a mapping.
- XREF and DVM mapping information.

Using the ABC Service XSLT Mapping Analyzer

This section discusses how to use the ABC service XSLT Mapping Analyzer, an ANT-based command-line utility.

Copyright © 2009, Oracle. All rights reserved.

To invoke the ABC service XSLT Mapping Analyzer tool in a Windows environment:

1. Add the folder containing generateMappingDoc.bat, located in $AIA_HOME/DeveloperTools/XSLDocGen/bin to the path environment variable.
2. After adding the XSLDocGen folder to the path environment variable, type:

```
generateMappingDoc inputFile={full path of the xsl file}
outputFormat={html} outputDir={outputdirname with path}
```

For example:

```
C:\>generateMappingDoc
inputFile=D:\AIAPIP\TestPIP\ReqABCS\Xform_SiebelProcessOrderReqMsg_t
o_ProcessSalesOrderReqMsg.xsl outputFormat=html
outputDir=D:\AIAPIP\output
```

Note: To get help with running the tool from a command prompt and learn how to specify the tool's parameters, access: `generateMappingDoc help`.

To invoke the ABC services XSLT Mapping Analyzer tool in a Linux environment:

1. Add the $AIA_HOME/DeveloperTools/XSLDocGen/bin folder to the path environment variable.
2. After adding the XSLDocGen folder to the path environment variable, type:

```
sh generateMappingDoc.sh inputFile={full path of the xsl file}
outputFormat={html} outputDir={outputdirname with path}
```

For example:

```
$sh generateMappingDoc
inputFile=/home/bin/AIAPIP/Xform_SiebelProcessOrderReqMsg_to_Process
SalesOrderReqMsg.xsl outputFormat=html
outputDir=/home/bin/AIAPIP/output
```

Note: To get help with running the tool from a command prompt and learn how to specify the tool's parameters, access: `sh generateMappingDoc.sh help`.

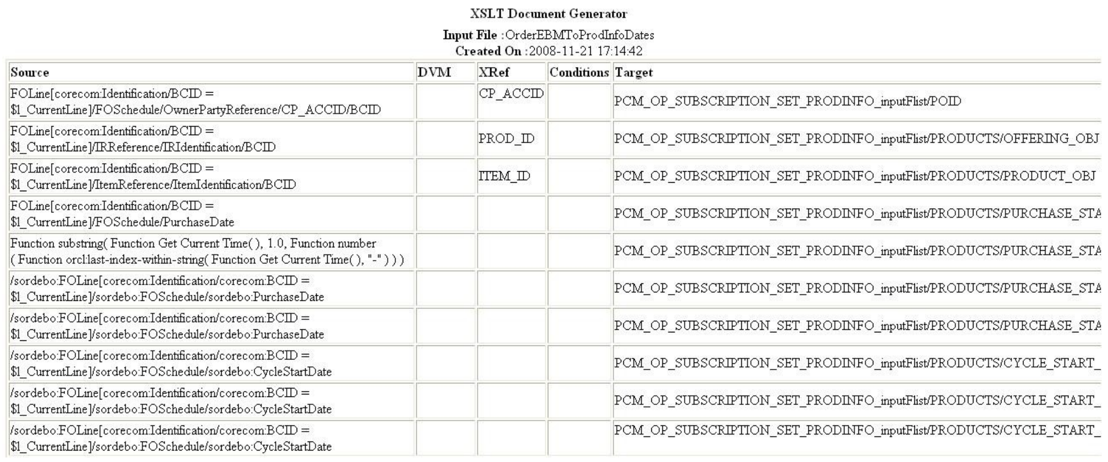
Please note the following:

- ouptutFormat and outputDir parameters are optional.
- There are two output files that are generated, one in XML format and one in HTML format. The files are named <inputxslfilename>_doc.xml and <inputxslfilename>_doc.html.
- Ensure that the AIA_HOME path variable is set and that aiaenv.sh or aiaenv.bat is present in the AIA_HOME/bin folder.

Once the command is executed, the output mapping information in XML and HTML formats is placed in the output directory as specified by the user. If the user has not specified an output directory, the output files will be placed in the same directory as the input XSL files.

Copyright © 2009, Oracle. All rights reserved.

This screenshot illustrates a sample output:



XSLT Document Generator

Input File :OrderEBMToProdInfoDates
Created On :2008-11-21 17:14:42

Source	DVM	XRef	Conditions	Target
FOLine[corecom:Identification/BCID = $1_CurrentLine]/FOSchedule/OwnerPartyReference/CP_ACCID/BCID		CP_ACCID		PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/POID
FOLine[corecom:Identification/BCID = $1_CurrentLine]/IRReference/IRIdentification/BCID		PROD_ID		PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/OFFERING_OBJ
FOLine[corecom:Identification/BCID = $1_CurrentLine]/ItemReference/ItemIdentification/BCID		ITEM_ID		PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PRODUCT_OBJ
FOLine[corecom:Identification/BCID = $1_CurrentLine]/FOSchedule/PurchaseDate				PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_STA
Function substring(Function Get Current Time( ), 1.0, Function number (Function orcl:last-index-within-string(Function Get Current Time( ), "-")))				PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_STA
/sordebo:FOLine[corecom:Identification/corecom:BCID = $1_CurrentLine]/sordebo:FOSchedule/sordebo:PurchaseDate				PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_STA
/sordebo:FOLine[corecom:Identification/corecom:BCID = $1_CurrentLine]/sordebo:FOSchedule/sordebo:PurchaseDate				PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_STA
/sordebo:FOLine[corecom:Identification/corecom:BCID = $1_CurrentLine]/sordebo:FOSchedule/sordebo:CycleStartDate				PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/CYCLE_START_
/sordebo:FOLine[corecom:Identification/corecom:BCID = $1_CurrentLine]/sordebo:FOSchedule/sordebo:CycleStartDate				PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/CYCLE_START_
/sordebo:FOLine[corecom:Identification/corecom:BCID = $1_CurrentLine]/sordebo:FOSchedule/sordebo:CycleStartDate				PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/CYCLE_START_

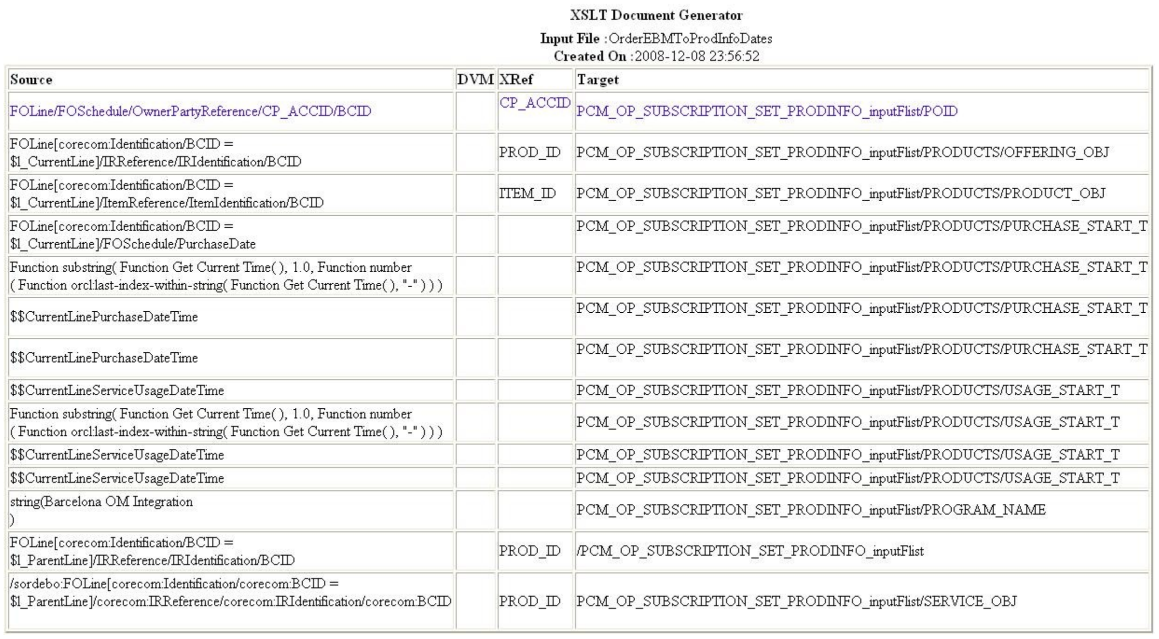
ABC service XSLT Mapping Analyzer sample output

Source	Displays the input xpath of the ABM or EBM.
DVM	Displays the DVM table name.
Xref	Displays the XREF.
Conditions	Displays the comments and conditions for the mapping.
Target	Displays the Xpath of the output elements.

The XSLT Mapping Analyzer also allows visual identification of extensions to delivered XSLT mappings. This enables you to review updated mappings when upgrading a process integration pack (PIP). This can help you identify conflicting or incompatible mappings provided by the newly upgraded PIP, which earlier, were handled through extensions. This can help pinpoint the implementation effort required for PIP upgrades in cases in which there were XSLT mapping extensions used with the original PIP.

Following is a sample output with extensions highlighted in blue.

Copyright © 2009, Oracle. All rights reserved.



XSLT Document Generator

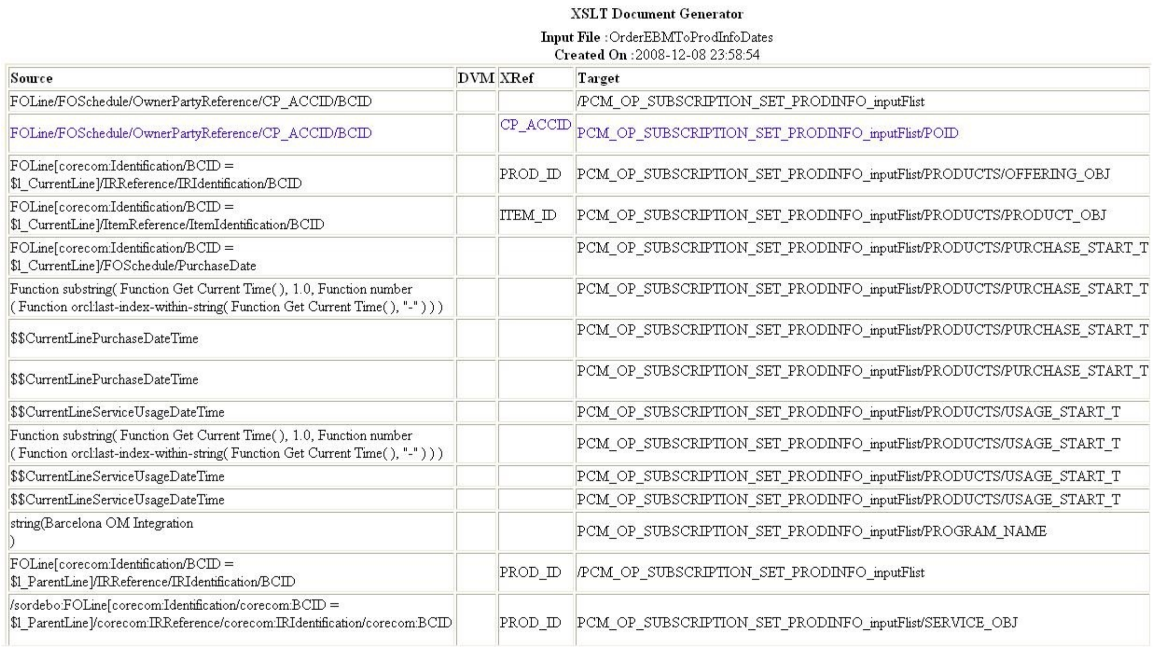
Input File :OrderEBMToProdInfoDates

Created On :2008-12-08 23:56:52

Source	DVM	XRef	Target
FOLine/FOSchedule/OwnerPartyReference/CP_ACCID/BCID		CP_ACCID	PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/POID
FOLine[corecom:Identification/BCID = $l_CurrentLine]/IRReference/IRIdentification/BCID		PROD_ID	PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/OFFERING_OBJ
FOLine[corecom:Identification/BCID = $l_CurrentLine]/ItemReference/ItemIdentification/BCID		ITEM_ID	PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PRODUCT_OBJ
FOLine[corecom:Identification/BCID = $l_CurrentLine]/FOSchedule/PurchaseDate			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_START_T
Function substring(Function Get Current Time( ), 1.0, Function number (Function orcl:last-index-within-string(Function Get Current Time( ), "-")))			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_START_T
$$CurrentLinePurchaseDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_START_T
$$CurrentLinePurchaseDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_START_T
$$CurrentLineServiceUsageDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/USAGE_START_T
Function substring(Function Get Current Time( ), 1.0, Function number (Function orcl:last-index-within-string(Function Get Current Time( ), "-")))			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/USAGE_START_T
$$CurrentLineServiceUsageDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/USAGE_START_T
$$CurrentLineServiceUsageDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/USAGE_START_T
string(Barcelona OM Integration)			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PROGRAM_NAME
FOLine[corecom:Identification/BCID = $l_ParentLine]/IRReference/IRIdentification/BCID		PROD_ID	/PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist
/sordebo:FOLine[corecom:Identification/corecom:BCID = $l_ParentLine]/corecom:IRReference/corecom:IRIdentification/corecom:BCID		PROD_ID	PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/SERVICE_OBJ

Sample XSLT Mapping Analyzer output displaying an extension in blue

The following screenshot displays mapping output for an updated PIP with one extension wherein the updated mapping conflicts with the extension.



XSLT Document Generator

Input File :OrderEBMToProdInfoDates

Created On :2008-12-08 23:58:54

Source	DVM	XRef	Target
FOLine/FOSchedule/OwnerPartyReference/CP_ACCID/BCID			/PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist
FOLine/FOSchedule/OwnerPartyReference/CP_ACCID/BCID		CP_ACCID	PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/POID
FOLine[corecom:Identification/BCID = $l_CurrentLine]/IRReference/IRIdentification/BCID		PROD_ID	PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/OFFERING_OBJ
FOLine[corecom:Identification/BCID = $l_CurrentLine]/ItemReference/ItemIdentification/BCID		ITEM_ID	PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PRODUCT_OBJ
FOLine[corecom:Identification/BCID = $l_CurrentLine]/FOSchedule/PurchaseDate			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_START_T
Function substring(Function Get Current Time( ), 1.0, Function number (Function orcl:last-index-within-string(Function Get Current Time( ), "-")))			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_START_T
$$CurrentLinePurchaseDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_START_T
$$CurrentLinePurchaseDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/PURCHASE_START_T
$$CurrentLineServiceUsageDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/USAGE_START_T
Function substring(Function Get Current Time( ), 1.0, Function number (Function orcl:last-index-within-string(Function Get Current Time( ), "-")))			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/USAGE_START_T
$$CurrentLineServiceUsageDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/USAGE_START_T
$$CurrentLineServiceUsageDateTime			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PRODUCTS/USAGE_START_T
string(Barcelona OM Integration)			PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/PROGRAM_NAME
FOLine[corecom:Identification/BCID = $l_ParentLine]/IRReference/IRIdentification/BCID		PROD_ID	/PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist
/sordebo:FOLine[corecom:Identification/corecom:BCID = $l_ParentLine]/corecom:IRReference/corecom:IRIdentification/corecom:BCID		PROD_ID	PCM_OP_SUBSCRIPTION_SET_PRODINFO_inputFlist/SERVICE_OBJ

Sample XSLT Mapping Analyzer output displaying for an updated PIP with one extension wherein the updated mapping conflicts with the extension

Chapter 32: Working with the PIP Auditor

The Process Integration Pack (PIP) Auditor verifies that a PIP's design and development adheres to Oracle Application Integration Architecture (AIA) guidelines. The PIP Auditor contains a number of tests that check for different standards mandated or recommended by Oracle AIA with respect to PIP design and development.

The PIP Auditor generates high-level HTML and detailed XML reports that provide information about violations to the mentioned standards.

You can download the PIP Auditor and supporting documentation from MetaLink **Note** 782351.1.

Appendix A: XML Structures of Exportable CAVS Definitions and Instances

This chapter provides the XML structures of exportable Composite Application Validation System (CAVS) definitions and instances.

Definition.xml

This is the structure of the Definitions.xml file created by the CAVS definition export feature.

This export feature should be used to migrate definitions between instances running on the same version of Foundation Pack.

Use this structure as a reference if you a receiving a validation error when importing definitions.

Edit this structure to create new definitions for importing to an Oracle Application Integration Architecture (AIA) Foundation Pack instance.

For more information about the definition export and import feature, see Exporting and Importing Definitions and Instances.

```
<DefinitionsList>

<!-- The section below is for one test/simulator definition. This
includes all definition details as well as XPATH conditions set by
the user. For each definition the section below will be repeated -->

<DefinitionsViewRORow>
   <DefinitionId>[Definition ID that was set in the previous
environment. During import, the target system will generate a new ID
for this field]</DefinitionId>
   <Type>[Test|Simulator]</Type>
   <Description>[String. Description of the test or
simulator]</Description>
   <State>[Locked|Unlocked]</State>
   <ServiceType>[Synchronous|Notify|Asynchronous two
way]</ServiceType>
   <UrlEndpoint>[URL]</UrlEndpoint>
   <SoapAction>[String. Valid soap action from the wsdl of the above
URL]</SoapAction>
   <SoapTransportType>[HTTP]</SoapTransportType>
   <MessageRequest>[SOAP Message. Request message along with CAVS
SOAP envelopes]</MessageRequest>
   <MessageResponse>[SOAP Message. Response message along with CAVS
SOAP envelopes]</MessageResponse>
   <Delay>[Integer greater than -1. Only in the case of ServiceType
Asynchronous two way]</Delay>
   <ServiceName>[String]</ServiceName>
```

```
      <ServiceVersion>[String]</ServiceVersion>
      <ProcessName>[String]</ProcessName>
      <PipName>[String]</PipName>
      <AuditedOn>[YYYY-MM-DD HH:MM:SS.M]</AuditedOn>
      <AuditedBy>[oc4jadmin]</AuditedBy>
      <!-- Namespace details from the request/response message. There
   can be more than one occurrence of the section below -->
      <NsXpathsForDefinitionsRO>
         <DefinitionNsXpathsViewRORow>
            <DefinitionId>[Definition ID mentioned
   above]</DefinitionId>
            <NamespaceAlias>[String. namespace alias]</NamespaceAlias>
            <Namespace>[valid namespace URL]</Namespace>
         </DefinitionNsXpathsViewRORow>
      </NsXpathsForDefinitionsRO>
      <!-- XPATH variables and values. There can be more than one
   occurrence of the section below -->
      <XpathsForDefinitionsRO>
         <DefinitionXpathsViewRORow>
            <DefinitionId>[Definition ID mentioned
   above]</DefinitionId>
            <XpathSeqId>[Non negative Integer]</XpathSeqId>
            <Xpath>[XPATH expression]</Xpath>
            <IsNodeText>[0|1.Applicable only for Simulator
   Definitions]</IsNodeText>
            <IsNodeKey>[0|1. Applicable only for Simulator
   Definitions]</IsNodeKey>
            <Condition>[OK|EQ|NE|LT|GE|LE|Not Null]</Condition>
            <IsSystemGenerated>[0|1]</IsSystemGenerated>
         </DefinitionXpathsViewRORow>
      </XpathsForDefinitionsRO>
   </DefinitionsViewRORow>


   <!-- The section below is for one group test definition. This
   includes all definition details as well as references to Test
   definitions that are mentioned above. For each such group definition
   the section below will be repeated -->

   <GroupDefinitions>
      <!-- There can be more than one occurrences of the section below
   -->
   <GroupDefinitionsViewRORow>
      <GroupDefinitionId>[Group Definition ID that was set in the
   previous environment. During import the target system will generate
   a new ID for this field]</GroupDefinitionId>
      <Description>[String]</Description>
      <ProcessName>[String]</ProcessName>
      <PipName>[String]</PipName>
      <GDDefinitionsViewRO>
         <!-- There can be more than one occurrences of the section
   below -->
         <GDDefinitionsViewRORow>
            <GroupDefinitionId>[Group Definition ID set
   above]</GroupDefinitionId>
            <SeqId>[Non negative Integer]</SeqId>
            <DefinitionId>[One of the Definition ID set in the
```

Copyright © 2009, Oracle. All rights reserved.

```
DefinitionsViewRORow section]</DefinitionId>
         <DefinitionSeqId>[Non negative Integer]</DefinitionSeqId>
         <ServiceType>[Synchronous|Notify|Asynchronous two
way]</ServiceType>
         <SoapTransportType>[HTTP]</SoapTransportType>
      </GDDefinitionsViewRORow>
   </GDDefinitionsViewRO>
</GroupDefinitionsViewRORow>
</GroupDefinitions>

</DefinitionsList>
```

Instance.xml

This is the structure of the Instance.xml file created by the CAVS instance export feature.

This export feature can be used to export a test or group instance in XML format that can be used with XML reporting tools to generate reports of test executions.

For more information about the instance export feature, see Exporting and Importing Definitions and Instances.

```
<InstancesList><?xml version = '1.0' encoding = 'UTF-8'?>
<InstancesViewRORow>
<!-- There would be more occurrences of this if more instances are
exported→
   <InstanceId>[Instance ID that was assigned by the environment in
which the instance was run]</InstanceId>
   <Type>[Test|Simulator|Group</Type>
   <Status>[Status of the instances being exported] </Status>
   <StartedOn>[Date and time at which the instance
started]</StartedOn>
   <EndedOn>[Date and time at which the instance ended]</EndedOn>
   <IsStaled>[0|1]</IsStaled>
   <DefinitionId>[Definition ID of the definition that generated the
instance]</DefinitionId>
   <Description>[Description of the definition ID that generated the
instance]</Description>
   <ServiceType>Synchronous|Asynchronous two-way|Asynchronous
(notify)</ServiceType>
   <SoapAction>[String. Valid SOAP action for the WSDL defined for
the definition ID]</SoapAction>
   <SoapTransportType>HTTP</SoapTransportType>
   <MessageRequest>actual request message</MessageRequest>
   <MessageResponse>actual response message</MessageResponse>
   <DefinitionsViewRO>
      <DefinitionsViewRORow>
         <DefinitionId>[Definition ID mentioned
above]</DefinitionId>
         <Type>[Type mentioned above] </Type>
         <Description>[Description mentioned above]</Description>
         <State>[Locked|Unlocked]</State>
         <ServiceType>[Service Type mentioned above] </ServiceType>
         <SoapAction>[SOAP Action mentioned above] </SoapAction>
         <SoapTransportType>HTTP</SoapTransportType>
         <MessageRequest>[Request message defined in the
corresponding Test or Simulator definition]</MessageRequest>
         <MessageResponse>[Response message defined in the
corresponding Test or Simulator definition]</MessageResponse>
         <AuditedOn>[YYYY-MM-DD HH:MM:SS.M]</AuditedOn>
         <AuditedBy>[oc4jadmin]</AuditedBy>
      </DefinitionsViewRORow>
   </DefinitionsViewRO>
   <InstanceXpathsViewRO>
      <InstanceXpathsViewRORow>
         <InstanceId>[Instance ID assigned to the
instance}</InstanceId>
         <XpathSeqId>[Non-negative integer] </XpathSeqId>
         <Status>[Status of the instance] </Status>
         <Xpath>/soap:Envelope</Xpath>
         <IsNodeKey>[0|1. Applicable only for Simulator
Definitions]</IsNodeKey>
         <Condition>[OK|EQ|NE|LT|GE|LE|Not Null]</Condition>
      </InstanceXpathsViewRORow>
   </InstanceXpathsViewRO>
   <InstanceNsXpathsViewRO>
      <InstanceNsXpathsViewRORow>
         <InstanceId>[Instance ID assigned to the instance]
</InstanceId>
         <NamespaceAlias>[String]</NamespaceAlias>
         <Namespace>[Valid namespace URL]</Namespace>
      </InstanceNsXpathsViewRORow>
   </InstanceNsXpathsViewRO>
</InstancesViewRORow></InstancesList>
```

Copyright © 2009, Oracle. All rights reserved.

Index

Copyright © 2009, Oracle. All rights reserved.

Copyright © 2009, Oracle. All rights reserved.