Oracle® Device and Drug Adverse Event Data Integration Pack for Siebel Adverse Event Complaint Management and Oracle Argus Safety

Implementation Guide

Release 11.1

E26803-01

November 2011

ORACLE®

Oracle Device and Drug Adverse Event Data Integration Pack for Siebel Adverse Event Complaint Management and Oracle Argus Safety Implementation Guide, Release 11.1

E26803-01

Primary Author: Supriya Shewale

Contents

Preface

Welcome to the Implementation guide for Oracle Device and Drug Adverse Event Data Integration Pack for Siebel Adverse Events and Complaints Management (AECM) and Oracle Argus Safety 11.1.

Audience

This document is intended for all Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP users.

Oracle AIA Guides

The following Oracle Application Integration Architecture (AIA) guides and resources provide more information on AIA Foundation Pack 11.1.1.5 on which this integration is built:

- Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.5)
- Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.5)
- Oracle Fusion Middleware Getting Started and Demo Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.5)
- Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.5)
- Oracle Fusion Middleware Installation and Upgrade Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.5)
- Oracle Fusion Middleware Reference Process Models User's Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.5)
- Oracle Fusion Middleware Product to Guide Index for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.5)
- Oracle Fusion Middleware Migration Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.5)

Additional Resources

The following resources are also available:

Resource	Location
Oracle Device and Drug Adverse Event Data Integration Pack for Siebel Adverse Event Complaint Management and Oracle Argus Safety Installation Guide	Oracle Technology Network: http://www.oracle.com/technology/
Known Issues and Workarounds	Oracle Technology Network: http://www.oracle.com/technology/
Release Notes	Oracle Technology Network: http://www.oracle.com/technology/
Documentation updates	Oracle Technology Network: http://www.oracle.com/technology/

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Part I

Understanding the Delivered Integration

Part I contains the following chapters:

- Chapter 1, "Understanding the Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP"
- Chapter 2, "Understanding the Synchronization of Siebel AECM Product Issue with Oracle Argus Safety System Case"

1

Understanding the Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP

This chapter provides an overview of the Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP. This chapter includes:

- Section 1.2, "Key Benefits of the PIP"
- Section 1.3, "Participating Applications Overview"
- Section 1.4, "Business Process Flow"
- Section 1.5, "Solution Assumptions and Constraints"

1.1 Overview

Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP automates the process of reporting adverse events for a drug or a medical device to the safety system, when Product Issues are logged in the product complaint management system.

Adverse events are entered as Product Issues in Siebel AECM. The AECM user can choose to send the Product Issue to the safety system. Siebel AECM writes messages to a queue. Product issue type indicates whether an issue is a serious adverse reaction to a drug or a medical device. The AIA layer transforms the Siebel AECM XML message into the ReportDrugSafetyReportEBM and creates an E2B+ file.

Oracle Argus Safety is an adverse event reporting and management system with an E2B+ interface. It enables you to import the files in E2B+ format and you can evaluate the files in the Oracle Argus Safety staging area. The adverse event case is then either accepted in Oracle Argus Safety as a new case or the case is appended as a follow-up to an existing case in Oracle Argus Safety. Oracle Argus Safety sends an acknowledgement message with the Safety case number when the case is created.

The Siebel AECM user may choose to send follow-ups and receive an acknowledgement from Oracle Argus Safety regarding the case. The Product Issue in Siebel AECM is updated with the case number and the date the follow-up was accepted into the safety system.

If the Oracle Argus Safety user rejects the case, a message regarding the case rejection is added to the Inbox of each product owner of the Product Issue. This acknowledgement message includes the reason for rejecting the case.

1.2 Key Benefits of the PIP

The Siebel AECM to Oracle Argus Safety integration helps eliminating error-prone and time-consuming manual processes by immediately delivering adverse events data to the Oracle Argus Safety application.

The transfer of adverse event cases from Siebel AECM to the Oracle Argus Safety is automated. This automation increases productivity and reduces the amount of reconciliation needed between the two systems. Thus, this integration aids in meeting the strict reporting timelines for adverse events.

The resulting Oracle Argus Safety unique identifier assigned to the appropriate Siebel AECM Product Issue record enhances traceability and helps to maintain the audit trail.

1.3 Participating Applications Overview

This section provides an overview of the participating applications.

1.3.1 Siebel Adverse Event and Complaint Management System

Siebel Adverse Event and Complaint Management system is specifically designed for medical, biotechnology, and pharmaceutical companies to improve regulatory compliance, product quality, and business performance. Siebel AECM combines a number of sub-processes that enable life sciences companies to create an integrated, traceable, and efficient adverse event and complaint management process with maximum compliance and operational efficiency.

Siebel AECM can capture and escalate complaints for investigation and regulatory reporting while maintaining a complete audit trail. This approach helps in complying with stringent regulations, thereby avoiding potential penalties for noncompliance. It enables employees to seamlessly collect adverse event and complaint information, identify the cause of the problem, and effectively assess whether the event has to be reported to the regulatory agency.

1.3.2 Oracle Argus Safety

Oracle Argus Safety is an advanced and comprehensive adverse events management system that helps life sciences companies enable regulatory compliance, drive product stewardship, and integrate safety and risk management into one comprehensive platform. Oracle Argus Safety is industry-proven and accepted, having been used for more than a decade at leading pharmaceutical, biotech, CRO (Clinical Research organization), and medical device manufacturers. It also facilitates internal company safety surveillance by analyzing the overall safety profile of both investigational compounds and marketed products.

Oracle's proactive approach to monitoring global guidance enables consistent regulatory compliance. Oracle Argus Safety supports electronic communication with trading partners and CROs, providing visibility into compliance across a company's global licensing partnerships.

Oracle Argus Safety also supports a company's end-to-end pharmacovigilance program by providing a simple and efficient way to comply with international and domestic regulatory safety reporting requirements from clinical trials through post-marketing surveillance. By integrating serious adverse events from clinical trials, companies have the opportunity to manage efficacy earlier in the drug lifecycle and can potentially address public health concerns sooner. It provides the most comprehensive global adverse events case data management and regulatory reporting in the pharmaceutical industry.

1.4 Business Process Flow

All adverse event reports are promptly investigated by designated global safety specialists within strict regulatory reporting timeframes. Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP automates the transfer of adverse event cases from Siebel AECM to the Oracle Argus Safety system.

The Siebel AECM user logs a Service Request (SR) into Siebel AECM. An issue is created in Siebel AECM against the appropriate Product. If the issue is an adverse event, Siebel AECM user may send the case to the safety system. The case is imported into Oracle Argus Safety by the Argus Interchange functionality. You can configure Oracle Argus Safety to automatically accept all new cases. If not configured for auto-acceptance, the safety user checks the case to find out if it is a duplicate case and then accepts the case as new, or accepts the case as a follow-up to an existing case or rejects the case. An acknowledgement is returned from Oracle Argus Safety with the Safety Case Number and the date the case was accepted or a reason why it was rejected.

Figure 1–1 High Level Business Process Task Flow

Refer to the Business Process Model diagram Part 1, and 2 and the process description given below:

Figure 1–2 Business Process Model Part – 1 (Continued on next page...)

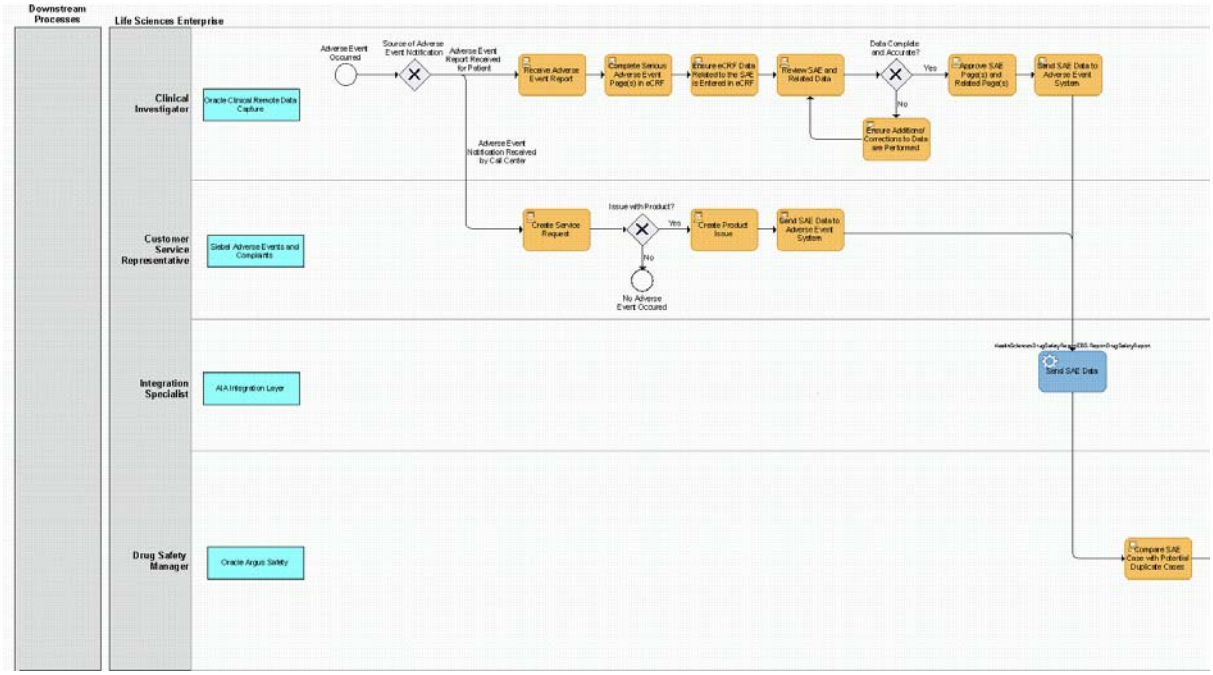
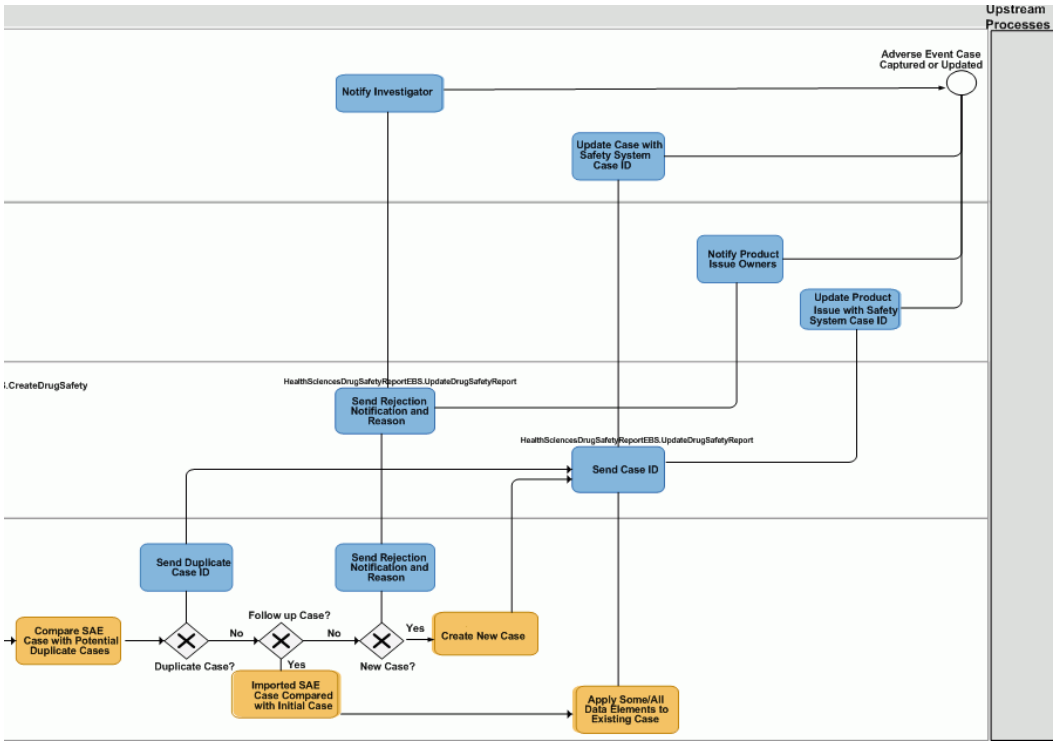


Figure 1–3 Business Process Model Part - 2



1.4.1 Description

The process flow description:

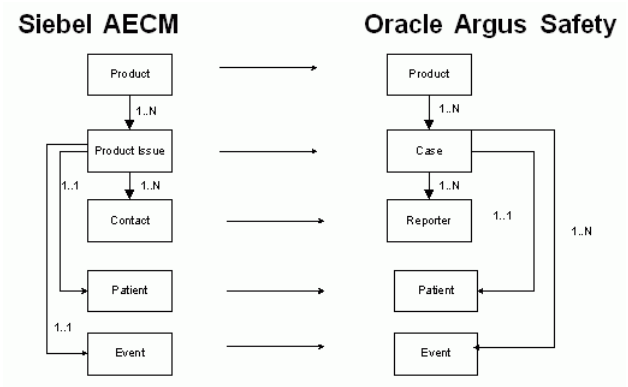
- An SR is entered in Siebel AECM system when a customer calls the Call Center. The Siebel AECM user creates a Product Issue for the product complaint.
- If the user feels the Product Issue should be reported to the safety group, the reportable field is set to **Yes** or **Potential**. This enables the **Send to Safety** button. When the user clicks this button, an XML message is written by Siebel to a JMS Queue on the SOA (Service-oriented Architecture) server.
- The Siebel Application Business Message (ABM) is transformed into the ReportDrugSafetyReportEBM.
- If Oracle Argus Interchange server is configured to auto accept incoming E2B cases, the file is immediately imported and a new case is created in Oracle Argus Safety. If auto accept incoming E2B cases is not configured, the file shows up as a Pending E2B Report in Oracle Argus Safety. The Oracle Argus Safety user can search to see if a duplicate case exists and can choose to accept the report as a new case or to accept the report as a follow-up to an existing case or reject the case.
- Oracle Argus Safety then writes an acknowledgement message. The case number is parsed out of the acknowledgement message and sent to Siebel AECM, where the Product Issue is updated with the safety system case number and the date the last information was received from Oracle Argus Safety.
- Follow-up information received by the call center may be sent to Oracle Argus Safety. The Oracle Argus Safety user can choose to append the follow-up information to the existing case or to a different case in Oracle Argus Safety. If the follow-up information is added to another case, the Safety Case ID is updated for the Product Issue in Siebel AECM to reflect the new case it was appended to.

- If the Oracle Argus Safety user rejects the case or the follow-up, a message is added to the Inbox of each product owner of the Product Issue. This message states that the Product Issue or the follow-up was rejected as a case in Oracle Argus Safety and includes the reason specified in the Oracle Argus Safety acknowledgement.
- The call center can choose to nullify the case or reject the case. The Oracle Argus Safety user has the option to manually delete the case in Oracle Argus Safety.

1.4.2 Logical Data Model Diagram

This section contains the Logical Data Model diagram.

Figure 1–4 Logical Model Mapping Siebel AECM and Oracle Argus Safety



1.5 Solution Assumptions and Constraints

These are the assumptions and constraints for this PIP:

- You must install the Oracle Argus Interchange module.
- There is a one-to-one mapping between a Product Issue in Siebel AECM and a Case in Oracle Argus Safety at any given point and time. You can link only one Product Issue to a particular case in Oracle Argus Safety at a given point in time; however, you can link the Product Issue to a different case at a different point in time.
- The design is based on the assumption that the Oracle Argus Safety user processes the Pending E2B Reports in the order that they were received. If the reports are not processed in the same order they were received, newer reports might be processed first and then the original report might be rejected as a duplicate.

2

Understanding the Synchronization of Siebel AECM Product Issue with Oracle Argus Safety System Case

This chapter provides an overview of synchronization of Siebel AECM Product Issue with Oracle Argus Safety System Case and discusses:

- Section 2.2, "Functional Process Flow"
- Section 2.3, "Participating Siebel AECM Interfaces"
- Section 2.4, "Participating Oracle Argus Safety Interfaces"
- Section 2.5, "Industry AIA Components"
- Section 2.6, "Integration Services"

2.1 Overview

The synchronization automates the transfer of adverse event cases from Siebel AECM to the Oracle Argus Safety system. The synchronization feature is described in the following section:

2.1.1 Description

Siebel AECM user enters an SR for the product complaint in Siebel AECM system when a customer calls the Call Center. The Siebel AECM user creates a Product Issue for the product complaint. The Siebel AECM user analyzes the Product Issue.

If the adverse event is for a medical device or combination drug/medical device product, Siebel AECM user must select Event Type value containing the string device in the Language Independent Code (LIC).

For more information about how to set the Language Independent Code (LIC) in Siebel AECM, see *Configuring Siebel Business Applications, Version 8.1, Rev. B*," Localizing Siebel Business Applications - Localizing a Multilingual List of Values".

The Reportable field in the **Product Issue List** indicates whether the Product Issue is for a reportable adverse event or a potential reportable adverse event. The **Send to Safety** button is used to send the Product Issue information to the Oracle Argus Safety. Enabling and disabling of **Send to Safety** button is controlled by the values selected in **Reportable** field in the **Product Issue List**. **Send to Safety** button is enabled when the **Reportable** field is set to **Yes** or **Potential**.

When the **Send to Safety** button is pressed, an XML message is written by Siebel AECM to a JMS Queue on the SOA server. The Siebel ABM is transformed into the ReportDrugSafetyReportEBM.

There are two configurations on integration layer, which give control on sending additional non E2B fields from Siebel AECM:

- **SendNonE2B:** (=true/false) property in AIAConfigurationProperties.xml, controls populating of non E2B, non device fields in the ReportDrugSafetyReportEBM. The following non E2B, non device fields of ReportDrugSafetyReportEBM are controlled by this property:
  - RegulatoryAuthoritySubmissionIndicationCode
  - DrugSafetyReportPatient/DrugSafetyReportReaction/SeverityCode

The default value for SendNonE2B property is false. If the SendNonE2B property value is set to true, **Reported FDA** and **Event Severity** fields in Siebel AECM are populated in the ReportDrugSafetyReportEBM.

- **SendDeviceFields:** (=true/false) property in AIAConfigurationProperties.xml, and Siebel AECM **Event Type** field value containing string **Device**, controls populating of device fields in the ReportDrugSafetyReportEBM. The following device fields of ReportDrugSafetyReportEBM are controlled by this property:
  - Product Issue Product Model#
  - Product Issue Product Catalog#
  - Product Issue Product Serial#
  - Product Issue Product Device Operator
  - Product Issue Product Mfg Date
  - Product Issue Product Expiration Date
  - Product Issue Product Implant Date
  - Product Issue Product Explant Date
  - Product Issue Investigation Tab Evaluated by Mfg
  - Product Issue Product Device Available
  - Indicator of whether the device was labeled as Single Use and was Reprocessed and Reused on a patient
  - Product Issue MDV Tab Usage of Device
  - Product Issue Product Return Date
  - Product Issue MDV Tab Mfg Narrative
  - Product Issue Investigation Tab Remedial Action Type
  - Product Issue Investigation Tab Remedial Action Other
  - Product Issue Product Reprocessor
  - Product Issue Product Reprocessor Address
  - Product Issue Product Reprocessor City
  - Product Issue Product Reprocessor State
  - Product Issue Product Reprocessor Country
  - Product Issue Product Reprocessor Zip

- Product Issue Investigation Tab Evaluation Codes Method Codes
- Product Issue Investigation Tab Evaluation Codes Result Codes
- Product Issue Investigation Tab Evaluation Codes Conclusion Codes
- Product Issues MDV Tab Patient Code or Device Code

When Siebel AECM writes the message to the JMS queue, the Product Issue is updated with the **Pending** status. The adverse event data is sent to Oracle Argus Safety. The integration has its own DTD specification for Oracle Argus Safety. This includes the addition of some non-E2B fields in order to send device information and additional information that can be shared between Siebel AECM and Oracle Argus Safety. The integration writes an XML file that conforms to the DTD defined in the file - **ich-icsr-v2.1-FDA-PIP.dtd**

If Oracle Argus Interchange server is configured to auto accept the cases, the file is immediately imported and a new case is created in Argus Safety. If auto accept is not configured, the file is shown as a Pending E2B Report in Oracle Argus Safety; sorted by the Transmission date. The transmission date is the date when Siebel AECM sent the message to Oracle Argus Safety. The Oracle Argus Safety user can perform a search to see if a duplicate case exists and can choose to accept the report as a new case, or accept the report as a follow-up to an existing case or reject the case. Oracle Argus Safety writes an acknowledgement message. The case number is parsed out of an acknowledgement message and sent back to Siebel AECM, where the Product Issue is updated with the safety system case number, and the date the case was accepted or rejected in Oracle Argus Safety. The Product Issue status is updated to **Submitted** (or any other equivalent status value defined by you).

When the case ID is received from Argus Safety, follow-up information received by the call center can be sent in the same manner as for the new cases. Until the case ID is received from Argus Safety, any additional information sent by the call center is considered as a Potential New case. The Oracle Argus Safety user can choose to append the follow-up information to a different case in Argus. When this happens the Safety Case ID is updated for the Product Issue in AECM to reflect the new case it is appended to.

If the Oracle Argus Safety user rejects the case or the follow-up, a message is added to the Inbox of each product owner of the Product Issue and the Product Issue status is changed to **Rejected** (or any other equivalent status value defined by you). This message states that the Product Issue is rejected as a case in Oracle Argus Safety and includes the reason specified in the Oracle Argus Safety acknowledgement.

The call center can decide to void the case. This information is sent over in the same manner and the Oracle Argus Safety user has the option to accept or reject the nullification. The case is not deleted in Oracle Argus Safety even if the user accepts the nullification. The Oracle Argus Safety user must decide whether to delete the case in Oracle Argus Safety and then manually delete it.

For more information on entering Product Issues and sending them to the Oracle Argus Safety system, see Chapter 33: Setting Up and Configuring Safety System Integration, of the *Siebel Life Sciences Guide, Version 8.1.1.6*.

For more information about the Oracle Argus Interchange functionality refer to the Argus Interchange User's Guide, Release 6.0.1.

For more information related to Pending E2B Reports, see the Argus Safety User's Guide, Release 6.0.1.

2.2 Functional Process Flow

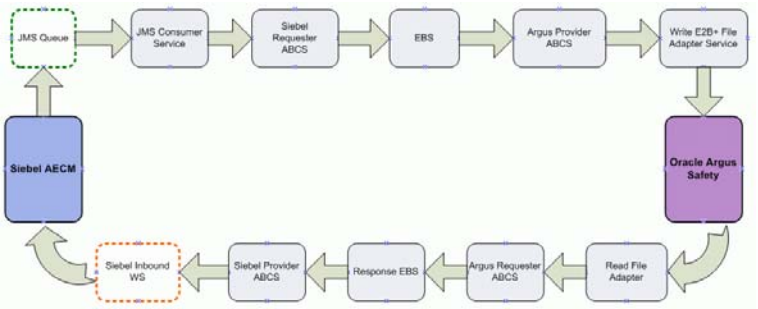
This section describes the functional process flow and includes the flow diagram.

2.2.1 Flow Diagram

The following sequence diagram shows:

- The flow of information through AIA components for the ReportDrugSafetyReport flow from Siebel AECM to Oracle Argus Safety
- The flow of information through AIA components for the ReportDrugSafetyReportResponse flow from Oracle Argus Safety to Siebel AECM.

Figure 2–1 Functional Process Flow



2.2.2 Description

ReportDrugSafetyReport Flow:

Siebel AECM dispatches Adverse Event Product Issue information to Oracle Argus Safety. The three functional flows listed below are implemented with the two technical flows namely the Report Drug Safety Report Flow and Report Drug Safety Report Response Flow. Dispatch of adverse event Product Issue from Siebel AECM to Oracle Argus Safety covers the following three functional flows:

- **Initial**: Dispatching new Adverse Event Product Issue information from Siebel AECM to Oracle Argus Safety system.
- **Follow-up**: Dispatching updated Product Issue information from Siebel AECM to Oracle Argus Safety system.
- **Void**: Dispatching an existing Product Issue that is voided from Siebel AECM to Oracle Argus Safety system.

Siebel AECM sends an XML message called Application Business Message (ABM) to a JMS queue on integration SOA server, when the Siebel AECM user clicks the Send to Safety button on the Siebel AECM Product Issue user interface. A JMS consumer, a mediator component on SOA server, reads the ABM from the JMS queue and routes it to the Siebel Requester Application Business Connector Services (ReportDrugSafetyReportSEBLReqABCSImpl). The Siebel Requester ABCS, a BPEL process on SOA server, transforms the Siebel ABM to the ReportDrugSafetyReport EBM. The Siebel Requester ABCS invokes the ReportDrugSafetyReport operation of HealthSciencesDrugSafetyReportEBS with ReportDrugSafetyReportEBM as the input message.

HealthSciencesDrugSafetyReportEBS, mediator component, routes the EBM to the Argus Provider Application Business Connector Service.

ReportDrugSafetyReportArgusProvABCSimpl, the Argus Provider ABCS converts the EBM to the format specified in the ich-icsr-v2.1-FDA-PIP.dtd file. This is an extension of E2B that is provided as part of this integration. The Argus Provider ABCS invokes a Write File Adapter service to write the XML file to the in directory of Oracle Argus Safety Interchange server. The Write File Adapter is a BPEL process which invokes the SOA File Adapter to write the XML files to the Oracle Argus Safety Interchange server. The file is then imported into Oracle Argus Safety. If Oracle Argus Safety is configured to auto-accept pending E2B reports, the case is created in Oracle Argus Safety. If not, the new case, the follow-up, or the nullification appears in the Pending E2B Reports in Oracle Argus Safety.

ReportDrugSafetyReportResponse Flow:

Oracle Argus Safety dispatches an acknowledgement to initial, follow up, and nullified reports. This information flows from Oracle Argus Safety to Siebel AECM. The acknowledgement files are generated for one of the following main reasons:

- Acknowledgement file is auto-generated by Oracle Argus Safety when E2B import validation criteria are not met.
- Acknowledgement file is generated when a safety user accepts or rejects a new case or follow-up in Oracle Argus Safety web application.

When a case is accepted or rejected in Oracle Argus Safety, an acknowledgement message is written to a directory on the Oracle Argus Safety Interchange server. A File Adapter service reads the acknowledgement file and passes it to the Requester ABCS, ReportDrugSafetyReportResponseArgusReqImpl. The File adapter moves the file to the archive directory after processing it. The filename has the conversation ID and a timestamp appended to it. The Requester ABCS transforms the acknowledgement message to the ReportDrugSafetyReportResponseEBM. The message is sent to the HealthSciencesDrugSafetyReportResponse EBS which routes it to the Provider ABCS, ReportDrugSafetyReportResponseSEBLProvABCSImpl.

The Provider ABCS updates the Product Issue in Siebel AECM with the Argus case ID, status (whether accepted or rejected) and the date on which safety system user accepted or rejected the case or the follow-up.

If the status is rejected, the Provider ABCS calls the Siebel web service to add a message to the Inbox of each owner of the Product Issue.

2.3 Participating Siebel AECM Interfaces

These Siebel AECM artifacts are used by this integration:

Siebel AECM Outbound and Inbound Web Services

- **Outbound** - LSMedicalToSafetyIntegProductIssueInterfaceTarget
- **Inbound** - LS_Medical_Product_Issue_Create_Inbox_Item_Inbound
- **Inbound** - LS_Medical_Update_Product_Issue_Inbound

2.4 Participating Oracle Argus Safety Interfaces

These Oracle Argus Safety artifacts are used by this integration:

Oracle Argus Safety Outbound and Inbound Web Services

- **Inbound** - ich-icsr-v2.1-FDA-PIP.dtd (This DTD file defines the E2B+ XML message format for this integration.)

- **Outbound** - FDA-icsrack-v1.1.dtd (This DTD file defines the Safety acknowledgement XML message format for this integration.)

2.5 Industry AIA Components

The integration flow uses the following components:

- DrugSafetyReportEBO
- ReportDrugSafetyReportEBM
- HealthSciencesDrugSafetyReportEBS
- HealthSciencesDrugSafetyReportResponseEBS

This integration uses the DrugSafetyReportEBO.

This integration uses ReportDrugSafetyReportEBM and the response message that is sent back is ReportDrugSafetyreportResponse

The industry EBO and EBM XML Schema Definition (XSD) files can be located by EBO within the $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/EBO/ parent folder.

The industry EBS Web Services Description Language (WSDL) files can be located by EBO within the $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/HealthSciences/EBO/ parent folder.

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see Oracle Application Integration Architecture – Foundation Pack: Development Guide, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository".

EBOs can be extended, for instance, to add new data elements. These extensions are protected, and will remain intact after a patch or an upgrade.

For more information, see Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide, "Extensibility for AIA Artifacts."

2.6 Integration Services

These are the services delivered with this integration:

- **SEBLCLINDrugSafetyReportJMSConsumer** - When the Siebel AECM user clicks the **Send to Safety** button, an XML message is written to a JMS queue, DrugSafetyReportSEBLQ, on the SOA server. This consumer service subscribes to that queue and read the messages and route them to the Siebel RequesterABCS - ReportDrugSafetyReportSEBLReqABCSImpl.
- **HealthSciencesDrugSafetyReportEBS** - This service routes the ReportDrugSafetyReport EBM to the ReportDrugSafetyReportArgusProvABCSImpl.
- **HealthSciencesDrugSafetyReportResponseEBS** - This service routes the ReportDrugSafetyReportResponse EBM to the ReportDrugSafetyReportResponseSEBLProvABCSImpl.
- **ReportDrugSafetyReportSEBLReqABCSImpl** - The Siebel ABM is received and transformed into the ReportDrugSafetyReportEBM.

- **ReportDrugSafetyReportResponseSEBLProvABCSImpl** - This service receives a ReportDrugSafetyReportEBM and calls the Siebel Web service LS_Medical_Update_Product_Issue_Inbound to update the Product Issue status, Safety System Case Number and Receipt date for the Product Issue. If the response code is rejected, it calls the Siebel web service LS_Medical_Product_Issue_Create_Inbox_Item_Inbound to add a message to the Inbox of each of the Product Issue owners.
- **ReportDrugSafetyReportArgusProvABCSImpl** - This BPEL service uses a file adapter component to write an E2B+ file in Oracle Argus Safety specified directory location. The E2B+ format is defined for this integration in the DTD file ich-icsr-v2.1-FDA-PIP.dtd.
- **ReportDrugSafetyReportResponseArgusReqABCSImpl** - This ABCS is invoked by a file adapter service that polls the Oracle Argus Safety directory to wait for an acknowledgement file to be written when the user accepts or rejects the Pending E2B report, or if the import fails. The case ID, safety received date, and rejection message are parsed from the error message in the acknowledgement message and sent back to Siebel AECM using the ReportDrugSafetyReportResponseEBM.
- **ReportDrugSafetyReportWriteE2BFileAdapter** - This File Adapter service is used to write the XML file to the Oracle Argus Safety Interchange server.
- **ReportDrugSafetyReportReadAckFileAdapter** - This service reads the acknowledgement message from the Oracle Argus Safety system directory, passes it to ReportDrugSafetyReportResponseArgusReqABCSImpl, and moves it to the **archive** directory underneath the out directory.

Part II

Implementing the Delivered Integration

Part II contains the following chapter:

- Chapter 3, "Implementing the Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP"

3

Implementing the Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP

This chapter discusses the steps you must complete to integrate data in Siebel AECM and Oracle Argus Safety PIP. It covers:

3.1 Prerequisites

These are the prerequisites for this PIP:

You can customize the list of values for the fields in Siebel AECM and Oracle Argus Safety. DVMs are created to map the value in Siebel AECM that matches the value expected by the Oracle Argus Safety import feature. You must review and update the DVMs to match the values in your systems. For more information, refer to the section Working with Domain Value Maps.

If you have added any fields to Siebel AECM or to Oracle Argus Safety that you would like to send from Siebel AECM to Oracle Argus Safety, you must follow the steps that are available in the section Extending the Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP.

Review and set the configuration settings for the integration appropriately for your business process. Refer to the section Configuration Properties for the Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP for setting up the configuration properties.

3.2 Data Requirements

This section discusses the data requirements. Data requirements indicate the mandatory data that you must provide to make the integration flows successful.

When a Product Issue is entered in Siebel AECM you must enter the data in the following fields before clicking the **Send to Safety** button. You will receive an error message from Siebel AECM, if the data is not entered in these fields:

- Event Type
- Report Type
- Event Description
- Awareness Date
- Country of Incidence
- At least one suspect product must be entered in the **Product** tab of the **Product Issues** screen.

  If Study is entered for the **Report Type**, the following must be entered in the **Patient** tab:

- Patient ID

  OR any of the following:

- Date of Birth
- Gender
- Age

3.3 Setting up Participating Applications

This section describes how to set up Oracle Argus Safety and Siebel AECM to utilize the integration.

3.3.1 Setting up Oracle Argus Safety

This section describes the tasks and general procedures that you must complete to set up Oracle Argus Safety.

The Argus Interchange feature requires some information to be entered in the Argus Console for the Frequency codelist. Therefore, before using the integration the Argus administrator must go to the Argus Console and navigate to the Frequency codelist.

For each value that exists in the codelist for **Frequency**, enter the following three fields:

- **Number of doses per day** - Enter the number of doses to be taken each day.
- **Dose** - Enter amount of dose to be taken each time.
- **Every N days** (where you enter N) - Enter interval of dosage frequency (such as Every 2 Hours, Every 5 Days, Every 6 Weeks.

3.3.2 Setting up Siebel AECM

This section describes the tasks and general procedures that you must complete to set up Siebel AECM.

If Siebel AECM is used for adverse events with Medical Device or combination Drug/Medical Device products, the Language Independent Code (LIC) in Siebel AECM Tools for the **Event Type** pick list must contain the string **device**.

Ensure that your Siebel AECM administrator sets the LIC for the pick list options to contain the string **device.**

The **Frequency** field in Siebel AECM has a drop down list of values. The LICs must be defined in a particular format in order for the frequency to go over correctly to Oracle Argus Safety. The display values can be anything.

The LIC value must be in one of the following formats. Note that these formats are case sensitive:

- Once a <unit of time>
- Twice a <unit of time>
- Thrice a <unit of time>
- Once Every <n> <unit of time>
- Twice Every <n> <unit of time>
- Thrice Every <n> <unit of time>
- <n> Times a <unit of time>
- <n> Times Every <n> <unit of time>

Where <n> is any numeric value and <unit of time> is one of the following:

- Minute or Minutes
- Hour or Hours
- Day or Days
- Week or Weeks
- Month or Months
- Year or Years

  Examples are: Twice a Day, Once Every 2 Weeks, 4 Times a Day, 3 Times Every 2 Months

For more information about how to set the Language Independent Code (LIC) in Siebel AECM, see *Configuring Siebel Business Applications, Version 8.1, Rev. B*, "Localizing Siebel Business Applications - Localizing a Multilingual List of Values".

If you are using Siebel Clinical and have defined Protocols in your application, you must enter the Study ID from Oracle Argus Safety for that study. Follow these steps to associate the Product Issue to the Study ID:

1. Navigate to **Administration Clinical** and select **Protocols List**.
2. Drill down to the **Protocol** and click **More** tab.
3. Scroll down to the **Integrations** section of the screen.
4. Enter the Oracle Argus Safety Study ID in the **Safety System Study** field.

5. This ensures that Oracle Argus Safety connects the case that is created for the Product Issue to the correct study.

For more information about the integration processes setup and configuring Siebel Adverse Event and Complaint Management system for integration with Oracle Argus Safety, see the *Siebel Life Sciences Guide*, Version 8.1.1.6.

3.4 Siebel AECM to Oracle Argus Safety Field Mapping

This section contains a table that describes the field mapping between Siebel AECM to Oracle Argus Safety and lists all the data that is sent from Siebel AECM to Oracle Argus Safety.

Siebel AECM	Oracle Argus Safety
Product Issues Screen Fields	
Country specified in the Address chosen for the Primary Contact of the Product Issue	Case Form / General/ Reporter Information/Country
Product Issue Country of Incidence	Case Form / General/General Information/Country of Incidence
Product Issue Report Type	Case Form/General Information/Report Type
If any of the seriousness checkboxes are checked this will be 'Y'	Case Form/ Analysis/ Case Serious
Death checkbox under Reported Outcome of Event	Case Form / Analysis / Case Serious / Notes
Life Threatening checkbox under Reported Outcome of Event	Case Form / Analysis / Case Serious / Notes
Hospitalization checkbox under Reported Outcome of Event	Case Form / Analysis / Case Serious / Notes
Disability or Permanent Damage checkbox under Reported Outcome of Event	Case Form / Analysis / Case Serious / Notes
Cogenital Anomaly or Birth Defect checkbox under Reported Outcome of Event	Case Form / Analysis / Case Serious / Notes
If value filled in for Other Serious under Reported Outcome of Event	Case Form / Analysis / Case Serious / Notes
Awareness Date	Case Form / General/General Information/Initial Receipt Date
Last Followup Date	Case Form/General/General Information/Followups/Follow Up Date
Reported FDA	Case Form/Reporter Information/Report Sent to Regulatory Authority by Reporter?(Note this will be set for first reporter only)
Product Issue #	Case Form/Additional Info/References/ID
Related Product Issue #	Case Form/ Additional Info/ References/ID
Reporter Qualification	Case Form/General /Reporter Information/Reporter Type
Product Issue Contact Division	Case Form/ General Information/ Reporter Information/ Department

Siebel AECM	Oracle Argus Safety
Product Issue Contact Organization	Case Form/ General Information/ Reporter Information/ Institution
Product Issue Contact First Name	Case Form/ General Information/ Reporter Information/ First Name
Product Issue Contact Last Name	Case Form/ General Information/ Reporter Information/ Last Name
Product Issue Contact Location Address	Case Form/ General Information/ Reporter Information/ Address
Product Issue Contact Location City	Case Form/ General Information/ Reporter Information/ City
Product Issue Contact Location State	Case Form/ General Information/ Reporter Information/ State/Province
Product Issue Contact Location Country	Case Form/ General Information/ Reporter Information/ Country
Product Issue Contact Location Zip Code	Case Form/ General Information/ Reporter Information/ Postal Code
SafetySystemStudyID entered for the Protocol	Case Form/General/Study Information/Study ID
Product Issues Patient Tab Patient ID	Case Form / Patient / Patient Information / Pat. ID
Product Issues Patient Tab Date of Birth	Case Form / Patient / Patient Information / Date of Birth
Product Issues Patient Tab Age	Case Form / Patient / Patient Information / Age, Case Form / Patient / Patient Information / Age Units
Product Issues Patient Tab Weight	Case Form / Patient / Patient Information / Weight
Product Issues More Info tab Reported Outcome of Event	Case Form / Events / Event Information / Outcome of Event
More Info tab Event severity	Case Form / Events / Event Information / Event Intensity
Product Issues More Info tab Tests/Data	Case Form / Patient / Lab Data / Lab Data / Notes
Drug Fields	
Product Issues Product Name	Case Form / Products / Product Information / Product Name
Product Issues Product Lot#	Case Form / Products / Dosage Regimens / Batch/Lot #
Product Frequency	Case Form / Products / Dosage Regimens / Frequency
Product Dose/Unit and UOM	Case Form / Products / Dosage Regimens / Dose Case Form / Products / Dosage Regimens / Daily Dosage
Product Route Used	Case Form / Products / Dosage Regimens / Patient Route of Administration
Product Indication	Case Form / Products / Product Indication / Coded Indication

Siebel AECM	Oracle Argus Safety
Device Fields	
Product Issue Product Name	Case Form/Products/Device/Product Name
Product Issue Product Lot#	Case Form/Products/Device Information/Lot #
Product Issue Product Model#	Case Form/Products/Device Information/Model #
Product Issue Product Catalog#	Case Form/Products/Device Information/Catalog #
Product Issue Product Serial#	Case Form/Products/Device Information/Serial #
Product Issue Product Device Operator	Case Form/Products/Device Information/Operator of Device
Product Issue Product Mfg Date	Case Form/Products/Device Information/Date of Mfr
Product Issue Product Expiration Date	Case Form/Products/Device Information/Expiration Date
Product Issue Product Implant Date	Case Form/Products/Device Information/Date Implanted
Product Issue Product Explant Date	Case Form/Products/Device Information/Date Explanted
Product Issue Investigation Tab Evaluated by Mfg	Case Form/Products/Device MedWatch Info/ Device evaluated by mfr?
Product Issue Product Device Available	Case Form/Products/Device Information/Device Available for Evaluation
Yes if Product Issue Product Reprocessed is checked and Labeled Single Use is checked and Usage of Device = "Reuse"	Case Form/Products/Device MedWatch Info/ Is this a Single use device that was Reprocessed and Reused on a patient
Product Issue MDV Tab Usage of Device	Case Form/Products/Device MedWatch Info/Usage of Device
Product Issue Product Return Date	Case Form/Products/Device Information/Returned to Manufacturer on
Product Issue MDV Tab Mfg Narrative	Case Form/Products/Device MedWatch Info/Additional Manufacturer Narrative
Product Issue Investigation Tab Remedial Action Type	Case Form/Products/Device MedWatch Info/Remedial Action Taken
Product Issue Investigation Tab Remedial Action Other	Case Form/Products/Device MedWatch Info/Remedial Action Taken Other
Product Issue Product Reprocessor	Case Form/Products/Device MedWatch Info/Name and Address of Reprocessor
Product Issue Product Reprocessor Address	Case Form/Products/Device MedWatch Info/Name and Address of Reprocessor
Product Issue Product Reprocessor City	Case Form/Products/Device MedWatch Info/Name and Address of Reprocessor
Product Issue Product Reprocessor State	Case Form/Products/Device MedWatch Info/Name and Address of Reprocessor
Product Issue Product Reprocessor Country	Case Form/Products/Device MedWatch Info/Name and Address of Reprocessor

Siebel AECM	Oracle Argus Safety
Product Issue Product Reprocessor Zip	Case Form/Products/Device MedWatch Info/Name and Address of Reprocessor
Product Issue Investigation Tab Evaluation Codes Method Codes	Case Form/Products/Device MedWatch Info/Evaluation Codes first row
Product Issue Investigation Tab Evaluation Codes Result Codes	Case Form/Products/Device MedWatch Info/Evaluation Codes second row
Product Issue Investigation Tab Evaluation Codes Conclusion Codes	Case Form/Products/Device MedWatch Info/Evaluation Codes third row
Patient when sending Patient, Device when sending Device	Case Form/Analysis/MedWatchInfo/FDA Codes/Patient or Device
Product Issues MDV Tab Patient Code or Device Code	Case Form/Analysis/MedWatchInfo/FDA Codes. **Note** that this integration can only populate 3 Patient Codes and 3 Device Codes into Oracle Argus Safety

3.5 Configuration Properties for the Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP

The table below describes the configuration properties.

Service Name	Property Name	Value	Description
ReportDrugSafetyReportSEBLReqABCSImpl	Default.SystemID	SEBLCLIN_01	Default System ID is associated with this service. It is used for searching the DVMs, and XREFS when no other system IDs are found in the system registry.
ReportDrugSafetyReportSEBLReqABCSImpl	SendNonE2B	false (default), true	This field controls sending of Non E2B, non device fields from Siebel AECM to Oracle Argus Safety. This includes **Reported to FDA** and **Event Severity** fields. If it set to false, then Non E2B fields are not sent to Oracle Argus Safety.
ReportDrugSafetyReportSEBLReqABCSImpl	SendDeviceFields	false (default), true	This field controls sending of Device specific fields from Siebel AECM to Oracle Argus Safety.
ReportDrugSafetyReportSEBLReqABCSImpl	ABCSExtension. PreXformABMtoEBM	false (default), true.	Custom extension prior to the execution of transformation of application business message (ABM) to Enterprise Business Message (EBM).

Service Name	Property Name	Value	Description
ReportDrugSafetyReportSEBLReqABCSImpl	ABCSExtension. PreInvokeEBS	false (default), true	Custom extension prior to the invocation of the enterprise business service (EBS).
ReportDrugSafetyReportSEBLReqABCSImpl	ABCSExtension.PostXformABMtoEBM	false(default),true	Custom extension just after execution of transformation of Application Business message (ABM) to Enterprise Business Message (EBM) and prior to the call to the Enterprise Business Service (EBS).
ReportDrugSafetyReportSEBLReqABCSImpl	Routing.HealthSciencesDrugSafetyReportEBS.ReportDrugSafetyReport.MessageProcessingInstruction.EnvironmentCode	PRODUCTION(default), CAVS	Setting the value to PRODUCTION indicates that the request must be sent to a concrete endpoint. Setting the value to CAVS indicates that the request must be sent to CAVS Simulator.
ReportDrugSafetyReportSEBLReqABCSImpl	ABCSExtension.PostInvokeEBS	false (default), true	Custom extension just after invocation of the Enterprise Business Service (EBS).
ReportDrugSafetyReportSEBLReqABCSImpl	Routing. HealthSciencesDrugSafetyReportEBS. ReportDrugSafetyReport.RouteToCAVS	false (default), true	To test the requester Application Business Connector Service (ABCS) or when the provider ABCS is not available, you would want the requester ABCS to call a simulator instead of actual Oracle AIA services.
ReportDrugSafetyReportSEBLReqABCSImpl	Routing.HealthSciences DrugSafetyReportEBS.ReportDrugSafetyReport. CAVS.EndpointURI	oramds:/apps/AIAMetaData/AIAComponents/InfrastructureServiceLibrary /AIAValidationSystemServlet/asyncresponsesimulator	End point URL for CAVS, when RouteToCAVS property is set to true

Service Name	Property Name	Value	Description
ReportDrugSafetyReportArgusProvABCSImpl	Routing.HealthSciences DrugSafetyReportEBS.ReportDrugSafetyReport. MessageProcessingInstruction.EnvironmentCode	PRODUCTION(default), CAVS	Setting the value to PRODUCTION indicates that the request must be sent to a concrete endpoint. Setting the value to CAVS indicates that the request must be sent to CAVS Simulator.
ReportDrugSafetyReportArgusProvABCSImpl	Default.SystemID	ARGUS_01	Default System ID associated with this service. It is used for populating the Target System ID when no routing rules match. The Target System ID is used for looking up DVMs and XREFs.
ReportDrugSafetyReportArgusProvABCSImpl	TruncateFields	warning(default), truncate, notruncation	This field is used for controlling truncation of certain fields that exceed maximum length accepted by Oracle Argus Safety. The options are described below: ■ **Warning** - When set to warning, fields get truncated but a warning email about truncated fields is sent to configured users. ■ **Truncate** - When set to truncation, fields get truncated but warning email is not sent. ■ **Notruncation** - When set to notruncate, fields do not get truncated. Oracle Argus Safety rejects the case and a message is added to the Inbox of each Product Issue owner in Siebel AECM.

Service Name	Property Name	Value	Description
ReportDrugSafetyReportArgusProvABCSImpl	ABCSExtension.PreXformEBMtoABM	false(default), true	Customer extension just prior to the execution of the EBM to ABM transformation.
ReportDrugSafetyReportArgusProvABCSImpl	ABCSExtension.PostXFormEBMtoABM	false(default), true	Customer extension just after transforming the Enterprise Business Message to the Argus XML format.
ReportDrugSafetyReportArgusProvABCSImpl	Routing.ReportDrugSafetyReportWriteE2BFileAdapter.RouteToCAVS	false(default), true	To test the provider ABCS or when the provider application is not available, you would want the provider ABCS to call a simulator instead of an actual provider application service.
ReportDrugSafetyReportArgusProvABCSImpl	Routing.ReportDrugSafetyReportWriteE2BFileAdapter.CAVS.EndpointURI	oramds:/apps/AIAMetaData/AIAComponents/InfrastructureServiceLibrary/AIAValidationSystemServlet/asyncresponsesimulator	End point URL for CAVS, when RouteToCAVS property is set to true.
ReportDrugSafetyReportArgusProvABCSImpl	ABCSExtension.PreInvokeABS	false(default), true	Customer extension just prior to the invocation of the file adapter service to write the XML file to the Oracle Argus Safety Interchange server.
ReportDrugSafetyReportArgusProvABCSImpl	ABCSExtension.PostInvokeABS	false(default), true	Customer extension just after invoking the File adapter service to write the XML file to the Oracle Argus Safety Interchange server.
ReportDrugSafetyReportResponseArgusReqABCSImpl	Default.SystemID	ARGUS_01	Default System ID associated with this service. It is used for looking up DVMs, XREFS, when no other system IDs are found in the system registry.
ReportDrugSafetyReportResponseArgusReqABCSImpl	ABCSExtension.PreXformABMtoEBM	false(default), true	Custom extension prior to the execution of transformation of the Argus acknowledgement message to the ReportDrugSafetyResponse message

Service Name	Property Name	Value	Description
ReportDrugSafetyReportResponseArgusReqABCSImpl	ABCSExtension.PostXformABMtoEBM	false(default), true	Customer extension just prior to the invocation of Enterprise Business Service and after transforming the Argus acknowledgement message to the ReportDrugSafetyResponse message.
ReportDrugSafetyReportResponseArgusReqABCSImpl	Routing.ReportDrugSafetyReportResponseReadAckFileAdapter.RouteToCAVS	false(default), true	To test the requester Application Business Connector Service (ABCS) or when the provider ABCS is not available, you would want the requester ABCS to call a simulator instead of actual Oracle AIA services.
ReportDrugSafetyReportResponseArgusReqABCSImpl	Routing.ReportDrugSafetyReportResponseReadAckFileAdapter.CAVS.EndpointURI	oramds:/apps/AIAMetaData/AIAComponents/InfrastructureServiceLibrary/AIAValidationSystemServlet/asyncresponsesimulator	End point URL for CAVS, when RouteToCAVS property is set to true
ReportDrugSafetyReportResponseArgusReqABCSImpl	ABCSExtension.PreInvokeEBS	false (default), true	Custom extension prior to the invocation of the enterprise business service (EBS).
ReportDrugSafetyReportResponseArgusReqABCSImpl	ABCSExtension.PostInvokeEBS	false (default), true	Custom extension after the invocation of the enterprise business service (EBS).
ReportDrugSafetyReportResponseArgusReqABCSImpl	Routing.HealthSciencesDrugSafetyReportEBSResponse.ReportDrugSafetyReportResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION(default), CAVS	Setting the value to PRODUCTION indicates that the request must be sent to a concrete endpoint. Setting the value to CAVS indicates that the request must be sent to CAVS Simulator.
ReportDrugSafetyReportResponseSEBLProvABCSImpl	Default.SystemID	SEBLCLIN_01	Default System ID associated with this service. It is used to set the TargetSystemID in the EBM header when no routing rules match. The TargetSystemID is used for looking up DVMs and XREFs.

Service Name	Property Name	Value	Description
ReportDrugSafetyReportResponseSEBLProvABCSImpl	ABCSExtension.PreXformEBMtoABM	false (default), true	Customer extension just prior to the execution of the EBM to ABM transformation.
ReportDrugSafetyReportResponseSEBLProvABCSImpl	ABCSExtension.PostXformEBMtoABM	false (default), true	Customer extension just prior to the invocation of the Siebel Web services.
ReportDrugSafetyReportResponseSEBLProvABCSImpl	Routing.LS_spcMedical_spcProduct_spcIssue_spcCreate_spcInbox_spcItem_spcInbound.RouteToCAVS	false (default), true	To test the provider ABCS or when the provider application is not available, you would want the provider ABCS to call a simulator instead of an actual provider application service.
ReportDrugSafetyReportResponseSEBLProvABCSImpl	Routing.LS_spcMedical_spcProduct_spcIssue_spcCreate_spcInbox_spcItem_spcInbound.CAVS.EndpointURI	oramds:/apps/AIAMetaData/AIAComponents/InfrastructureServiceLibrary/AIAValidationSystemServlet/syncresponsesimulator	End point URL for CAVS, when RouteToCAVS property is set to true
ReportDrugSafetyReportResponseSEBLProvABCSImpl	Routing.LS_spcMedical_spcProduct_spcIssue_spcCreate_spcInbox_spcItem_spcInbound.MessageProcessingInstruction.EnvironmentCode	PRODUCTION(default), CAVS	Setting the value to PRODUCTION indicates that the request must be sent to a concrete endpoint. Setting the value to CAVS indicates that the request must be sent to CAVS Simulator.
ReportDrugSafetyReportResponseSEBLProvABCSImpl	ABCSExtension.PreInvokeABS	false (default), true	Customer extension just prior to the invocation of Siebel web services.
ReportDrugSafetyReportResponseSEBLProvABCSImpl	ABCSExtension.PostInvokeABS	false (default), true	Customer extension just after invoking the Siebel Web services and after invoking Application Service.

Properties specific to Siebel Session Pool Manager are described below:

Service Name	Property Name	Value	Description
AIASessionPoolManager	all_hosts	NOSERVER SEBLCLIN_01	SPM can work with multiple application web server instances. In this property, list the hosts for which SPM can create a session token pool. Separate the host names by spaces. Each host has its own pool. This property is not prefixed with a HostId value.
AIASessionPoolManager	all_ hosts.ProxySettings_Enabled	False	To enable SPM use proxy settings while calling the application web server, set this property to TRUE. Set this property to FALSE to not use proxy settings.
AIASessionPoolManager	all_ hosts.Proxy.Host	Specified by values populated in the Configuration Wizard.	It determines the server to be set in the system properties for http.proxyHost property. In the java.net API used by SPM, proxies are supported through two system properties: http.proxyHost and http.proxyPort. They must be set to the proxy server and port respectively. This value is only set when ProxySettings_ Enabled is set to TRUE.
AIASessionPoolManager	all_ hosts.Proxy.Port	Specified by values populated in the Configuration Wizard.	It determines the port to be set in the system properties for the http.proxyPort property. In the java.net API used by SPM, proxies are supported through two system properties: http.proxyHost and http.proxyPort. They must be set to the proxy server and port respectively. This value is only set when ProxySettings_ Enabled is set to TRUE.

Service Name	Property Name	Value	Description
AIASessionPoolManager	SEBLCLIN_01.EndpointURI	Populated by values specified in the Configuration Wizard. (For example, <InternetProtocol>://<hostname>:<port>/eai_<language>/start.swe?SWEEXTSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1	It determines the endpoint URI that Session Pool Manager uses to connect to the application web server.
AIASessionPoolManager	SEBLCLIN_01.UserId	Populated by values specified in the PIP Configuration Wizard.	It determines the user ID that is used to connect to the application web server.
AIASessionPoolManager	SEBLCLIN_01.Password	Populated by values specified in the PIP Configuration Wizard.	It determines the password that is used to connect to the application web server. This value is stored in the KEY store and is encrypted.
AIASessionPoolManager	SEBLCLIN_01.PredictExpiration_Idle	780000	Indicates the maximum time in milliseconds that a session token can be idle before expiring. We recommend that you set this value to a value lower than the actual maximum idle time configured for the application web server. We recommend a value lower than the actual value to compensate for the gap between the time at which the application web server responded and the time at which the BPEL flow called SPM to release the session token.

Service Name	Property Name	Value	Description
AIASessionPoolManager	SEBLCLIN_01.PredictExpiration_Age	82800000	Indicates the maximum age in milliseconds that a session token can reach before expiring. We recommend that you set this value to a value lower than the actual maximum age configured for the application web server. The creation time registered in the application web server is some seconds earlier than the one registered in SPM. A value of 1 or 2 minutes is a good start. For example, if the maximum age configured on the application web server is 15 minutes, set this property to 13 minutes.
AIASessionPoolManager	SEBLCLIN_01.InvalidSessionErrorCodes	10944642|SBL-BPR-00162|SBL-DAT-00175|11338608|SBL-UIF-00880	It determines the list of error codes that the application web server can return for a fault when the session token is not valid.
AIASessionPoolManager	SEBLCLIN_01.ClassName	oracle.apps.aia.core.sessionpool.CRMSiebelSession	It determines the full class name that SPM uses to get the session tokens from the application server. The class listed in this property implements the oracle.apps.aia.core.sessionpool.PoolableResource interface.

3.6 Updating Server Information for Siebel AECM and Oracle Argus Safety

At times, applications are moved to new servers or databases for various reasons. This section describes how to update the information that was provided during Install time when the integration is already installed and deployed.

To modify information about Siebel AECM, follow these steps:

1. The information about the Siebel AECM Host name and Port given during install is used to define the EndPointURI to call the Siebel Web services. If you have moved Siebel AECM to a new server, you must update the SEBCLIN_01.EndPointURI property in the AIAConfigurationProperties.xml file.

   Navigate to $AIA_INSTANCE/AIAMetaData/config directory and open the file AIAConfigurationProperties.xml. Search for SEBLCLIN_01.EndpointURI and replace the Host name and Port, if they have changed.

2. Upload updated AIAConfigurationProperties.xml file to MDS. Follow these steps to upload the file to MDS:

   a. Update the fileset element in the $AIA_INSTANCE/config/UpdateMetaDataDP.xml file to point to the location of the AIAConfigurationProperties.xml file as shown below:

      <fileset dir=$AIA_INSTANCE/AIAMetaData/">

      <include name=" config/AIAConfigurationProperties.xml" />

      </fileset>

   b. Upload to MDS using the command given below. Wait till you see a build successful result.

      ```
      source $AIA_INSTANCE/bin/aiaenv.sh

      ant -f <AIA_
      HOME>/Infrastructure/Install/config/UpdateMetaData.xml
      ```

3. If you have changed the user name and password for the Siebel EAI user that you had specified during installation, follow the steps given below. To help you change passwords, AIA provides a utility called UpdateStore. AIA stores its passwords in Keystore. UpdateStore utility aids you in modifying the existing passwords in the Keystore.

   Steps:

   a. Source the aiaenv.sh in the $AIA_INSTANCE/bin.

   b. Navigate to $AIA_HOME/util/

   c. Execute `ant -f updateStore.xml updateStore -DAdminUsername=<weblogic adminusername> -DAdminPassword=<weblogic admin password>`

      Update **AIA Keystore** screen appears.

   d. Enter the **Existing Username**, **Existing Password**, **New Username**, and **New Password** and the Xpath of the password that you want to change.

      The Xpath is /properties/participatingapplications/sc/server/eai/password

4. If the Siebel Enterprise Server name is different, you must log into the AIA Application and navigate to the System Registry. You must change the value of Internal ID for the row with System ID = SEBCLIN_01 to the name of the new Siebel Enterprise Server.

To configure a different Oracle Argus Safety Interchange server, follow these steps:

Note: This scenario assumes that there is an existing mount point between integration pack SOA_server and Oracle Argus Safety Interchange server. The mount point is reconfigured to point to the new Oracle Argus Safety Interchange server.

On integration pack SOA_server, change the file mount on SOA_server to point to the new Oracle Argus Safety Interchange (ESM) server.

To configure a different folder for Oracle Argus Safety Interchange server, follow these steps:

Note: This scenario assumes that there is an existing folder on integration pack SOA_server that is mounted on to Oracle Argus Safety Interchange server. You may choose the integration pack to use a different folder on integration pack SOA_server to mount with Oracle Argus Safety Interchange server.

Update $AIA_INSTANCE/config/AIAInstallProperties.xml using the following steps:

1. To change the E2B files input directory: Search for element <xmldir> and replace the value contained within the <xmldir> node with the new value.
2. To change the Acknowledgement files output directory: Search for element <ackdir> and replace the value contained within the <ackdir> node with the new value.

   Note that <xmldir> and <ackdir> nodes are present within the node structure of <properties>, then <participatingapplications>, then <argus>, then <server>, and then <case>.

   For more information about configuring Oracle Argus Safety, see *Oracle® Device and Drug Adverse Event Data Integration Pack for Siebel Adverse Event Complaint Management and Oracle Argus Safety Installation Guide*, "Performing Post-Installation Configurations".
3. You need to upload the modified AIAInstallProperties.xml file to the MDS before the deployment.
   a. Navigate to $AIA_INSTANCE/config
   b. Copy UpdateMetaDataDp.xml to a backup file.
   c. Edit the file as follows:

```
<?xml version="1.0" standalone="yes"?>
<DeploymentPlan component="Metadata" version="3.0">
<Configurations>
<UpdateMetadata wlserver="fp" >
! <fileset dir="${AIA_HOME}/AIAMetaData">
<include name="config/AIAInstallProperties.xml" />
</fileset>
</UpdateMetadata>
</Configurations>
</DeploymentPlan>
```

   d. Navigate to $AIA_ INSTANCE/bin

      source `aiaenv.sh`

      ```
      ant -f $AIA_
      HOME/Infrastructure/Install/config/UpdateMetaData.xml
      ```

      Wait to see the BUILD SUCCESSFUL message.
4. Create a custom deployment plan to deploy:
   - ReportDrugSafetyReportReadAckFileAdapter
   - ReportDrugSafetyReportWriteE2BFileAdapter composites
   a. Open the file $AIA_ HOME/pips/DrugDeviceAESEBLandArgus/DeploymentPlans/DrugDeviceAESEBLandArgusCustomDP.xml
   b. Replace <Deployments> </Deployments> with the following:

      <Deployments>

```
<Composite compositeName="ReportDrugSafetyReportWriteE2BFileAdapter
"

compositedir="$AIA_
HOME/services/core/ArgusSafety/AdapterServices/ReportDrugSafetyRepo
rtWriteE2BFileAdapter " revision="1.0"
wlserver="pips.DrugDeviceAESEBLandArgus" action="deploy"
overwrite="true"/>

<Composite compositeName="ReportDrugSafetyReportReadAckFileAdapter
"

Compositedir="$AIA_HOME/services/core/ArgusSafety/AdapterServices
/ReportDrugSafetyReportReadAckFileAdapter " revision="1.0"
wlserver="pips.DrugDeviceAESEBLandArgus" action="deploy"
overwrite="true"/>

</Deployments>
```

c. Save DrugDeviceAESEBLandArgusCustomDP.xml

5. Run customDP by executing the following command:

a. Set the environment variables by executing "`source <AIA_HOME>/aia_instances/AIA11115/bin/aiaenv.sh`"

b. Run the custom deployment command for deploying the customized artifacts:

```
ant -f $AIA_
HOME/Infrastructure/Install/AID/AIAInstallDriver.xml
-DDeploymentPlan=$AIA_
HOME/pips/DrugDeviceAESEBLandArgus/DeploymentPlans/DrugDev
iceAESEBLandArgusCustomDP.xml -DPropertiesFile=$AIA_
INSTANCE/config/AIAInstallProperties.xml -l <location and
name of the log file>
```

3.7 Identifying Cross-References

Cross-references map and connect the records within the application network, and enable these applications to communicate in the same language. The integration server stores the relationship in a persistent way so that others can refer to it.

For more information about cross-references, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*, "Working with Cross References".

This is the cross-reference for Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP:

XREFTABLENAME	COLUMN NAME	DESCR	USAGE
DRUGSAFETYREPORT_CASEID	COMMON	System generated GUID to uniquely identify the case.	This column is populated when a row is created in the table when a new Product Issue is sent to Oracle Argus Safety.
DRUGSAFETYREPORT_CASEID	SEBL_01	Product Issue ID	This column is populated when a row is created in the table when a new Product Issue is sent to Oracle Argus Safety

XREFTABLENAME	COLUMN NAME	DESCR	USAGE
DRUGSAFETYREPORT_CASEID	ARGUS_01	The current Argus Case ID that the Product Issue is associated with.	This column is populated with the Case ID in Oracle Argus Safety that a new case/follow-up is accepted into. If a follow-up is accepted into a different case than the original product issue was accepted into, this field is updated with the new Case ID.

3.8 Working with Domain Value Maps

Domain value maps (DVMs) are a standard feature of the Oracle SOA Suite and enable you to equate lookup codes and other static values across applications. For example, "FOOT" and "FT" or "US" and "USA."

DVMs are static in nature, though administrators can add and update additional maps as needed. Transactional business processes never update DVMs-they only read from them.

Most of the DVMs below are pre-populated with the default values that ship with Siebel AECM and the default values that are expected by Oracle Argus Safety when importing an E2B+ file.

For fields which are part of the E2B standard, Oracle Argus Safety expects the values required by E2B when importing the case into Oracle Argus Safety. This PIP pre-populates the DVMs with these codes for you.

For more information about Language Independent Codes (LIC), see *Configuring Siebel Business Applications, Version 8.1, Rev. B*,"Localizing Siebel Business Applications - Localizing a Multilingual List of Values".

For more information on the codes, see *Electronic Transmission of Individual Case Safety Reports Message Specification v2.3.*

These are the DVMs for Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP:

DVM Name	DVM Column Name	Comments
DRUGSAFETYREPORT_REPORTTYPE	SEBLCLIN_01, COMMON, ARGUS_01	This maps the Report Type display value in Siebel AECM to the E2B code expected by Oracle Argus Safety.
DRUGSAFETYREPORT_QUALIFICATIONCODE	SEBLCLIN_01, COMMON, ARGUS_01	This maps the language independent code (LIC) defined in Siebel tools for the Qualification code value to the E2B code expected by Oracle Argus Safety for Reporter type.
GENDER	SEBLCLIN_01, COMMON, ARGUS_01	This maps the display values for gender in Siebel AECM to the E2B codes expected by Oracle Argus Safety.

DVM Name	DVM Column Name	Comments
DRUGSAFETYREPORT_ ADVERSEEVENTREPORTEDOUTCOME	SEBLCLIN_01, COMMON, ARGUS_ 01	This maps the Reported Outcome for the adverse event in Siebel AECM to the Reported Outcome for the adverse event in Oracle Argus Safety.
DRUGSAFETYREPORT_ ADVERSEEVENTSEVERITY	SEBCLIN_01, COMMON, ARGUS_ 01	This maps the adverse event severity in Siebel AECM to the Adverse Event intensity in Oracle Argus Safety.
DRUGSAFETYREPORT_ DRUGADMINISTRATIONROUTE	SEBLCLIN_01, COMMON, ARGUS_ 01	This maps the Route Used for a Product in Siebel AECM to the Patient Route of Administration in Oracle Argus Safety.
DRUGSAFETYREPORT_DEVICEOPERATOR	SEBCLIN_01, COMMON, ARGUS_ 01	This maps the Device Operator in Siebel AECM to the Operator of Device in Oracle Argus Safety.
DRUGSAFETYREPORT_DEVICEUSE	SEBLCLIN_01, COMMON, ARGUS_ 01	This maps Usage of Device for the Product Issue in Siebel AECM to Usage of Device in Oracle Argus Safety.
DRUGSAFETYREPORT_ DEVICEREREMEDIALACTION	SEBLCLIN_01, COMMON, ARGUS_ 01	This maps the Remedial Action Type in Siebel AECM to the Remedial Action Taken in Oracle Argus Safety. **Note**: Do not change the ARGUS_01 values in this DVM. The display in Oracle Argus Safety is dependent upon the values we ship with for this DVM.
DRUGSAFETYREPORT_STATUS	SEBLCLIN_01, COMMON, ARGUS_ 01	This maps the acceptance/rejection of the case in Oracle Argus Safety to the Product Issue status in Siebel AECM.
STATE DVM	SEBLCLIN_01, COMMON, ARGUS_ 01	This maps the state in Siebel AECM to the state part of an address in Oracle Argus Safety. By default the 50 states in the U.S. are populated for Siebel AECM and COMMON. There are no values pre-populated for ARGUS_01. Oracle Argus Safety does not validate the state against a codelist so there is no need to modify this list. However, if the list is not populated the value specified in the COMMON column is used in the addresses in Oracle Argus Safety.

DVM Name	DVM Column Name	Comments
COUNTRY DVM	SEBLCLIN_01, COMMON, ARGUS_01	This maps the country of the Product Issue contact to the country of the reporter in Oracle Argus Safety. This also maps the Country Of Incidence in Siebel AECM to the Country of Incidence in Oracle Argus Safety. It also maps the country specified in an address in Siebel AECM that is sent to an address in Oracle Argus Safety.
UNIT OF MEASURE DVM	SEBLCLIN_01, COMMON, ARGUS_01	This maps the Unit of Measure used in Siebel AECM for the product Dose/Unit to the unit of measure in Oracle Argus Safety for the Product Dosage.

For more information about working with DVMs, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite 11g Release 1,* "Working with Domain Value Maps" and "Using Oracle SOA Composer with Domain Value Maps".

3.9 Handling Errors

Error Messages are described in the table below:

Error Condition	Message Name	Message Text
Requester ABCS throws an exception when downstream services are not available	ORAMED-03302:[Exception in one way execution]	Unexpected exception in one-way operation "ReportDrugSafetyReport" on reference "ReportDrugSafetyReportSEBLReqABCSImpl".Possible Check whether the reference service is properly configured and running or look at exception for analyzing the reason or contact Oracle Support Services. Refer to the section Using the Message Resubmission Utility for the steps to resubmit failed messages.
DVM values not found	DVMValueNotFound	"The value entered for the REPORTTYPE does not exist in <SEBLCLIN_01> column for the DRUGSAFETYREPORT_REPORTTYPE DVM. Please add the value to the DRUGSAFETYREPORT_REPORTTYPE DVM."
DVM values not found	DVMValueNotFound	When E2B field reporttype DVM look up fails because report type data does not match DVM value, the following text is used: "The value entered for the REPORTTYPE does not exist in <ARGUS_01> column for the DRUGSAFETYREPORT_REPORTTYPE DVM. Please add the value to the DRUGSAFETYREPORT_REPORTTYPE DVM."
TruncateFields is set to 'warning' and one or more fields have exceeded maximum length.	ArgusProviderTruncateFieldsWarning	The following fields were truncated due to exceeding the length required by Oracle Argus Safety: <DTD field name 1> : <length required by Argus> … <DTD field name N> : <length required by Argus>

Error Condition	Message Name	Message Text
Siebel inbound web services are down. In such case, AIA Session Pool Manager is set to retry 3 times to connect to the Siebel Server with 500 milli seconds between each retry attempt.	404	Page Not Found.
Response SEBL Provider ABCS fails to call Siebel Web Services.	ORAMED-03302:[Exception in oneway execution]	Unexpected exception in one-way operation "ReportDrugSafetyReportResponse" on reference "ReportDrugSafetyReportResponseSEBLProvABCSImpl ".Possible Check whether the reference service is properly configured and running or look at exception for analyzing the reason or contact Oracle Support Services. Please Refer to the section Using the Message Resubmission Utility for the steps to resubmit failed messages.

For more information about AIA error handling, see the Oracle Application Integration Architecture - *Foundation Pack: Core Infrastructure Components Guide,* "Setting Up and Using Error Handling and Logging."

3.10 Using the Message Resubmission Utility

This section describes the steps for using the Resubmission utility when the flows fail. For example, if a flow fails because of a system fault like a server being down for maintenance, you can resubmit the flow when the server is up.

AIA provides Message Resubmission Utility to re-submit failed messages for the flow from Siebel AECM to Argus Safety. The following steps give an overview of resubmitting messages to the JMS queue.

To use the Message Resubmission Utility:

1. Set the following values in the ResubmissionParams.properties file located in $AIA_HOME/util/AIAMessageResubmissionUtil:
   a. jms.app.userName=<weblogic administrator user> (For example, weblogic):
   b. jms.app.password=< Weblogic administrator password>
   c. jms.moduleName=SEBLJDBCJMSModule
   d. jms.resourceCFJndi= jms/aia/AIA_SEBLDrugSafetyReportJMSQueueCF
   e. jms.errorResourceCFJndi= AIA_SEBLDrugSafetyReportJMSQueueCF
   f. resourceType=1
   g. resourceName= AIA_SEBLDrugSafetyReportJMSQueue
   h. messageID= Click the link in the email notification of the fault to get the value of message ID. The flow that is faulted is indicated in the Enterprise Manager. Use the value of <corecom:SenderMessageID> from the EBM header as the message ID.
2. For Windows: execute `$AIA_INSTANCE\bin\aiaenv.bat`.

3. For Linux: source `$AIA_INSTANCE/bin/aiaenv.sh`.
4. Navigate to $AIA_HOME/util/AIAMessageResubmissionUtil and execute the following ant command:

   ```
   ant -f MessageResubmit.xml -l $AIA_
   HOME/util/AIAMessageResubmissionUtil/MessageResubmit.log
   ```

5. The MessageResubmit.xml script references the edited ResubmissionParams.properties file. Once run, the script resets the message status back to a ready state so that the transaction can resume its flow.

Using the Resubmission Utility for the Failure of Flow from Oracle Argus Safety to Siebel AECM

Follow these steps for using the resubmission utility when the flow from Oracle Argus Safety to Siebel AECM fails:

1. Set the following values in the ResubmissionParams.properties file located in $AIA_HOME/util/AIAMessageResubmissionUtil:
   a. jms.app.userName=<weblogic administrator user> (e.g. weblogic)
   b. jms.app.password=<weblogic administrator password>
   c. jms.moduleName=SEBLJDBCJMSModule
   d. jms.resourceCFJndi= jms/aia/AIA_SEBLDrugSafetyReportJMSQueueCF
   e. jms.errorResourceCFJndi= AIA_SEBLDrugSafetyReportJMSQueueCF
   f. resourceType=3
   g. resourceName= default/ReportDrugSafetyReportReadAckFileAdapter!1.0
   h. messageID= Get the value of messageID by navigating to the Enterprise Manager from the link in the email notification. Click the **Recovery** link in the faults section. Enter the group ID shown as the value of messageID.
2. For Windows: execute `$AIA_INSTANCE\bin\aiaenv.bat`
3. For Linux: source `$AIA_INSTANCE/bin/aiaenv.sh`
4. Navigate to $AIA_HOME/util/AIAMessageResubmissionUtil and execute the following ant command:

   ```
   ant-f MessageResubmit.xml -l $AIA_
   HOME/util/AIAMessageResubmissionUtil/MessageResubmit.log
   ```

   The MessageResubmit.xml script references the edited ResubmissionParams.properties file. Once run, the script resets the message status back to a ready state so that the transaction can resume its flow.

3.11 Extending the Device and Drug Adverse Event: Siebel AECM and Argus Safety PIP

This integration can be extended to send the fields that you have added to Siebel AECM or Oracle Argus Safety.

This section describes an example of how the extended fields can be sent across. In this example, assume that:

- **Patient Extension** field is added **Product Issues** screen in Siebel AECM
- **Patient Extension** field is added **Patient** tab in Oracle Argus Safety

- **Patient Extension** field is added with the name PatientExtension to the Integration Object using Siebel Tools after the field **AETypeCalc**

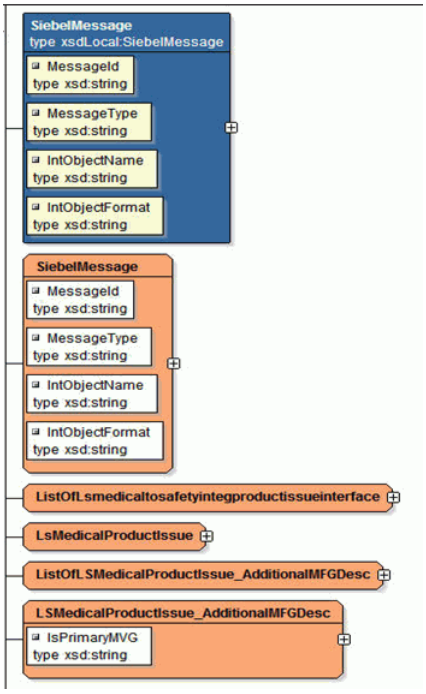
Overview of the Steps:

1. Add new Element to Siebel AECM xsd. Upload to MDS
2. Add new Element to Custom EBO xsd. Upload to MDS
3. Verify Custom EBO is uploaded to MDS
4. Add new Element to Oracle Argus Safety xsd. Upload to MDS
5. Add custom transformations to custom xsl in ReportDrugSafetyReportSEBLRequesterABCS composite
6. Add custom transformations to custom xsl in ReportDrugSafetyReportArgusProviderABCS composite
7. Create a custom deployment plan to deploy composites on SOA server
8. Modify Argus DTD, and run SQL to add new field to Oracle Argus Safety
9. Test Extension field.

To map the newly added custom field in Siebel AECM payload to the corresponding newly added custom field in Oracle Argus Safety follow these steps:

1. **Add New Element to Siebel xsd**:
   a. It is advisable that you back up the .xsd file first, before adding a new element to it. Start Jdeveloper and open file <AIA_HOME>/AIAMetaData/AIAComponents/ApplicationObjectLibrary/SiebelAECM/V1/schemas/ListOfLSMedicalToSafetyIntegProductIssueInterfaceTarget.xsd

Figure 3–1 Adding New Element to Siebel xsd



b. As per the example used in this section, PatientExtension element is mapped in the Siebel AECM. This element is available at the location:

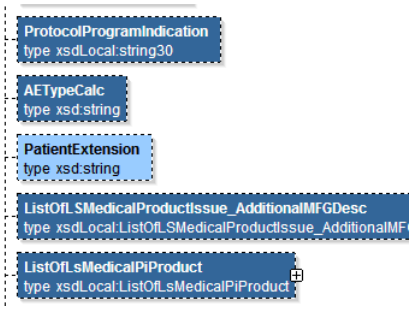
LsMedicalProductIssue > **PatientExtension** below the **AETypeCalc** element

Expand the element **LsMedicalProduct Issue** and locate the **AETypeCalc** element.

Right click **AETypeCalc** element and select Insert after element - **AETypeCalc**> element.

c. Name this element as **PatientExtension** of type xsd:string as shown below and save the changes.

Figure 3–2 Element named as PatientExtension



d. Upload the modified xsd to MDS using these steps:

1. Place the modified xsd on SOA server in the following location: <AIA_HOME>/AIAComponents/ApplicationObjectLibrary/SiebelAECM/V1/schemas
2. Update the fileset element in the UpdateMeataDataDP.xml file (at $AIA_INSTANCE/config) to point to the location of the xsd file as shown below:

   <fileset dir="${AIA_HOME}/AIAMetaData">

   <include name="AIAComponents/ApplicationObjectLibrary/SiebelAECM/V1/schemas/ListOfLSMedicalToSafetyIntegProductIssueInterfaceTarget.xsd" />

   </fileset>

3. Upload to MDS using the command below. Wait till you see a build successful result.

   source `/slot/ems3390/oracle/AIA11115/aia_instances/AIA11115/bin/aiaenv.sh`

   `ant-f/slot/ems3390/oracle/AIA11115/Infrastructure/Install/config/UpdateMetaData.xml`

2. **Add a New Element to the EBO**:

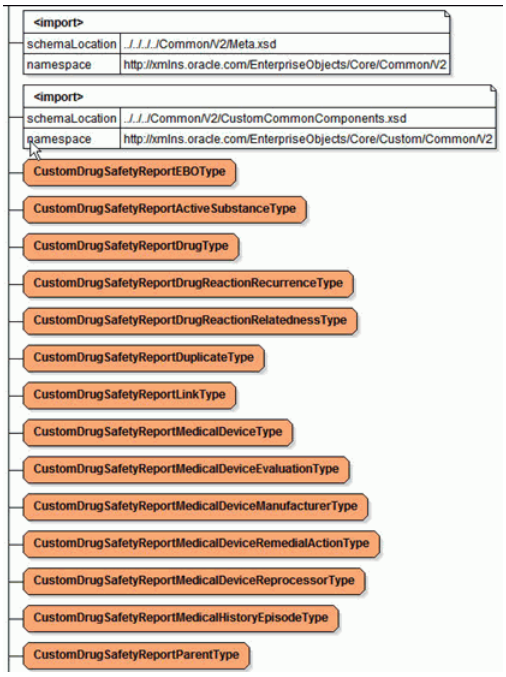
a. In the current example since we are adding **Patient Extension** field we shall extend the component DrugSafetyReportPatient in the EBO to include this field.

b. It is advisable that you back up the .xsd file first, before adding a new element to the EBO. Start JDeveloper and open the file: <AIA_HOME>/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry

/HealthSciences/Custom/EBO/DrugSafetyReport/V1/CustomDrugSafetyReportEBO.xsd

c. There are several types defined in the xsd, as shown below:

Figure 3–3 xsd File Types

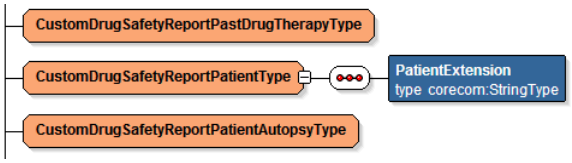


d. Each type name is a place holder for the Custom field it extends. For example, the custom field place holders for DrugSafetyReportPatient is CustomDrugSafetyReportPatientType.

e. Right click CustomDrugSafetyReportPatientType, and select **Insert inside complexType** > **Sequence**

f. Right click the new sequence and select **Insert inside Sequence** > **element** to add a new element to the sequence. Name the element as **PatientExtension**. The new element should appear as shown below:

Figure 3–4 Added PatientExtension Element



g. Set Datatype for the new element that is added. Right click the element and select **Set Type**. From the drop-down list select **corecom:StringType**. Save the xsd file.

h. Upload the custom file to MDS using these steps:

1. Place the xsd file in the following directory: <AIA_HOME>/AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/Custom/EBO/DrugSafetyReport/V1/

2. Update the $AIA_INSTANCE/config/UpdateMetaDataDP.xml to have following fileset entry:

   <fileset dir="${AIA_HOME}/AIAMetaData">

   <include name="AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/Custom/EBO/DrugSafetyReport/V1/CustomDrugSafetyReportEBO.xsd" />

   </fileset>

3. Upload to MDS using the command given below. Wait till you see a build successful result.

   source

   ```
   $AIA_INSTANCE/bin/aiaenv.sh
   ```

   ```
   ant-f <AIA_
   HOME>/Infrastructure/Install/config/UpdateMetaData.xml
   ```

3. **Verify Custom EBO xsd Upload to MDS**:

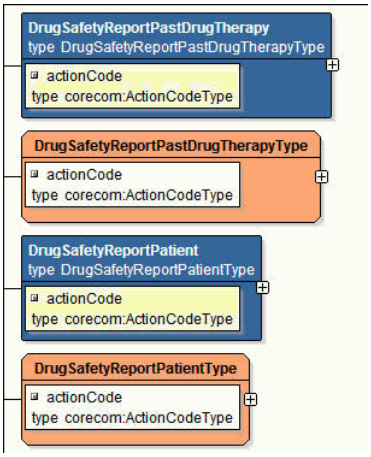
   (Assumes that JDeveloper Resource Palette is configured to browse SOA_MDS database for a given database.)

   a. In JDeveloper, open the **Resource Palette** and navigate to the files mentioned above at the specified locations.

   b. Double click the files to open, and verify the changes made.

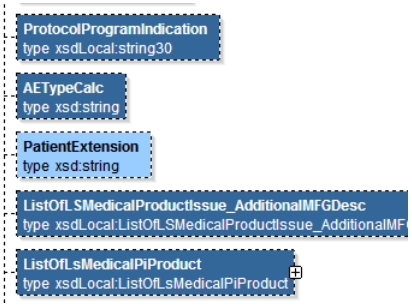
   c. Also verify the AIAMetaData\AIAComponents\EnterpriseObjectLibrary\Industry\HealthSciences\EBO\DrugSafetyReport\V1\DrugSafetyReportEBO.xsd to check whether the changes made to the CustomDrugSafetyReportEBO.xsd are reflected. Locate the DrugSafetyReportPatient element in the xsd.

Figure 3–5 DrugSafetyReportPatient Element



   d. Expand the element and navigate to the end to locate the Custom element.

Figure 3–6 Custom Element



4. **Modify Argus xsd**:

   a. It is advisable that you back up the .xsd file first, before modifying it.

      In Jdeveloper open file <AIA_HOME>/ AIAComponents/ApplicationObjectLibrary/ArgusSafety/V1/schemas/ich-icsr-v2.1_integration.xsd

   b. Locate the element under which the new field should appear. In this example, the field should be present under the location: **ichicsr** > **safetyreport** > **patient**.

      Always add the custom element at the end of the sequence of children elements.

   c. Add the reference properties of newly created reference in the same xsd. For example, for patientextn_extension, add the following to the same xsd:

Figure 3–7 Add Reference Properties

```
<xs:element name="patientextn_extension">
   <xs:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:string">
            <xs:attribute name="lang" use="optional" type="xs:string"/>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
</xs:element>
```

   d. Save the changes to the xsd.

   e. Upload the xsd to MDS using these steps:

   1. Copy the XSD to a directory structure similar to that of MDS.

   2. Update the $AIA_INSTANCE/config/UpdateMetaDataDP.xml to have following fileset entry:

      <fileset dir="${AIA_HOME}/AIAMetaData">

      <include name="AIAComponents/ApplicationObjectLibrary/ArgusSafety/V1/schemas/ich-icsr-v2.1_integration.xsd" />

      </fileset>

   3. Upload to MDS using the command given below. Wait till you see a build successful result.

      source `$AIA_INSTANCE/bin/aiaenv.sh`

      `ant-f <AIA_HOME>/Infrastructure/Install/config/UpdateMetaData.xml`

5. **Add Transformation to ReportDrugSafetyReportSEBLReqABCSImpl**

   Assumes that JDeveloper is set up with the following configurations:

   - Resource Palette is configured to browse SOA_MDS database for a given database.
   - JavaWS application is configured to which the .jpr files can be added.

   a. Get the following ReportDrugSafetyReportSEBLReqABCSImpl and ReportDrugSafetyReportArgusProvABCSImpl from application server to local JDeveloper environment.

   This may involve copying corresponding composite projects from the application server's folders <AIA_HOME>/services/core/SiebelAECM/RequesterABCS and $AIA_HOME/services/ core/ArgusSafety/ProviderABCS to the local JDeveloper environment.

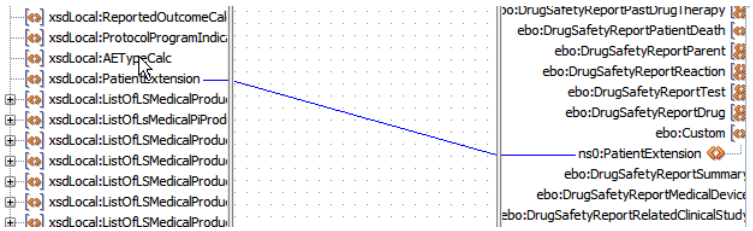
   b. It is advisable that you back up the .xsl file first, before the mapping activity.

   In JDeveloper, click **Open**, navigate to ReportDrugSafetyReportSEBLReqABCSImpl.jpr

   c. Expand the project and, under the xsl folder, open Xform_DrugSafetyReportABMReqMsg_to_DrugSafetyReportEBMReqMsg_Custom.xsl

   d. In the JDeveloper design view, map the newly added custom field **PatientExtension** in Siebel ABM and **PatientExtension** in ReportDrugSafetyReportEBM

Figure 3–8 Map Custom Field



   e. Navigate to the **Source** tab. Note that a template contains the mapping that you created above under "<xsl:template match="/">" element.

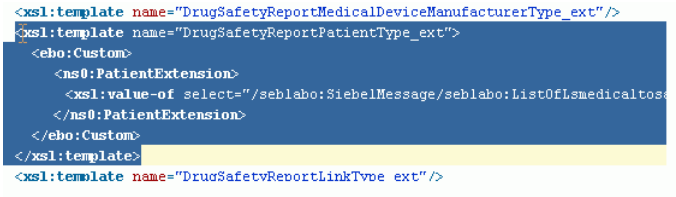
Figure 3–9 Template with the Mapping

```
<xsl:template match="/">
  <ebo:ReportDrugSafetyReportEBM>
    <ebo:DataArea>
      <ebo:ReportDrugSafetyReport>
        <ebo:DrugSafetyReportPatient>
          <ebo:Custom>
            <ns0:PatientExtension>
              <xsl:value-of select="/seblabo:SiebelMessage/seblabo:I
            </ns0:PatientExtension>
          </ebo:Custom>
        </ebo:DrugSafetyReportPatient>
      </ebo:ReportDrugSafetyReport>
    </ebo:DataArea>
  </ebo:ReportDrugSafetyReportEBM>
</xsl:template>
```

   f. There are multiple templates provided in the custom xsl. For example, there is ReportDrugSafetyReportDrugType_ext for extending DrugSafetyReportDrug

component of the ebo, ReportDrugSafetyReportMedicalDeviceType_ext for extending DrugSafetyReportDevice component of the ebo. Cut the content enclosed in <ebo:custom> tag and paste the same in the appropriate template. For the current example, ReportDrugSafetyReportPatientType_ext template should be used.

Figure 3–10 Adding Custom Field in the Template



Remove the content enclosing <xsl:template match="/"> tag, which was auto-generated by the designer. Save your changes to custom xsl.

Note: The xsl may also be modified using other text editors. Please ensure that namespaces (For example: ebo, ns0 are appropriately assigned in such cases).

6. **Add Transformation to ReportDrugSafetyReportArgusProvABCSImpl**

   Assumes that JDeveloper is set up with the following configurations:

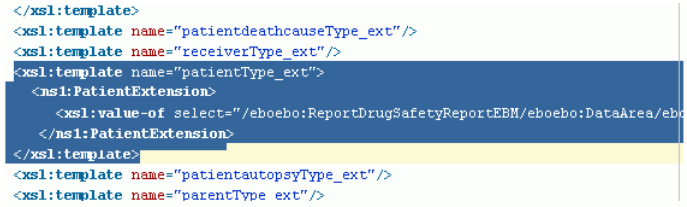
   - Resource Palette is configured to browse SOA_MDS database for a given database
   - JavaWS application is configured to which .jpr files can be added.

   a. It is advisable that you back up the .xsl file first. In JDeveloper, click **Open,** navigate to Xform_DrugSafetyReportReqMsg_to_DrugSafetyReportRespMsg_Custom.xsl

   b. In the JDeveloper design view, map the custom field **PatientExtension** in ReportDrugSafetyReportEBM to patientextn_extension in Argus ABM.

   c. Navigate to **Source** view. Note that there are multiple templates provided in the custom xsl. Design view adds the mapping created in step **b** above in a default template <xsl:template match="/">. Modify the default template by cutting the content enclosed by "_extension" element and pasting it in the correct template. For example, <patientextn_extension> should be moved to patientType_ext template as shown below.

Figure 3–11 patientType_ext Template



Remove the content surrounding <xsl:template match="/">, which was auto-generated by the designer. Save your changes to the custom xsl.

7. **Deploy the Composites**:

   a. On the application server, ensure that ReportDrugSafetyReportSEBLReqABCSImpl and

ReportDrugSafetyReportArgusProvABCSImpl are updated with customizations.

(This involves copying the modified custom xsls to the application server's folders $AIA_HOME/services/core/SiebelAECM/RequesterABCS/ReportDrugSafetyReportSEBLReqABCSImpl/xsl and $AIA_HOME/services/core/ArgusSafety/ProviderABCS/ReportDrugSafetyReportArgusProvABCSImpl /xsl correspondingly).

b. Open the file $AIA_HOME/pips/DrugDeviceAESEBLandArgus/DeploymentPlans/DrugDeviceAESEBLandArgusCustomDP.xml

Replace <Deployments> </Deployments> with the following:

<Deployments>

<Composite compositeName="ReportDrugSafetyReportArgusProvABCSImpl"

compositedir="${AIA_HOME}/services/core/ArgusSafety/ProviderABCS/ReportDrugSafetyReportArgusProvABCSImpl" revision="1.0" wlserver="pips.DrugDeviceAESEBLandArgus" action="deploy" overwrite="true"/>

<Composite compositeName="ReportDrugSafetyReportSEBLReqABCSImpl" compositedir="${AIA_HOME}/services/core/SiebelAECM/RequesterABCS/ReportDrugSafetyReportSEBLReqABCSImpl" revision="1.0" wlserver="pips.DrugDeviceAESEBLandArgus" action="deploy" overwrite="true"/>

</Deployments>

Save the file.

c. Set the environment variables by executing "source <AIA_HOME>/aia_instances/<AIA_INSTANCE_NAME>/bin/aiaenv.sh"

d. Run the custom deployment command for deploying customized artifacts

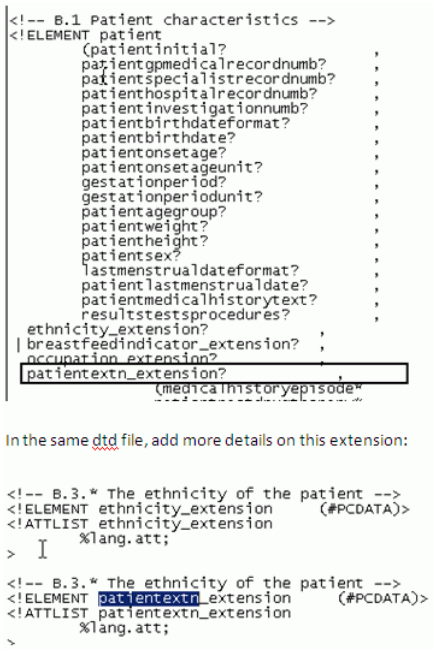
```
ant-f $AIA_HOME/Infrastructure/Install/AID/AIAInstallDriver.xml -DDeploymentPlan=$AIA_HOME/pips/DrugDeviceAESEBLandArgus/DeploymentPlans/DrugDeviceAESEBLandArgusCustomDP.xml -DPropertiesFile=$AIA_INSTANCE/config/AIAInstallProperties.xml -l <location and name where you want the log file written>
```

8. Create Custom Field in Oracle Argus Safety:

a. Modify the DTD file. On Oracle Argus Safety Interchange server, open the <ORACLE_HOME>/Argus/ESMService/DTDFiles/ich-icsr-v2.1-FDA-PIP.DTD.

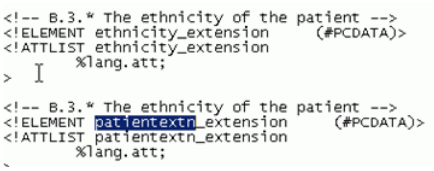
Add an extension field called patientextn_extension under patient as shown below:

Figure 3–12 patientextn_extension Field



In the same DTD file, add more details on this extension:

Figure 3–13 patientextn_extension



Save DTD file and exit. Ensure that you always add the extension elements to end of the sequence of children elements. For example, patientextn_extension is added after occupation_extension. This change in the DTD is equivalent to the change in the xsd in **Step 4.b**: **Modifying Argus xsd**.

b. Add field export and import logic. CFG_E2B table needs to have a row for each DTD_ELEMENT that is imported. The following SQLs are provided as an example for importing patientextn_extension. The values in the SQLs need to be modified depending on type of the field that is being added.

Login to SQL script execution tool such as SQL Developer or SQL Plus using ESM schema owner credentials for your Argus Safety database.

Execute the following SQLs:

1. insert into cfg_e2b (PROFILE, DTD_ELEMENT, HIE_LEVEL, DTD_TYPE, DELETED, DTD_LENGTH, LANGUAGE,MANDATORY_DTD_ELEMENT, MANDATORY, ORDER_OF_EXECUTION, AE_SELECT_STMT_ELEMENT_ASSOC, AE_SELECT_STMT_COL_POSITION, PARENT_ELEMENT, DATA_ELEMENT, AE_CASE_FORM_GUI, DTD_ELEMENT_TITLE, REPEATABLE, ALLOW_USER_PROC, AE_SELECT_STMT, AE_USER_PROC, DTD_ELEMENT_TYPE, UPDATE_FOR_NULLIFICATION, CHILD_ONLY_SQL, ALWAYS_IMPORT, DTD_ELEMENT_TITLE_J, ALLOW_JAPANESE_CHARACTERS, FIELD_LOCATION, FIELD_LABEL)

values ('ICH-ICSR V2.1 MESSAGE TEMPLATE - FDA PIP', 'PATIENTEXTN_EXTENSION', 7, 'DEFINED', null, 40, 'en', '', '', 121.4, 'PATIENT', 20, 'PATIENT', '', 'Case Form / Patient / Patient Information / patientextn', 'The patientextn of the patient', -1, null, null, null, 1, null, null, 0, '', null, '', '');

2. Update CFG_E2B

   Set AE_USER_PROC =

   'DECLARE'|| CHR(10) ||

   ' v_xml varchar2(32767);'|| CHR(10) ||

   ' l_return number := 0;'|| CHR(10) ||

   ' v_error_message varchar2(4000);'|| CHR(10) ||

   ' l_value varchar2(25);'|| CHR(10) ||

   ' v_prod_name varchar2(32767);'|| CHR(10) ||

   ' lb_return boolean := FALSE;'|| CHR(10) ||

   ' v_dtd_elements esm_imp.vtab; '|| CHR(10) ||

   'BEGIN'|| CHR(10) ||

   ' v_xml := trim(ESM_IMP.F_READ_EXTENSION(:REPORT_ID,:DTD_ELEMENT));'|| CHR(10) ||

   ' if v_xml is not null then'|| CHR(10) ||

   ' v_dtd_elements(1) := ''DRUGCHARACTERIZATION'';'|| CHR(10) ||

   ' v_dtd_elements(2) := ''MEDICINALPRODUCT'';'|| CHR(10) ||

   ' v_dtd_elements(3) := ''DRUGDOSAGEFORM'';'|| CHR(10) ||

   ' v_prod_name := ESM_IMP.F_READ(:REPORT_ID,''MEDICINALPRODUCT'');'|| CHR(10) ||

   ' if v_prod_name is null then'|| CHR(10) ||

   ' v_dtd_elements(2) := ''ACTIVESUBSTANCENAME'';'|| CHR(10) ||

   ' end if;'|| CHR(10) ||

   ' lb_return := ESM_IMP.F_CHECK_AND_BUILD_KEY(:REPORT_ID,:PARENT_ELEMENT,v_dtd_elements); '|| CHR(10) ||

   ' l_value := v_xml;'|| CHR(10) ||

   ' l_return := ESM_IMP.F_WRITE(:REPORT_ID,:PARENT_ELEMENT,:DTD_ELEMENT,:PROFILE,''CASE_PAT_INFO'',''UD_TEXT_1'',l_value);'|| CHR(10) ||

   ' endif;'|| CHR(10) ||

   'END;'

   WHERE DTD_ELEMENT = 'PATIENTEXTN_EXTENSION' AND PROFILE = 'ICH-ICSR V2.1 MESSAGE TEMPLATE - FDA PIP';

3. Commit;

   Note that, the above SQLs need to be customized based on the type of the field. For example, if a user defined device extension is being added, then the table name parameter in SQL# 2 will be

"ESM_IMP.F_WRITE(:REPORT_ID,:PARENT_ELEMENT,:DTD_ELEMENT,:PROFILE,''CASE_PROD_DEVICE'',''UD_TEXT_1'',l_value);"

9. **Test the Custom Extension**:

   a. Login to Siebel AECM, Create a Product Issue and populate the custom extension field that was added. Click **Send to Safety** button. A message is sent to integration pack JMS queue.

   b. On integration pack's Enterprise Manager Console, select the instance that was triggered. Verify the input payload at the ReportDrugSafetyReportSEBLReqABCSImpl. Ensure Siebel AECM ABM has the custom field.

   c. Verify the input payload (EBM) ReportDrugSafetyReportEBS. Ensure that EBM custom field **PatientExtension** has the appropriate value populated.

   d. Verify the output of ReportDrugSafetyReportArgusProvABCSImpl. Ensure that Argus E2B file has the custom field at the appropriate place.

   e. Login to Oracle Argus Safety. Verify that the E2B case has the custom extension field with correct value.

The following DVMs are added to the Foundation Pack to support additional fields you might send to Oracle Argus Safety. If you have added any of these fields to Siebel AECM, you should use the DVM when populating the EBM.

DVM Type	DVM Column Name	Comments
DRUGSAFETYREPORT_OBSERVESTUDYTYPE	COMMON, ARGUS_01	This maps to Case Form / General / Study Information / Observe Study Type in Oracle Argus Safety.
OCCUPATION	COMMON, ARGUS_01	This maps to Case Form / Patient / Patient Details/Occupation in Oracle Argus Safety.
RACE	COMMON, ARGUS_01	This maps to Case Form / Patient / Patient Information / Ethnicity in Oracle Argus Safety.
DRUGSAFETYREPORT_AGEGROUP	COMMON, ARGUS_01	This maps to Case Form / Patient / Patient Information / Age Group in Oracle Argus Safety.
DRUGSAFETYREPORT_DRUGACTIONTAKEN	COMMON, ARGUS_01	This maps to Case Form / Products / Product Details / Action Taken in Oracle Argus Safety.
DRUGSAFETYREPORT_DRUGFORMULATION	COMMON, ARGUS_01	This maps to Case Form / Products / Product Information / Formulation in Oracle Argus Safety.

3.12 Configuring Multiple Siebel AECM Servers to a Single Instance of SOA Server

If you have more than one server for Siebel AECM, you can use the steps provided in this section for configuring multiple AECM environments to send product issues from multiple Siebel AECM servers to Oracle Argus Safety.

Perform the following steps to configure additional Siebel AECM instances:

1. Update AIA System Registry:
   a. Navigate to the AIA Console URL: http://<server name>:<portnumber>/AIA.
   b. Log in with server admin user name and password.
   c. Navigate to **Setup**, then to **Systems**.
   d. Click **Create** to create an additional row for each instance of Siebel AECM to be configured. As an example, add one extra instance of AECM named SEBLCLIN_02 (assuming it as original instance of AECM is SEBLCLIN_01).
   e. Enter values in the following mandatory fields:

Field	Description
Internal ID	This is the Siebel Enterprise Server Name (For example: siebel). Contact the Siebel administrator for getting this value.
System Code	A unique value to identify this system. This is used for DVM and XREF lookups as well as routing.
System Type	A unique value to identify the system type.

2. Modify the DVM and Cross Reference (XRef) entries. The Domain Value Maps (DVMs) are stored under $AIA_HOME/AIAMetaData/**dvm** folder. Cross References (XRefs) are stored under $AIA_HOME/AIAMetaData/**xref** folder. You need to modify the following DVMs and XRefs:
   - DRUGSAFETYREPORT_CASEID.xref
   - STATE.dvm
   - COUNTRY.dvm
   - DRUGSAFETYREPORT_REPORTTYPE.dvm
   - DRUGSAFETYREPORT_QUALIFICATIONCODE.dvm
   - GENDER.dvm
   - DRUGSAFETYREPORT_ADVERSEEVENTREPORTEDOUTCOME.dvm
   - DRUGSAFETYREPORT_DEVICEREMEDIALACTION.dvm
   - UNIT_OF_MEASURE.dvm
   - DRUGSAFETYREPORT_ADVERSEEVENTSEVERITY.dvm
   - DRUGSAFETYREPORT_DRUGADMINISTRATIONROUTE.dvm
   - DRUGSAFETYREPORT_DEVICEUSE.dvm
   - DRUGSAFETYREPORT_DEVICEOPERATOR.dvm
   - DRUGSAFETYREPORT_OCCURRENCELOCATION.dvm

- DRUGSAFETYREPORT_STATUS.dvm

In the XRefs add an additional column for SEBLCLIN_02 (second instance of AECM and any other AECM instances, as you may require). In DVMs add additional Column Name and alter to add the column values appropriately.

The following code is an example for COUNTRY.dvm and DRUGSAFETYREPORT_CASEID.xref. You may need to modify certain values as per your target environment.

COUNTRY.dvm

```
<?xmlversion='1.0' encoding='UTF-8'?>
<dvm name="COUNTRY" xmlns="http://xmlns.oracle.com/dvm">
<description>Country LoVs</description>
<columns>
<column name="COMMON"/>
<column name="SEBLCLIN_01"/>
<column name="SEBLCLIN_02"/>
<column name="OC_01"/>
<column name="SEBL_01"/>
<column name="ARGUS_01"/>
</columns>
<rows>
<row>
<cell>United States</cell>
<cell>USA</cell>
<cell>USA</cell>
<cell/>
<cell/>
<cell>US</cell>
</row>
</rows>
</dvm>
```

DRUGSAFETYREPORT_CASEID.xref

```
<?xmlversion="1.0" encoding="UTF-8" ?>
<!--Upgraded by Xref Upgrade Utility 1.0 -->
<xref xmlns="http://xmlns.oracle.com/xref">
<table name="DRUGSAFETYREPORT_CASEID">
<description></description>
<columns>
<column name="COMMON"/>
```

<column name="SEBLCLIN_01"/>

<column name="OC_01"/>

<column name="SEBLCLIN_02"/>

</columns>

</table>

</xref>

For more information about how to update MDS and upload updated DVMs and XRefs to MDS, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1*, "Building AIA Integration Flows", "Updating MDS".

Note that, any change made to the MDS will overwrite the files. However, the data in the Xref_data table will not be affected.

You can verify DVMs and XRefs by accessing these files in MDS. You can also verify DVMs in SOA Composer.

3. Update AIAInstallProperties.xml.

   a. Update the AIAInstallProperties.xml file in the $AIA_INSTANCE_HOME/config to add the following SEBLCLIN_02 database entry as a child node of /properties/participatingapplications.

      Remember to replace the variables in the curly braces {} with the actual values.

      <sc>

      <server>

      <internal>

      <id>{SID}</id>

      </internal>

      <eai>

      <username>{username}</username>

      <password>{password}/password>

      </eai>

      <host>{host }</host>

      <port>{port}</port>

      <version>8.1.1.6</version>

      <Language>enu</Language>

      <InternetProtocol>http://</InternetProtocol>

      </server>

      </sc>

   b. Encrypt the AIAInstallProperties password by running the following commands:

   - source `$AIA_INSTANCE_HOME/bin/aiaenv.sh`
   - cd `$AIA_HOME/util`

- `ant -f updateStore.xml addToStore -DAdminUsername=< weblogic Admin username> -DAdminPassword=<weblogic admin password>`
- A popup window is displayed. Enter values for:
  - **Username**: Admin user username for SEBLCLIN_02 URL
  - **Password**: Admin user password for SEBLCLIN_02 URL
  - **Path**: Xpath of the SEBLCLIN_02 database password field in the AIAInstallProperties.xml.

  For example: /properties/participatingapplications/sc2/db/password

  This updates the encrypted password in the MDS as well as in the $AIA_INSTANCE_HOME/config file. Inspect the AIAInstallProperties.xml file in $AIA_INSTANCE/config to verify that the password for SEBLCLIN_02 database is encrypted.

4. Modify the service configurations. Navigate to ${AIA_HOME}/aia_instances/inst1/AIAMetaData/config and open AIAConfigurationProperties.xml for modification.

   Under Session Pool Manager properties, add the following properties:

   <!--=== Specific values for Siebel Clinical Trial Management SEBLCLIN_02 === -->

   <Property name="SEBLCLIN_02.EndpointURI">https://<siebelserverhostname>:<siebelserverpost>/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&WSSOAP=1</Property>

   <Property name="SEBLCLIN_02.UserId"><username></Property>

   <Property name="SEBLCLIN_02.Password">participatingapplications.sc2.server.eai.password</Property>

   <Property name=" SEBLCLIN_02.PredictExpiration_Idle">780000</Property>

   <Property name="SEBLCLIN_02.PredictExpiration_Age">82800000</Property>

   <Property name="SEBLCLIN_02.InvalidSessionErrorCodes">10944642|SBL-BPR-00162|SBL-DAT-00175|11338608|SBL-UIF-00880</Property>

   <Property name="SEBLCLIN_02.ClassName">oracle.apps.aia.core.sessionpool.CRMSiebelSession</Property>

   Upload the AIAConfigurationProperties file to MDS using the following steps:

   a. Modify the AIAConfigurationProperties.xml at $AIA_INSTANCE_HOME/AIAMetaData/config

   b. Edit $AIA_INSTANCE_HOME/config/UPdateMetaDataDP.xml to include AIAConfigurationProperties.xml. Following is a sample UpdateMetaDataDP.xml:

      <?xmlversion="1.0" standalone="yes"?>

      <DeploymentPlan component="Metadata" version="3.0">

      <Configurations>

      <UpdateMetadata wlserver="fp" >

```
<fileset dir="${AIA_INSTANCE}/AIAMetaData">
<include name="config/AIAConfigurationProperties.xml"/>
</fileset>
</UpdateMetadata>
</Configurations>
</DeploymentPlan>
```

c. Save the changes. Run the following commands:

– source `$AIA_INSTANCE_HOME/bin/aiaenv.sh`

– cd `$AIA_HOME/Infrastructure/Install/config`

– `ant -f UpdateMetaData.xml`

5. Set up a new reporting destination in Argus Console. Enter the following values for the fields:

   - **Agency Information** Tab:

     Agency Name - AECM_ARGUS_INTEGRATION2

   - **Local Company Contact** Tab:

     Company Name - INTEGRATIONS2

   - **EDI** Tab:

     Agency Identifier - SEBLCLIN_02

For more information about the fields and the values that you need to enter in **Agency Information**, **Local Company**, and **EDI** tab; apart from the one mentioned above, refer *Oracle® Device and Drug Adverse Event Data Integration Pack for Siebel Adverse Event Complaint Management and Oracle Argus Safety, Installation Guide, Release 11.1*, "Configuring Argus for Using Extension Profile".

3.13 Monitoring Instances on SOA Server

You can monitor the SOA composite instances at run time using the composite sensors. Composite sensors enable you to search for instances using certain parameters. For example, a composite instance fails at some point in the flow. As an admin user, you can search for that failed composite instance on Enterprise Manager, using a pre-defined composite sensor as a search parameter. For example, you can use ProductIssueNumberSensor to search for all composite instances that have a given Product Issue Number.

To perform a composite instance search follow these steps:

1. Navigate to the component. For example: ReportDrugSafetyReportSEBLReqABCSImpl
2. Select the **Instances** tab and click **Add Fields** button in the Search parameter selection area. Using **Add Fields** button, you can add additional search parameters such as Product Issue Number or Case ID.

   The following composite sensors are available to monitor composite instances during the runtime:

Field Name	Component Name	Composite Sensor Name
Product Issue Number	ReportDrugSafetyReportSEBLReqABCSImpl	ProductIssueNumberSensor
Product Issue Number	ReportDrugSafetyReportResponseSEBLProvABCSImpl	ProductIssueNumberSensor
AE Case ID	ReportDrugSafetyReportResponseSEBLProvABCSImpl	AECaseIdSensor

The screen below shows composite instance search feature, where all composite instances matching a specific Product Issue Number can be searched using **ProductIssueNumberSensor**. Thus, you can easily find the specific composite instances.

Figure 3–14 Composite Instance Search