Oracle® Study, Subject, and Visit Synchronization Integration Pack for Siebel Clinical and Oracle Clinical

Implementation Guide

Release 11.1

E36156-01

August 2013

Oracle Study, Subject, and Visit Synchronization Integration Pack for Siebel Clinical and Oracle Clinical Implementation Guide, Release 11.1

E36156-01

Contents

4 Automating the Update of Activity Completion Date and Status Based on Data Entered in Oracle Clinical or Oracle Clinical Remote Data Capture

Part II Implementing the Delivered Integrations

5 Implementing the PIP

6 Configuring Multiple Oracle Clinical Database Instances

7 Using the Resubmission Utility

Index

Preface

Welcome to the Oracle Study, Subject and Visit Synchronization Integration Pack for Siebel Clinical and Oracle Clinical 11.1.

Audience

This document is intended for all Oracle Study, Subject and Visit Synchronization Integration Pack for Siebel Clinical (SC) and Oracle Clinical (OC) for Application Integration Architecture (AIA) users.

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.

Oracle Application Integration Architecture Guides

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

- *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)*

Related Documents

The following resources are also available:

Resource	Location
Oracle Application Integration Architecture: Product-to-Guide Index	My Oracle Support: `https://support.oracle.com/`
Known Issues and Workarounds	My Oracle Support: `https://support.oracle.com/`
Release Notes	Oracle Technology Network: `http://www.oracle.com/technology/`
Oracle Study, Subject, and Visit Synchronization Integration Pack for Siebel Clinical and Oracle Clinical Installation Guide	Oracle Technology Network: `http://www.oracle.com/technology/`
Documentation updates	My Oracle Support: `https://support.oracle.com/`

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Understanding the Delivered Integrations

Part I contains the following chapters:

1

About Study Subject and Visit Synch: Siebel Clinical – Oracle Clinical PIP

This chapter discusses:

- Section 1.1, "Key Benefits of the PIP"
- Section 1.2, "Common Terms Used in this Guide"
- Section 1.3, "Business Process Flows"
- Section 1.4, "Solution Assumptions and Constraints"

1.1 Key Benefits of the PIP

The Study, Subject and Visit Synch: Siebel Clinical – Oracle Clinical PIP integrates two complementary clinical technology applications.

This integration includes the following processes:

- Study Site Coordination (including address, investigator, and so on)
- Study Subject Coordination
- Activity Completion Integration

The Study, Subject and Visit Synch: Siebel Clinical – Oracle Clinical PIP enables timely exchange of data between Oracle Clinical and Siebel Clinical. Information sent from Siebel Clinical, (for example, investigator and site details) automates the creation of a study site and eliminates the manual creation of a Site, Investigator, and Study Site. Information sent from Oracle Clinical, such as patient numbers and information related to the completion of Oracle Clinical items (Visits, DCIs, DCMs, Questions), facilitates efficient and accurate patient counts and activity completion tracking in Siebel Clinical. The automation of these processes ultimately provides the means to effect timely and accurate Investigator payments. It also aids accurate tracking and response to issues related to site performance and protocol adherence.

1.1.1 General Business Process

The general business process is as follows:

- In Siebel Clinical, sites are created at a protocol level. Siebel Clinical sends protocol site information to Oracle Clinical, including address, investigator name, and site number. This information is used to automatically create or update Investigators, Sites, and Study Sites in Oracle Clinical.
- When a patient has enrolled in a study or commences the screening visit process, Oracle Clinical sends information to Siebel Clinical about the patient to create a

subject for the appropriate protocol site in Siebel Clinical. Subsequent updates to the patient are transmitted to Siebel Clinical, which then updates the patient in that system.

- Siebel Clinical can schedule activities to be completed during patient visits, including the designation of a patient visit as an individual activity. Certain data collection tasks in Oracle Clinical may correspond to completion of activities in Siebel Clinical. For example, an ECG DCI in Oracle Clinical will correspond to a visit activity in Siebel Clinical. The collection or completion of a visit or specific sets of data within a visit in Oracle Clinical can be used to update the status of an activity in Siebel Clinical.

1.2 Common Terms Used in this Guide

Table 1–1 describes some of the common terms used in this guide:

Table 1–1 Common Terms Used in this Guide

Siebel Clinical Term	Oracle Clinical Term	Definition
Protocol	Study	A document that describes the objective(s), design, methodology, statistical considerations, and organization of a trial. Within Siebel Clinical, protocol is synonymous with study.
Account Address	Site	The organization with which the investigator on the study is associated. This entity is not associated with a Study. An account is not equivalent to a site in Oracle Clinical. In Siebel Clinical, the account includes all the locations of an organization. In Oracle Clinical, a site is a particular location where a clinical study can be conducted. However, in Oracle Clinical, you cannot include the same site in the study with different principal investigators. In Siebel Clinical, an account can belong to multiple protocol sites in a protocol with a different principal investigator assigned to each one.
Principal Investigator (PI)	Investigator	The physician or clinician responsible for conducting the trial.
None	Patient Position	An identifier in Oracle Clinical that is a placeholder for a real participant in a clinical study. Patient Positions are created, based on the target enrollment in a study and assigned to a study site. As each subject is enrolled or data is collected for that subject, a patient position will be assigned.
Subject	Patient	Persons recruited by investigators that participate in the trial at a study site.
Subject Visit Template	DCI Book	The expected events (visit and activities or procedures) that are conducted throughout the course of the trial as specified in the study protocol.
Subject Visit Schedule	None	The planned schedule of events for a particular subject at a site based upon the Subject Visit Template. Once the actual events occur the information in the Schedule is updated.
None	Completion Criteria	A set of parameters, defined in Oracle Clinical and based on visit, clinical planned event, received DCI, received DCM, or question(s) response(s), which can be used to assign a completion date to a Siebel Clinical activity.
Activity	None	Required procedures or tasks in the visit schedule in Siebel Clinical.

1.3 Business Process Flows

This section provides an overview and discusses the following process flows:

- Section 1.3.1, "Create or Update Study Sites"
- Section 1.3.2, "Create or Update Study Subjects"
- Section 1.3.3, "Update Visit or Activity Completion Date and Status"

1.3.1 Create or Update Study Sites

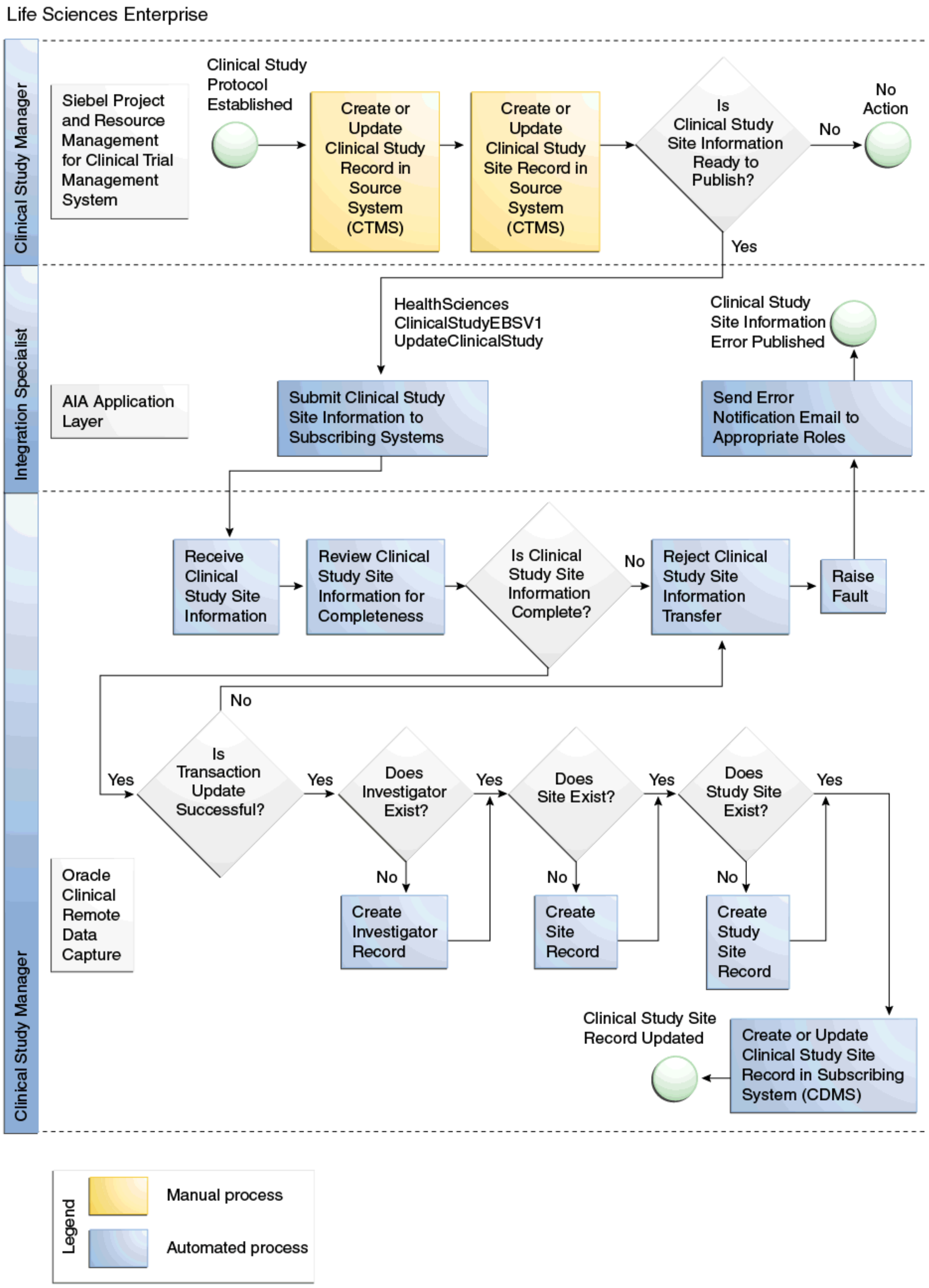
During the study set up process, protocols are created in Siebel Clinical under a Program. Clinical Studies are defined in Oracle Clinical under a Program and Project. To utilize this integration, you must manually enter Study code from Oracle Clinical into the CDMS Study ID field of the corresponding protocol in Siebel Clinical. Once the CDMS study ID is entered, the Synchronize Activities Sites flag must be selected.

When a protocol site is created in Siebel Clinical and when it is ready to be shared with Oracle Clinical, the *Activate for Synchronization* check box is selected, which creates a study site in Oracle Clinical. All future updates to the protocol site will be sent to Oracle Clinical. If the principal investigator for the protocol site does not exist in Oracle Clinical, the system will create one.

The combination of the primary address and account assigned for the protocol site are considered as a site in Oracle Clinical. If this combination does not exist in Oracle Clinical, a new site will be created.

Figure 1–1 shows the Create or Update Study Sites flow:

Figure 1–1 Create or Update Study Sites Flow Diagram


Life Sciences Enterprise
Clinical Study Manager
Siebel Project and Resource Management for Clinical Trial Management System
Clinical Study Protocol Established
Create or Update Clinical Study Record in Source System (CTMS)
Create or Update Clinical Study Site Record in Source System (CTMS)
Is Clinical Study Site Information Ready to Publish?
No
No Action
Yes
Integration Specialist
AIA Application Layer
HealthSciences ClinicalStudyEBSV1 UpdateClinicalStudy
Submit Clinical Study Site Information to Subscribing Systems
Clinical Study Site Information Error Published
Send Error Notification Email to Appropriate Roles
Clinical Study Manager
Receive Clinical Study Site Information
Review Clinical Study Site Information for Completeness
Is Clinical Study Site Information Complete?
No
Reject Clinical Study Site Information Transfer
Raise Fault
No
Yes
Is Transaction Update Successful?
Yes
Does Investigator Exist?
Yes
Does Site Exist?
Yes
Does Study Site Exist?
Yes
Oracle Clinical Remote Data Capture
No
Create Investigator Record
No
Create Site Record
No
Create Study Site Record
Clinical Study Site Record Updated
Create or Update Clinical Study Site Record in Subscribing System (CDMS)
Legend
Manual process
Automated process

1.3.2 Create or Update Study Subjects

In Oracle Clinical, patient positions are created for the target enrollment for a study and assigned to study sites based on the target enrollment for the study site. The enrollment date for a patient position indicates that a patient has been enrolled in the study and assigned this enrollment ID. This creates a subject for the appropriate protocol site in Siebel Clinical and the subject will be screened and enrolled against the active Subject Visit Template.

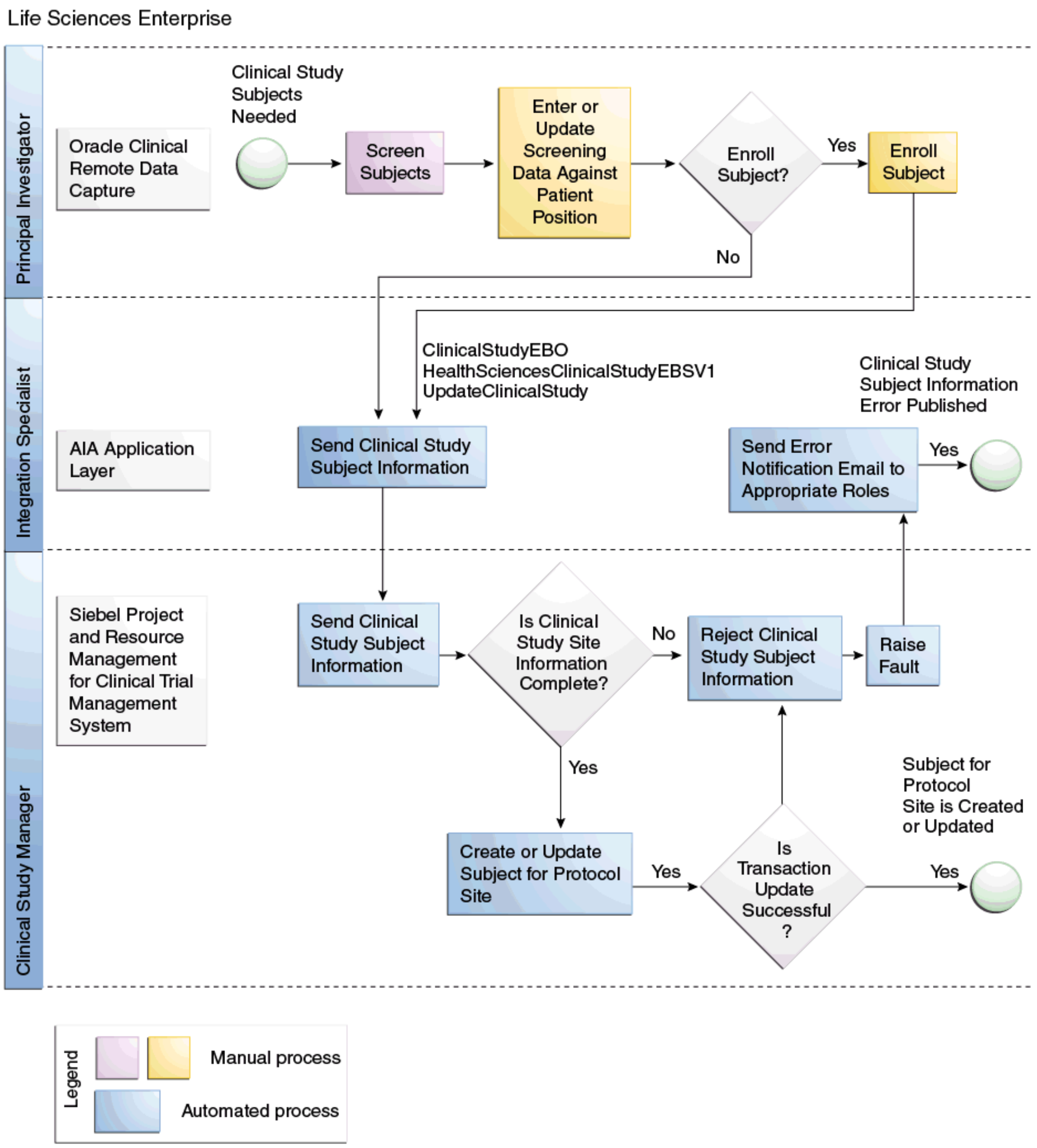
In Oracle Clinical or Oracle Clinical Remote Data Capture (OCRDC), if data is entered for a patient position and no enrollment date has been specified, it is assumed that the patient is undergoing the screening process assigned to this screening ID. This creates a subject for the appropriate protocol site in Siebel Clinical and the subject will be screened against the active Subject Visit Template.

Siebel Clinical requires the subject's initials and date of birth to create a subject. These are not required fields in Oracle Clinical. Therefore, if the patient's initials are not entered in Oracle Clinical, the enrollment ID populates the initials in Siebel Clinical. If the date of birth of the patient was not entered in Oracle Clinical, the default value of Jan 1, 1800 is used in Siebel Clinical. The user should recognize this date as invalid. You can enter the correct data manually in Siebel Clinical.

To assign an Enrollment Visit Template to a subject in Siebel Clinical, the date the informed consent was signed is required. Informed Consent Signature date is an optional field in Oracle Clinical, and if it is not entered Siebel Clinical assigns a default value of Jan 1, 1900. This will adversely impact the dates in the Visit Schedule for the subject and you must manually correct it in Siebel Clinical.

To assign a Screening Visit Template to a subject in Siebel Clinical, the screen date is required. However, this is not a required field in Oracle Clinical. Therefore, if this date is not entered in Oracle Clinical, the subject's birth date is used. This will adversely impact the dates in the Visit Schedule for the subject and you must manually correct it in Siebel Clinical.

Figure 1–2 Create or Update Study Subjects Flow Diagram


Life Sciences Enterprise
Principal Investigator
Integration Specialist
Clinical Study Manager
Oracle Clinical Remote Data Capture
Clinical Study Subjects Needed
Screen Subjects
Enter or Update Screening Data Against Patient Position
Enroll Subject?
Yes
No
Enroll Subject
AIA Application Layer
ClinicalStudyEBO
HealthSciencesClinicalStudyEBSV1
UpdateClinicalStudy
Send Clinical Study Subject Information
Clinical Study Subject Information Error Published
Send Error Notification Email to Appropriate Roles
Yes
Siebel Project and Resource Management for Clinical Trial Management System
Send Clinical Study Subject Information
Is Clinical Study Site Information Complete?
No
Reject Clinical Study Subject Information
Raise Fault
Yes
Subject for Protocol Site is Created or Updated
Create or Update Subject for Protocol Site
Yes
Is Transaction Update Successful ?
Yes
Legend
Manual process
Automated process

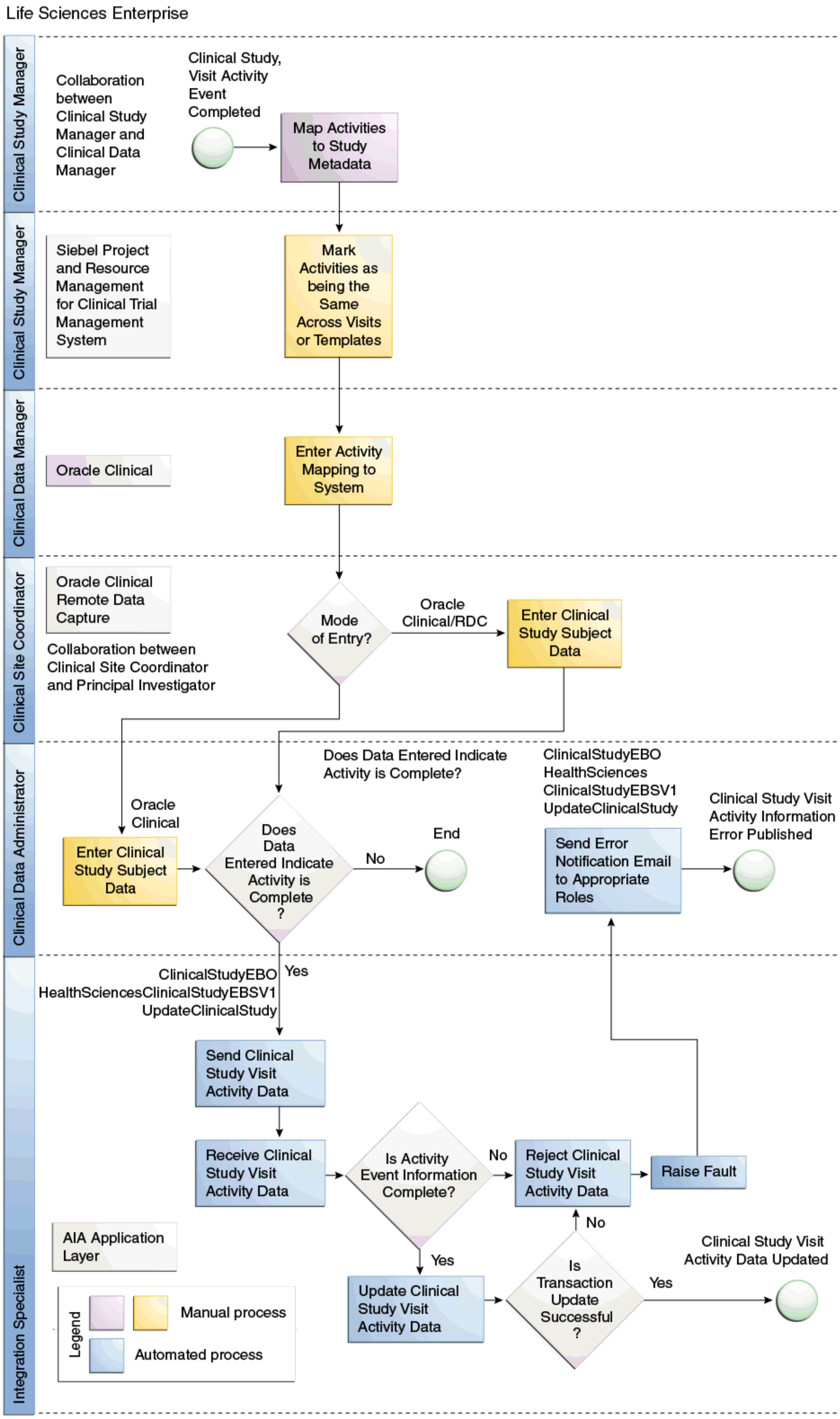
1.3.3 Update Visit or Activity Completion Date and Status

This workflow updates the completion date of Subject visits and Subject visit activities based on data entered into Oracle Clinical or OCRDC. To initiate this, the Study Manager and Data Manager should collaborate and determine what data must be completed in OCRDC for a visit or activity to be considered complete in Siebel Clinical. They must also decide how to determine the visit completion date in Oracle Clinical. Enhancements have been made to Siebel Clinical to indicate that multiple activities or visits share the same completion criteria. Enhancements have been made to Oracle Clinical to define the activity completion criteria in terms of Oracle Clinical data collection objects such as Data Collection Instruments (DCI), Data Collection Modules (DCM), or DCM questions.

After the Activity Completion criteria are defined, you can schedule a batch job to initiate a process that analyzes patient data to evaluate whether any visits or visit activities have been completed since the last time the process ran. If any activity completion criteria has been added or changed, all patient data will be analyzed.

This batch job sends messages regarding the completion date of either activities or visits or both, which updates the completion date and status of the associated activities, or visits for the appropriate Subject Visit schedule in Siebel Clinical.

Figure 1–3 Update Visit or Activity Completion Date and Status Flow Diagram


Life Sciences Enterprise
Clinical Study Manager
Collaboration between Clinical Study Manager and Clinical Data Manager
Clinical Study, Visit Activity Event Completed
Map Activities to Study Metadata
Clinical Study Manager
Siebel Project and Resource Management for Clinical Trial Management System
Mark Activities as being the Same Across Visits or Templates
Clinical Data Manager
Oracle Clinical
Enter Activity Mapping to System
Clinical Site Coordinator
Oracle Clinical Remote Data Capture
Collaboration between Clinical Site Coordinator and Principal Investigator
Mode of Entry?
Oracle Clinical/RDC
Enter Clinical Study Subject Data
Clinical Data Administrator
Does Data Entered Indicate Activity is Complete?
ClinicalStudyEBO HealthSciences ClinicalStudyEBSV1 UpdateClinicalStudy
Clinical Study Visit Activity Information Error Published
Oracle Clinical
Enter Clinical Study Subject Data
Does Data Entered Indicate Activity is Complete ?
No
End
Send Error Notification Email to Appropriate Roles
Integration Specialist
ClinicalStudyEBO HealthSciencesClinicalStudyEBSV1 UpdateClinicalStudy
Yes
Send Clinical Study Visit Activity Data
Receive Clinical Study Visit Activity Data
Is Activity Event Information Complete?
No
Reject Clinical Study Visit Activity Data
Raise Fault
No
Yes
AIA Application Layer
Legend
Manual process
Automated process
Update Clinical Study Visit Activity Data
Is Transaction Update Successful ?
Yes
Clinical Study Visit Activity Data Updated

1.4 Solution Assumptions and Constraints

These are the solution assumptions and constraints:

1.4.1 Assumptions

The following are the assumptions:

- For the Clinical Study Subject workflow integration to proceed, you must populate the enrollment date for a patient in Oracle Clinical. This can be done in various ways through the Patient Positions form. The enrollment dates can be loaded in Oracle Clinical data entry, by batch loading enrollment dates or using Patient Synchronization in a derivation procedure to populate the date from a CRF.
- Investigators defined in Oracle Clinical may be used on any Clinical Study. The integration assumes that reassigning the principal investigator for a protocol site will not cause the investigator to be deleted from Oracle Clinical.
- When a protocol site is deleted in Siebel Clinical, you may not want to delete the corresponding study site in Oracle Clinical. Hence, this is not currently part of the integration.
- As patient data can be deleted and re-entered in Oracle Clinical, it will not be desirable to remove the subject in Siebel Clinical when the enrollment date was left blank or patient data was removed.

1.4.2 Constraints

The following are the constraints:

- Protocols are created in Siebel Clinical under a Program, and Clinical Studies are created in Oracle Clinical under a Program and Project. Therefore, mapping a Clinical study to a Protocol is not an automatic process and must be done manually.
- The investigator address is not required in Oracle Clinical. The synchronization of investigators is a separate business process and the investigator address will not be populated in Oracle Clinical by this integration.
- Unplanned visits in Oracle Clinical will not be associated with Siebel Clinical activities. Since no mapping will exist, Oracle Clinical will not be able to determine whether an unplanned activity has been completed or not.
- Due to size limitations for Study Site code in Oracle Clinical, the Protocol Site Number cannot be longer than 10 characters. Oracle recommends that you incorporate this length restriction in your Siebel Clinical environment to avoid errors.
- The changes to the investigator data (for example last name or phone number) will not be propagated to Oracle Clinical until some other data in the protocol site is changed or the investigator is assigned to another protocol site that is synchronized.
- To create a site in Oracle Clinical, you need to specify the value for the State to be defined as a region within the region you specify for the Country. This restriction does not exist in Oracle Clinical User Interface.

2 Synchronizing Clinical Study Sites

This chapter provides an overview of the synchronization of Clinical Study Sites between Siebel Clinical and Oracle Clinical and discusses:

- Section 2.1, "Overview"
- Section 2.2, "Study Site Integration Flow"
- Section 2.3, "Siebel Clinical Interfaces"
- Section 2.4, "Oracle Clinical Interfaces"
- Section 2.5, "Industry AIA Components"
- Section 2.6, "Integration Services"

2.1 Overview

Study site information is entered as a Protocol Site in Siebel Clinical and a Study Site in Oracle Clinical. This integration is unidirectional and happens from Siebel Clinical to Oracle Clinical. Updates to the investigator, site, or study site in Siebel Clinical are synchronized to Oracle Clinical. Updates made in Oracle Clinical for investigator, site, or study site are not synchronized back to Siebel Clinical.

When the Activate for the Synchronization flag is selected for a protocol site in Siebel Clinical, a corresponding study site will be created in Oracle Clinical. If the principal investigator responsible for the protocol site does not exist in Oracle Clinical, one will be created. The combination of the account and the primary address for the Protocol Site is considered as a Site in Oracle Clinical. If this combination does not exist, a new site is created.

Deleting a principal investigator, account, or protocol site in Siebel Clinical will not cause deletion of any objects in Oracle Clinical.

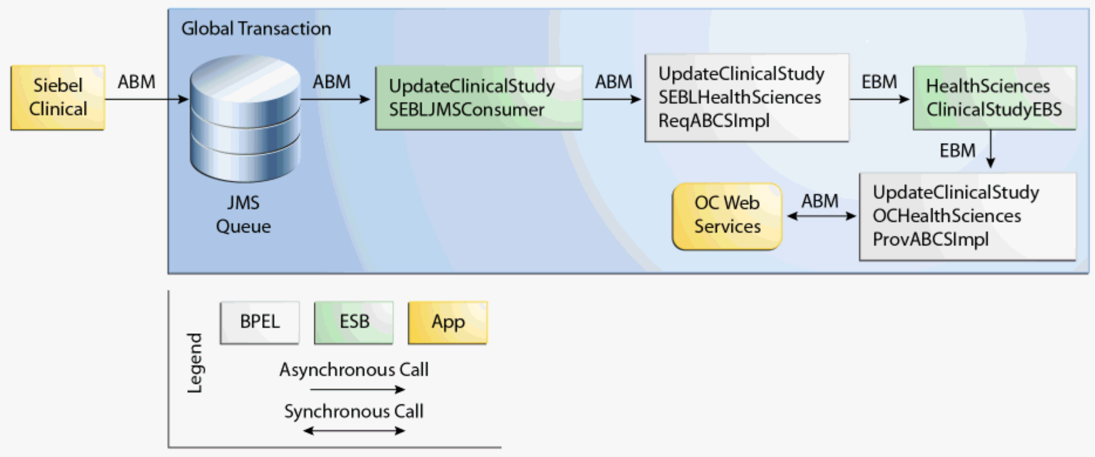
2.2 Study Site Integration Flow

This section provides the activity diagrams and functional descriptions for the Study Site Integration flow.

Flow Diagram

The following diagram shows the Study Site Integration flow:

Figure 2–1 Study Site Integration Flow


Global Transaction
Siebel Clinical
ABM
JMS Queue
ABM
UpdateClinicalStudy SEBLJMSConsumer
ABM
UpdateClinicalStudy SEBLHealthSciences ReqABCSImpl
EBM
HealthSciences ClinicalStudyEBS
EBM
OC Web Services
ABM
UpdateClinicalStudy OCHealthSciences ProvABCSImpl
Legend
BPEL
ESB
App
Asynchronous Call
Synchronous Call

2.2.1 Create Study Site Flow

The following points describe the Create Study Site flow:

1. When the Activate for Synchronization check box is selected for a Protocol Site, Siebel Clinical writes a message Application Business Message (ABM) to the AIA_SiebelClinical_ClinicalStudyJMSQueue Queue.
2. The UpdateClinicalStudySEBLJMSConsumer consumer picks the ABM and passes it to the UpdateClinicalStudySEBLHealthSciencesReqABCSImpl (requester Application Business Connector Service (ABCS)).
3. The requester ABCS transforms the ABM into an UpdateClinicalStudyEBM, populating the ClinicalStudyInvestigator, ClinicalStudySite and ClinicalStudyStudySite cross-reference tables. It retrieves State and Country information for the Site Address from the corresponding DVM tables.
4. The Enterprise Business Message (EBM) is then passed onto the HealthSciencesClinicalStudyEBS (EBS).
5. The EBS will pass the EBM to the UpdateClinicalStudyOCHealthSciencesProvABCSImpl (provider ABCS) based on the configuration routing rules.
6. The provider ABCS receives the EBM and transforms it into either a Create Investigator ABM (if the Investigator is new) or an Update Investigator ABM (if the Investigator already exists). The ABM is then passed to the appropriate Oracle Clinical Web service.
7. If the investigator value in the CreateInvestigator ABM is longer than 10 characters, it is replaced with the sequence number used for the investigator_id field in the ocl_investigators table.
8. If a new Investigator has been created, the ClinicalStudyInvestigator cross-reference table is updated.
9. The provider ABCS then transforms the EBM into either a Create Site ABM (if the Site is new) or an Update Site ABM (if the Site already exists). The Oracle Clinical state and country values are retrieved from the corresponding DVM tables.
10. The ABM is then passed to the appropriate Oracle Clinical Web service.
11. If the site value in the Create Site ABM is longer than 10 characters, it is replaced with the sequence number used for the site_id field in the ocl_sites table.

12. If a new Site has been created, the ClinicalStudySite cross-reference table is updated.
13. The provider ABCS then transforms the EBM into a Create StudySite ABM, which includes Oracle Clinical identifiers for the Site and Investigator (from the corresponding cross-reference tables).
14. A synchronous call to the Oracle Clinical Web service is made and the ID of the newly created StudySite is returned. This can then be added to the ClinicalStudyStudySite cross-reference table.

2.2.2 Update Study Site Flow

The following points describe the Update Study Site flow:

1. When the Activate for Synchronization check box is selected for a Protocol Site and its information is updated (for example address of Protocol Site), Siebel Clinical writes a message (ABM) to the AIA_SiebelClinical_ClinicalStudyJMSQueue queue.
2. The UpdateClinicalStudySEBLJMSConsumer consumer service picks the message and passes it to the UpdateClinicalStudySEBLHealthSciencesReqABCSImpl (requester ABCS).
3. The requester ABCS transforms the ABM into an UpdateClinicalStudy EBM. This EBM contains integration IDs for the Investigator, Site, and StudySite, which are retrieved from the corresponding cross-reference tables. The state and country values in the site address are retrieved from the corresponding DVMs.
4. The EBM is then passed onto the HealthSciencesClinicalStudyEBS (EBS).
5. The EBS passes the EBM to the UpdateClinicalStudyOCHealthSciencesProvABCSImpl (provider ABCS), based on the configuration routing rules.
6. The provider ABCS receives the EBM and transforms it into either a Create Investigator ABM (if the Investigator is new) or an Update Investigator ABM (if the Investigator already exists). The ABM is then passed to the appropriate Oracle Clinical Web service.
7. If the investigator value in the CreateInvestigator ABM is longer than 10 characters, it is replaced with the sequence number used for the investigator_id field in the ocl_investigators table.
8. If a new Investigator has been created, the ClinicalStudyInvestigator cross-reference table is updated.
9. The provider ABCS then transforms the EBM into either a Create Site ABM (if the Site is new) or an Update Site ABM (if the Site already exists). The Oracle Clinical values for the site address state and country are retrieved from the corresponding DVM tables.
10. The ABM is then passed to the appropriate Oracle Clinical Web service.
11. If the site value in the CreateSite ABM is longer than 10 characters, it is replaced with the sequence number used for the site_id field in the ocl_sites table.
12. If a new Site has been created the ClinicalStudySite cross-reference table is updated.
13. The provider ABCS then transforms the EBM into an Update Study Site ABM, which includes Oracle Clinical identifiers for the Site and Investigator (from the corresponding cross-reference tables).

14. A synchronous call to the Oracle Clinical Web service is made.

2.3 Siebel Clinical Interfaces

The following table describes the Siebel Outbound Web services:

Table 2–1 Siebel Outbound Web Services

Name	Schema
ClinicalProtocolSite	Clinical Protocol Site Business Object

2.4 Oracle Clinical Interfaces

The integration uses the following Oracle Clinical artifacts:

Inbound to Oracle Clinical Web Services:

- **SiteService** - This service lets the integration to create, query, and update Sites in Oracle Clinical.
- **InvestigatorService** - This service lets the integration to create, query and update investigators in Oracle Clinical.
- **StudySiteService** - This service lets the integration to create and update Study Sites in Oracle Clinical.

2.5 Industry AIA Components

The integration flow uses the following components:

- ClinicalStudy Enterprise Business Object (EBO)
- HealthSciencesClinicalStudyEBSV1
- UpdateClinicalStudyEBM

The industry EBO and EBM XML Schema Definition (XSD) files can be located by EBO within the following parent folder:

$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/EBO/

The industry EBS Web Services Description Language (WSDL) files can be located by EBO within the following parent folder:

$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/HealthSciences/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in Oracle Enterprise Repository or Oracle Clinical.

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1*, "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 Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack 11g R1*, "Extensibility for AIA Artifacts".

2.6 Integration Services

The following are the services delivered with this integration:

- UpdateClinicalStudySEBLJMSConsumer

  This is a Java Message Service (JMS) consumer service that reads a JMS message from the AIA_SiebelClinical_ClinicalStudyJMSQueue in the Fusion Middleware database.

- UpdateClinicalStudySEBLHealthSciencesReqABCSImpl

  The requester ABCS is responsible for transforming the Siebel Protocol Site ABM into an UpdateClinicalStudy EBM.

- HealthSciencesClinicalStudyEBS

  The EBS service is responsible for passing the message from the Requester ABCS to the Provider ABCS.

- UpdateClinicalStudyOCHealthSciencesProvABCSImpl

  The Provider ABCS is responsible for transforming the UpdateClinicalStudyEBM into the appropriate ABMs to call the appropriate Oracle Clinical Web services.

3

Synchronizing Clinical Study Subject Information

This chapter provides an overview of the synchronization of Clinical Study Subject information between Oracle Clinical and Siebel Clinical and discusses:

3.1 Overview

After enabling patient integration for an Oracle Clinical study, the synchronization process can be initiated. In Oracle Clinical, patient positions are created for the target enrollment of a study and assigned to study sites based on the target enrollment for it. The enrollment date for a patient position indicates that a patient has been enrolled in the study and assigned an Enrollment ID. This creates a subject for the appropriate Protocol Site in Siebel Clinical and the subject is screened and enrolled against the active Subject Visit Template.

If data is collected for a patient position in Oracle Clinical or OCRDC and no enrollment date has been specified, the system assumes this to be a patient undergoing the screening process and assigns a Screening ID. This creates a subject for the appropriate Protocol Site in Siebel Clinical and will be screened against the active Subject Visit Template.

To create a subject in Siebel Clinical, the subject's initials and birth date are required. However, these are not required fields in Oracle Clinical and if the patient's initials are not available in Oracle Clinical, the Enrollment ID is used as the initials in Siebel Clinical. If the patient's birth date is not available in Oracle Clinical, it is set to Jan 1, 1800 in Siebel Clinical. This is an invalid date that can be used to identify subjects whose information has to be updated. You can manually enter the correct data in Siebel Clinical, if desired.

To assign an Enrollment Visit Template to a subject in Siebel Clinical, the date the informed consent was signed on is required. This date is optional in Oracle Clinical and if it is not entered, a default value of Jan 1, 1900 is entered in Siebel Clinical. This will alter the dates in the Visit Schedule for the subject and you may want to manually correct this in Siebel Clinical.

To assign a Screening Visit Template to a subject in Siebel Clinical the screen date is required. This is not a mandatory field in Oracle Clinical and if this date is not available, the subject's birth date is used in Siebel Clinical. This adversely impacts the dates in the subject's Visit Schedule and should be manually corrected in Siebel Clinical.

When a patient is enrolled in a study or begins the screening visit process, Oracle Clinical sends patient information to Siebel Clinical so that a subject may be created for the appropriate Protocol Site in Siebel Clinical. Oracle Clinical transfers the Study ID and Patient ID, which identify the patient along with the following information:

- Patient_Position_id
- Initials
- Sex
- Birth Date
- Informed Consent Signature Date
- Death Date
- StudySite the patient is assigned to
- Start Date of the patient assignment
- DCI Book assigned to the patient
- Frozen Flag
- Include in Efficacy Flag
- Include in Safety Analysis Flag
- Position Type (Screening, Enrollment, Replacement)
- Patient Dropped Flag
- Early Termination Flag
- Enrollment Date
- Exclude from Efficacy Reason
- Inclusion/Exclusion Date
- Exclude from Safety Reason
- Patient Reference
- Randomization Date
- Last Pregnancy Date
- First Screening Date
- Termination Date
- Patient Status History

This process flow is triggered by one of the following events:

- Specifying an enrollment date for a patient position in Oracle Clinical
- Entering the first CRF for a patient who does not have an enrollment date
- Updating information in any of the fields listed for an enrolled patient or a patient with collected data.

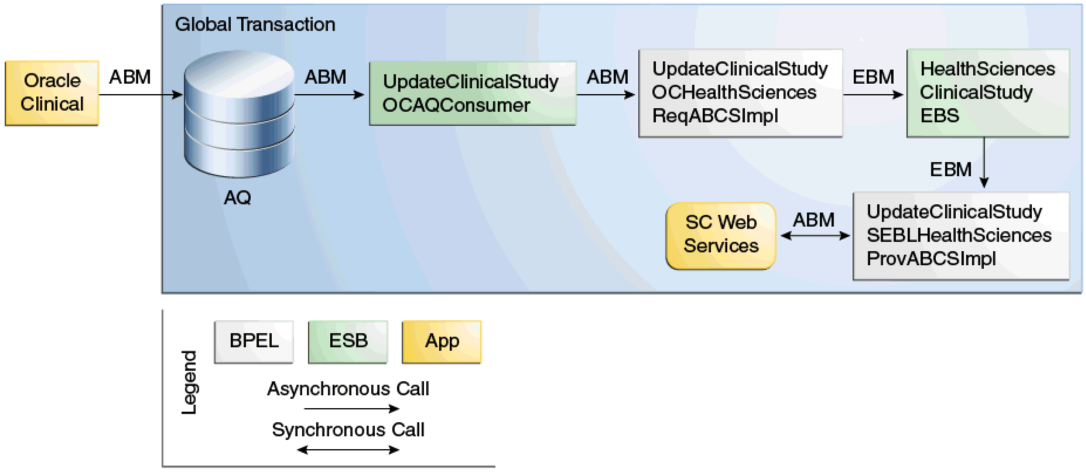
3.2 Clinical Study Subject Information Flow

This section provides the activity diagrams and functional descriptions for the Clinical Study Subject Information flow.

3.2.1 Flow Diagram

The following diagram shows the Clinical Study Subject information flow:

Figure 3–1 Clinical Study Subject Information Flow


Global Transaction
Oracle Clinical
ABM
AQ
ABM
UpdateClinicalStudy OCAQConsumer
ABM
UpdateClinicalStudy OCHealthSciences ReqABCSImpl
EBM
HealthSciences ClinicalStudy EBS
EBM
SC Web Services
ABM
UpdateClinicalStudy SEBLHealthSciences ProvABCSImpl
Legend
BPEL
ESB
App
Asynchronous Call
Synchronous Call

3.2.2 Create Study Subject

The following list describes the Create Study Subject information flow:

1. When Patient Integration is enabled for a study and either the enrollment date is entered for a patient position or the first CRF is entered for a patient position with no enrollment date, Oracle Clinical writes a message (ABM) to the Clinical_Study_Queue Advanced Queuing (AQ).
2. The UpdateClinicalStudyOCAQConsumer consumer service picks the message and passes to the UpdateClinicalStudyOCHealthSciencesReqABCSImpl (requester ABCS).
3. The message will then be transformed into an UpdateClinicalStudyEBM and:
   a. The ClinicalStudySubject ID value is updated in the CLINICALSTUDY_CLINICALSTUDYSUBJECTID cross-reference table.
   b. The ClinicalStudySite identification is retrieved from the CLINICALSTUDYSTUDYSITE cross-reference table.
   c. The Clinical Subject status is retrieved from the ClinicalStudySubject_Status DVM table.
4. The UpdateClinicalStudyEBM is passed to the HealthSciencesClinicalStudyEBS (EBS).
5. The EBS uses the configuration routing rules to pass the EBM onto the UpdateClinicalStudySEBLHealthSciencesProvABCSImpl (provider ABCS).

6. The provider ABCS transforms the EBM into an ABM for the Siebel Clinical ClinicalSubject Web service and retrieves the following information during the transformation:
   - The Siebel Clinical ID for the Protocol Site from the ClinicalStudyStudySite cross-reference table.
   - The Siebel Clinical ID for the Subject from the ClinicalStudy_ClinicalStudySubjectId cross-reference table.
   - The Clinical Subject status from the ClinicalStudySubject_Status DVM table.
7. A synchronous call is made to the Siebel Clinical ClinicalSubject Web service; the ID of the newly created Subject is returned, which will then be added to the ClinicalStudy_ClinicalStudySubjectId cross-reference table.

3.2.3 Update Study Subject

The following list describes the Update Study Subject information flow:

1. When Patient Integration is enabled for a study in Oracle Clinical and information from the patient positions table, the patient_statuses table is changed or the patient position assignment to a study site is changed, a Clinical Study ABM is written to the Clinical_Study_Queue AQ.
2. The UpdateClinicalStudyOCAQConsumer adapter service picks the message and passes to the UpdateClinicalStudyOCHealthSciencesReqABCSImpl (requester ABCS).
3. The requester ABCS retrieves the following for populating the UpdateClinicalStudyEBM:
   - Clinical Study Subject identification from the ClinicalStudy_ClinicalStudySubjectId cross-reference table.
   - Clinical Study Site identification from the ClinicalStudyStudySite cross-reference table.
   - ClinicalStudySubject Status from the ClinicalStudySubject_Status DVM.
4. The message is then transformed into an UpdateClinicalStudy EBM and passed to the HealthSciencesClinicalStudyEBS (EBS).
5. The EBS will follow routing rules to pass the EBM to the UpdateClinicalStudySEBLHealthSciencesProvABCSImpl (provider ABCS).
6. The provider ABCS retrieves the following before transforming the EBM into an ABM for the Siebel Clinical Web service:
   - Siebel Clinical ID for the Protocol Site from the ClinicalStudyStudySite cross-reference table.
   - Siebel Clinical Row ID for the Subject from the ClinicalStudy_ClinicalStudySubjectId cross-reference table.
7. A synchronous call to the Siebel Clinical ClinicalSubject Web service is made.

These processes are part of a global transaction. Compensating steps will be taken if an error occurs at any point, which prevents the process completion.

3.3 Siebel Clinical Interfaces

The integration uses the following Siebel Clinical artifacts:

Table 3–1 Siebel Inbound Web Services

Name	Schema
ClinicalSubject	Clinical Subject Internal Integration Object

See Also:

Siebel Life Sciences Guide, Version 8.1, Rev. H for more information about Siebel Web Services.

3.4 Oracle Clinical Interfaces

Outbound from Oracle Clinical Event Interfaces:

Oracle Clinical writes Clinical Study ABM to the CLINICAL_STUDY_QUEUE AQ.

3.5 Industry AIA Components

The integration workflow uses the following components:

- ClinicalStudy EBO
- HealthSciencesClinicalStudyEBSV1
- UpdateClinicalStudyEBM

The industry EBO and EBM XML Schema Definition (XSD) files can be located by EBO within the following parent folder:

$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/EBO/

The industry EBS Web Services Description Language (WSDL) files can be located by EBO within the following parent folder:

$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/HealthSciences/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in Oracle Enterprise Repository or Oracle Clinical.

See Also:

Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository", for more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link.

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.

See Also:

Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, "Understanding Extensibility".

3.6 Integration Services

These are the services delivered with this integration:

- UpdateClinicalStudyOCAQConsumer

This is an AQ adapter consumer service that reads an XML message from the AIA_ClinicalStudyQ in the Oracle Clinical database. This service routes the ABM to the Requester ABCS, UpdateClinicalStudyOCHealthSciencesReqABCSImpl.

- UpdateClinicalStudyOCHealthSciencesReqABCSImpl

  This requester ABCS is responsible for transforming the Oracle Clinical ClinicalStudy ABM into an UpdateClinicalStudyEBM.

- HealthSciencesClinicalStudyEBS

  This EBS service is responsible for passing the message from the Requester ABCS to the Provider ABCS.

- UpdateClinicalStudySEBLHealthSciencesProvABCSImpl

  This Provider ABCS is responsible for transforming the UpdateClinicalStudyEBM into the appropriate ABMs to call the Siebel Clinical ClinicalSubject Web service.

See Also:

Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository", for more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link.

4

Automating the Update of Activity Completion Date and Status Based on Data Entered in Oracle Clinical or Oracle Clinical Remote Data Capture

This chapter provides an overview of automating the update of activity completion date and status based on data that is collected in Oracle Clinical or Oracle Clinical Remote Data Capture (OCRDC) and discusses:

- Section 4.1, "Overview"
- Section 4.2, "Clinical Study Subject Activity Flow"
- Section 4.3, "Siebel Clinical Interfaces"
- Section 4.4, "Oracle Clinical Interfaces"
- Section 4.5, "Industry AIA Components"
- Section 4.6, "Integration Services"

4.1 Overview

In Siebel Clinical, activities are scheduled to happen at patient visits. The patient visit itself is considered an activity. The data in Oracle Clinical can be used to send activity status information to Siebel Clinical. For this to happen, the completion criteria for an activity or visit must be defined. The completion criteria is defined by indicating which visit, DCIs, DCMs, or DCM questions in Oracle Clinical must be completed for the activity or visit to be considered complete. An activity may be specific to a visit or could occur at multiple visits.

You can schedule a new batch job in Oracle Clinical that analyzes the patient data entered since the last time the batch job was run to see if the completion criteria have been met or changed. The process can be configured to run periodically on a custom schedule.

- When the Track Activities menu item is selected and the study in context is not enabled for Subject Activity integration, an error occurs, and you are prompted to select a new study.
- If the study context does not exist when the Track Activities menu is selected, you are prompted to select a new study.
- You can change the study context while in the Track Activities form by selecting **Special** > **Click Study** or **Change Study**.

Automating the Update of Activity Completion Date and Status Based on Data Entered in Oracle Clinical or Oracle Clinical Remote Data Capture 4-1

- During the process, the system will evaluate new patient data that has been modified since the last time the process ran to determine if activities are complete or not. If any activity completion criteria has been made active since the last time the process ran, all patient data is checked to see if any activities have been completed.
- The integration will send a message to Siebel Clinical to update the completed date and status of the activity or visit.

For more information, see Setting Up the Participating Applications

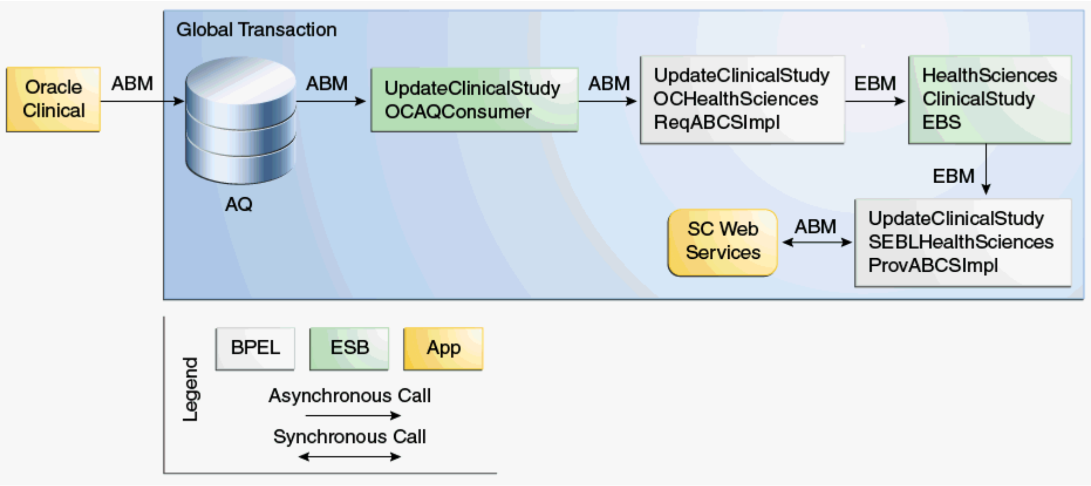
4.2 Clinical Study Subject Activity Flow

This section provides the activity diagrams and functional descriptions for Clinical Study Subject Activity Flow. This is not a synchronization flow but data from Oracle Clinical is used to update the completion date and status of a Subject activity or visit in Siebel Clinical.

4.2.1 Flow Diagram

The following diagram shows the Clinical Study Subject Activity flow:

Figure 4–1 Clinical Study Subject Activity Flow


Global Transaction
Oracle Clinical
ABM
AQ
ABM
UpdateClinicalStudy OCAQConsumer
ABM
UpdateClinicalStudy OCHealthSciences ReqABCSImpl
EBM
HealthSciences ClinicalStudy EBS
EBM
SC Web Services
ABM
UpdateClinicalStudy SEBLHealthSciences ProvABCSImpl
Legend
BPEL
ESB
App
Asynchronous Call
Synchronous Call

4.2.2 Update Subject Visit or Activity Completion

The following list describes the update subject visit or activity completion flow:

1. When Subject Activity Integration is enabled for a study in Oracle Clinical and activity completion criteria have been defined, running the Track Activity batch job in Oracle Clinical will write a ClinicalStudy (ABM) to the Clinical_Study_ Queue Advanced Queue (AQ) when the completion date of an activity has changed.

   For more information about Oracle Clinical prerequisites, see Setting Up the Participating Applications

2. The UpdateClinicalStudyOCAQConsumer adapter service picks the message and passes it to the UpdateClinicalStudyOCHealthSciencesReqABCSImpl (requester ABCS).

3. The message will then be transformed into an UpdateClinicalStudy EBM and the ClinicalStudySubject Identification is retrieved from the ClincialStudy_ClinicalStudySubjectId cross-reference table.
4. The UpdateClinicalStudyEBM is passed onto the HealthSciencesClinicalStudyEBS (EBS).
5. The EBS will follow routing rules to pass the EBM onto the UpdateClinicalStudySEBLHealthSciencesProvABCSImpl (provider ABCS).
6. The provider ABCS transforms the EBM into an ABM for the Siebel Clinical Web service. During this process, the Siebel Clinical row ID for the Subject from the ClinicalStudy_ClinicalStudySubjectId cross-reference table is retrieved.
7. A synchronous call is made to the Clinical Subject Web service in Siebel Clinical.

These processes are part of a global transaction. Compensating steps will be taken if an error occurs at any point that prevents the process completion.

4.3 Siebel Clinical Interfaces

The following table describes the Siebel Inbound Web services:

Table 4–1 Siebel Inbound Web Services

Name	Schema
ClinicalSubject	Clinical Subject Internal Integration Object

4.4 Oracle Clinical Interfaces

Outbound from Oracle Clinical Event Interfaces:

Oracle Clinical writes ClinicalStudy ABM to the CLINICAL_STUDY_QUEUE AQ.

4.5 Industry AIA Components

The integration flow uses the following components:

- ClinicalStudy EBO
- HealthSciencesClinicalStudyEBSV1
- UpdateClinicalStudyEBM

The industry EBO and EBM XML Schema Definition (XSD) files can be located by EBO within the following parent folder:

$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/EBO/

The industry EBS Web Services Description Language (WSDL) files can be located by EBO within the following parent folder:

$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Industry/HealthSciences/EBO/

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in Oracle Enterprise Repository or Oracle Clinical.

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide*

Automating the Update of Activity Completion Date and Status Based on Data Entered in Oracle Clinical Remote Data Capture 4-3

for Oracle Application Integration Architecture Foundation Pack 11g Release 1, "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 Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1,* "Understanding Extensibility".

4.6 Integration Services

These are the services delivered with this integration:

- UpdateClinicalStudyOCAQConsumer

  This is an AQ adapter consumer service that reads an XML message from the AIA_ClinicalStudyQ in Oracle Clinical database. This service routes the ABM to the Requester ABCS, UpdateClinicalStudyOCHealthSciencesReqABCSImpl.

- UpdateClinicalStudyOCHealthSciencesReqABCSImpl

  This requester ABCS is responsible for transforming the Oracle Clinical ClinicalStudy ABM into an UpdateClinicalStudyEBM.

- HealthSciencesClinicalStudyEBS

  This EBS service is responsible for passing the message from the Requester ABCS to the Provider ABCS.

- UpdateClinicalStudySEBLHealthSciencesProvABCSImpl

  This Provider ABCS is responsible for transforming the UpdateClinicalStudyEBM into the appropriate ABMs that will call the appropriate Siebel Clinical Web service.

Part II

Implementing the Delivered Integrations

Part II contains the following chapters:

5 Implementing the PIP

This chapter discusses:

- Section 5.1, "Prerequisites"
- Section 5.2, "Data Requirements"
- Section 5.3, "Setting Up the Participating Applications"
- Section 5.4, "Identifying Cross-References"
- Section 5.5, "Working with Domain Value Maps"
- Section 5.6, "Configuring Email Notification"
- Section 5.7, "Extending the PIP"
- Section 5.8, "Configuration Properties"
- Section 5.9, "Updating Server Information for Siebel Clinical and Oracle Clinical"
- Section 5.10, "Error Handling"

5.1 Prerequisites

This section discusses the following:

- Section 5.1.1, "Prerequisites for Synchronizing Clinical Study Sites"
- Section 5.1.2, "Prerequisites for Synchronizing Clinical Study Subject"
- Section 5.1.3, "Prerequisites for Automating the Update of Activity Completion Date and Status Based on Data Entered in Oracle Clinical or Oracle Clinical Remote Data Capture"

5.1.1 Prerequisites for Synchronizing Clinical Study Sites

In Oracle Clinical, the state and country values for the site address should be a valid region defined in the system. Siebel Clinical provides a list of values for states and countries that can be customized by each company. You must populate the State and Country Domain Value Map (DVM) before the integration can be used.

For more information about DVMs, see Section 5.5.

In addition to populating the DVM, there may be states and countries available in Siebel Clinical that are not recognized as regions in Oracle Clinical. Scripts have been provided to help you identify the missing regions. Run these scripts and compare the output. Enter any missing states or countries into Oracle Clinical as regions.

Note: While Oracle Clinical permits a hierarchy of regions, Siebel Clinical does not. You must enter states as regions within the country region in OC for this integration.

5.1.1.1 Downloading the Lists of States and Countries in Oracle Clinical

Perform the following steps to download a list of States and Countries in Oracle Clinical:

1. Log in to the Fusion Middleware Server and change the directory to the following location under the AIA Home:

   ```
   $AIA_HOME/data/StudySubjectVisitSyncSCandOC/sql
   ```

2. Log in to SQL*PLUS as rxa_des user.
3. SQL*PLUS > start OracleClinicalRegionDataDownload.sql.
4. SQL*PLUS > exit.
5. Verify the list of states and countries in the OracleClinicalRegionDataDownload.out file.

5.1.1.2 Downloading the Lists of States and Countries in Siebel Clinical

Perform the following steps to download a list of States and Countries in Siebel Clinical: For Oracle Database:

1. Log in to the Fusion Middleware Server and change the directory to the following location under the AIA Home:

   ```
   $AIA_HOME/data/StudySubjectVisitSyncSCandOC/sql
   ```

2. Log in to SQL*PLUS as Siebel table owner.
3. SQL*PLUS > start SiebelClinicalRegionDataDownload.sql.
4. SQL*PLUS > exit.
5. Verify the list of states and countries in the SiebelClinicalRegionDataDownload.out file.

For Microsoft SQL Server

Perform the following steps to run the script:

1. Ensure that the MS SQL server client is installed in the user machine.
2. Create a system DSN for the database connection to the Siebel database.
3. Open the MS SQL server client and log in to the database by using the System DSN created in the previous step.
4. Run the SQL script file - SiebelClinicalRegionDataDownload.sql.

Output File Format

The output file format is as follows:

```
Region Code~Name~Description~Region Type Code
ASIA~ASIA~ASIA continent~CONT
EUROPE~Europe~Europe~CONT
DE~Germany~Germany~COUNTRY
IND~INDIA~INDIA~COUNTRY
```

5.1.1.3 Increasing the Action Interval in Siebel

You must perform the following task for updates of the Protocol Site to come over from Siebel Clinical to Oracle Clinical:

- Navigate to **Admin** > **Server Management**.
- Click **Tasks**.
- Query for Workflow Monitor Agent.
- Set the **ActionInterval** parameter to any number less than 120 (for example, 5).

5.1.2 Prerequisites for Synchronizing Clinical Study Subject

You must populate the ClinicalStudySubjectStatus DVM.

For more information about DVMs, see Section 5.5.

5.1.3 Prerequisites for Automating the Update of Activity Completion Date and Status Based on Data Entered in Oracle Clinical or Oracle Clinical Remote Data Capture

You must define clinical items for each activity and visit in the active Subject Visit Templates in Siebel Clinical.

For more information about defining clinical items in Siebel Clinical, see the *Siebel Life Sciences Guide*, Version 8.1, Rev. D.

In Oracle Clinical, the study must have the Subject Activity integration enabled.

You must enter each Clinical Item as an Activity in the Maintain Activity Completion Criteria form and the completion criteria defined for it.

You must schedule the Track Activities batch job to run on a regular basis.

For more information about Oracle Clinical prerequisites, see Section 5.3.

5.2 Data Requirements

Data requirements indicate the mandatory data that must be provided to make the integration flows successful.

This section discusses the following process integrations:

- Section 5.2.1, "Data Requirements for Synchronizing Clinical Study Sites"
- Section 5.2.2, "Data Requirements for Synchronizing Clinical Study Subject Information"
- Section 5.2.3, "Data Requirements for Automating the Update of Activity Completion Date and Status Based on Data Entered in Oracle Clinical or Oracle Clinical Remote Data Capture"

5.2.1 Data Requirements for Synchronizing Clinical Study Sites

The following protocol site information is required for the integration to successfully create and update a Study Site in Oracle Clinical:

A primary address must be defined for the Protocol Site. The primary address must include the following defined values:

- Address Line
- City

- State
- Zip Code
- Country
- Phone

Each principal investigator assigned to a protocol site must have these defined values:

- First Name
- Last Name
- Phone Number

5.2.2 Data Requirements for Synchronizing Clinical Study Subject Information

The patient position enrollment date in Oracle Clinical must be populated for a subject to be assigned an Enrollment Visit Template in Siebel Clinical.

5.2.3 Data Requirements for Automating the Update of Activity Completion Date and Status Based on Data Entered in Oracle Clinical or Oracle Clinical Remote Data Capture

For more information about defining clinical items in Siebel Clinical, see the *Siebel Life Sciences Guide*, Version 8.1, Rev. D.

For more information about mapping activity completion criteria in Oracle Clinical, see Section 5.3.2.4.3.

5.3 Setting Up the Participating Applications

This section discusses:

- Section 5.3.1, "Setting up Siebel Clinical"
- Section 5.3.2, "Setting up Oracle Clinical"

5.3.1 Setting up Siebel Clinical

This section describes the tasks and general procedures that you must complete to integrate a Siebel Clinical protocol with an Oracle Clinical study.

For more information about configuring Siebel Clinical for integration with Oracle Clinical, see the *Siebel Life Sciences Guide*, Version 8.1, Rev. D.

5.3.1.1 Synchronizing Clinical Study Sites

Study site information is entered as a Protocol Site in Siebel Clinical and is used to create a Study Site in Oracle Clinical. This integration is unidirectional, from Siebel Clinical to Oracle Clinical.

For Study Sites to be synchronized between Siebel Clinical and Oracle Clinical, you must assign the CDMS Study ID for the protocol and select the **Synchronize Active Sites** check box. When the Siebel Clinical Protocol Site is ready to be synchronized with the associated Oracle Clinical study, you must select a primary address from the list of Protocol Site addresses and select the **Activate Synchronization** check box. When this is complete, the Oracle Clinical study and the Siebel Clinical protocol are linked and the first Siebel Clinical protocol site is associated with a newly created study site in Oracle Clinical.

When the Activate for Synchronization flag is selected for a Protocol Site in Siebel Clinical, a Study Site is created in Oracle Clinical. If the principal investigator responsible for the protocol site does not exist in Oracle Clinical, one is created.

The combination of the primary address of the protocol site and the account are considered as a Site in Oracle Clinical. If this combination does not exist, a new site is created in Oracle Clinical.

Updates to the investigator, site, or study site in Siebel Clinical are synchronized to Oracle Clinical. Updates made in Oracle Clinical for investigator, site, or study site are not synchronized back to Siebel Clinical. Deletion of a principal investigator account or protocol site in Siebel Clinical will not result in deletion of any objects in Oracle Clinical.

For more information about configuring Siebel Clinical for integration with Oracle Clinical, see the *Siebel Life Sciences Guide*, Version 8.1, Rev. D. For more information about configuring relevant integration components, see Chapter 2.

5.3.1.2 Synchronizing Clinical Study Subject Information

After you have enabled Patient Integration for an Oracle Clinical study, the synchronization process for patient information can be initiated. In Oracle Clinical, patient positions are created for the target enrollment of a study and assigned to study sites based on the target enrollment for the study site. If the enrollment date is entered for a patient position, it indicates that a patient has been enrolled in the study and assigned an Enrollment ID. This creates a subject for the appropriate Protocol Site in Siebel Clinical and the active Enrollment Visit Template to be assigned to the subject.

If data is collected for a patient position in Oracle Clinical or OCRDC, and no enrollment date has been specified, the system assumes this to be a patient undergoing the screening process and assigns a Screening ID. This creates a subject for the appropriate Protocol Site in Siebel Clinical and assigns the active Screening Visit Template to it.

Although Siebel Clinical requires the subject's initials and birth date to create a subject, this information is not mandatory in Oracle Clinical. If the patient's initials are not entered in Oracle Clinical, the Enrollment ID is used as the initials in Siebel Clinical. If the patient's birth date is not entered in Oracle Clinical, the birth date is set to Jan 1, 1800 in Siebel Clinical. This is an invalid date that can be used to identify subjects whose information should be updated. You can then manually correct the data in Siebel Clinical.

The Informed Consent Date is not collected in Oracle Clinical. Therefore, subjects that are created in Siebel Clinical by the integration will have the Informed Consent Date set to the birth date of the subject. This will alter the dates in the Visit Schedule for the subject and should be manually corrected in Siebel Clinical.

When Siebel Clinical assigns a Screening Visit Template to a subject, the screen date is required. This is not a required field in Oracle Clinical. Therefore, if this date is not collected for a subject in Oracle Clinical, the system populates a default value of Jan 1, 1900. This will impact the dates in the Visit Schedule for the subject and should be manually corrected in Siebel Clinical.

For more information about synchronization, see Section 1.3.2 and Chapter 3.

5.3.1.3 Automating the Update of Activity Completion Criteria

In Siebel Clinical, activities are scheduled to occur at patient visits. The patient visit itself is considered an activity. Based on completion criteria you have defined for an activity or visit, the data in Oracle Clinical can be used to send information about the status of an activity to Siebel Clinical. Completion criteria are defined by indicating which visit, DCIs, DCMs, or DCM questions in Oracle Clinical must be completed for the associated activity or visit to be considered complete.

Clinical Items

In Siebel Clinical, Subject Visit Templates are defined for screening and enrollment visits. When protocols are amended, new versions of Subject Visit Templates are created and activated. Although the same activity may occur at different visits in a Subject Visit Template or across different versions of a visit template, Siebel Clinical considers each activity as a separate entity.

To minimize the effort involved in defining the same completion criteria for the same activity multiple times, the concept of a clinical item has been introduced in Siebel Clinical. When a new activity is added to a Subject Visit Template, by default, the system sets the clinical item to the activity description.

Visits are also considered as activities. When a new visit is added to a Subject Visit Template, the clinical item defaults to the visit name.

If a visit has the same completion criteria as that of a visit in another version or in an entirely different Subject Visit Template, the visits can be assigned to the same Clinical Item. This lets the completion criteria to be specified once in Oracle Clinical. However, the same clinical item cannot be added to more than one visit in the same Subject Visit Template version.

An example of this is a Screening Visit Template comprising of two visits, Visit 1 and Visit 2. If a protocol amendment changes Visit 2, a new version of the Screening Visit Template is created. You can create a clinical item value that is assigned to Visit 1 in both versions of the Screening Visit Template.

If a Subject Visit activity occurs at multiple visits or in multiple Subject Visit Templates or Subject Visit Template versions, the same clinical item can be assigned to that set of activities. For example, a clinical item can be assigned to the set of identical lab tests that are performed at each of the visits. This enables the completion criteria for that activity to be defined once in Oracle Clinical against a single clinical item.

For more information on defining clinical items, see *Siebel Life Sciences Guide, Version 8.1, Rev. D.*

5.3.2 Setting up Oracle Clinical

In Siebel Clinical, when the Activate for Synchronization field is selected for a Protocol Site, a corresponding Study Site is created in Oracle Clinical. A study site in Oracle Clinical cannot exist without Investigator and Site details. If the corresponding information is not available in Siebel Clinical, it will be created in Oracle Clinical. These new objects are used to create the Study Site in Oracle Clinical. All sites and investigators transferred by the integration are created as active.

5.3.2.1 Synchronizing Clinical Study Sites

In Oracle Clinical, any updates made to a site or an investigator associated with an integrated study may be overwritten by updates from Siebel Clinical.

5.3.2.1.1 The LocalStudySite.ChangeStudySite Procedure

Oracle recommends you to modify the PL/SQL procedure, LocalStudySite.ChangeStudySite, to prevent study sites created by user, RXA_WS, from being updated. This prevents someone in Oracle Clinical from breaking the link between Protocol Site in Siebel Clinical and Study Site in Oracle Clinical.

5.3.2.2 Enabling the Integration

Oracle Clinical has new options that facilitate the integration with Siebel Clinical. One option enables the transfer of patient_position information once the enrollment date has been entered or data has been entered against a patient position without an enrollment date. The other option permits completion criteria for Siebel Clinical activities to be defined in Oracle Clinical and send messages when those activities have been completed.

The following are new settings:

- **Enable Patient Integration** - When this is set to Y, and the enrollment date is entered, or data is entered for a patient position with no enrollment date, messages about patient_positions, patient_status, and the study sites assignments for the patient _positions are written to an Advanced Queue. All subsequent updates to these patient positions are sent to Siebel Clinical.
- **Enable Subject Activity Integration** - When this is set to Y, you will be able to define completion criteria for visits and activities in Siebel Clinical and schedule jobs to send messages to Siebel Clinical when the activities have been completed.

The following sections describe how to set these integrations at the database or study level:

5.3.2.2.1 Enabling Integration for All New Studies

To enable integration at the database level for all new studies, navigate to **Admin** > **Integration Database Settings**.

The system displays the Integration Database Settings window. There are two integration settings available:

- Enable Patient Integration
- Enable Subject Activity Integration

For each setting the default value is N, which indicates that the integration is turned off. If you want to enable one or both of these integrations on all new studies, change the settings to `Y` and all new studies will be created with the corresponding integrations enabled.

> **Note:** Changing these settings will not affect any existing studies.

If you want to enable the integration only for a few studies, proceed to the study level settings described in the following section.

5.3.2.2.2 Enabling Integration for a Single Study

To enable integration at the study level for an individual study, navigate to **Design** > **Integration Study Settings**. The system displays the Integration Study Settings window and specifies the active study. At the database level, there are two integration settings available:

- Enable Patient Integration
- Enable Subject Activity Integration

The default values for these settings are based on the value of the database level setting defined in **Admin** > **Integration Database Settings** form at the time the study was created.

To enable the relevant integration for the study, change the value from N to Y.

Note: Both the Patient Integration and the Subject Activity Integration must be enabled to use the Subject Activity integration.

5.3.2.3 Synchronizing Clinical Study Subject Information

Enrollment Date and Informed Consent Date

Siebel Clinical requires an enrollment date to be specified for enrolled patients. If you use the CRF data entry instead of the Patient Positions form to capture this value, you can use the pat_sync function in a derivation procedure to populate the enrollment date in the Patient Positions table. You can also use pat_sync to enter the Informed Consent Date of a patient.

For more information about patient synchronization, see *Oracle Clinical Documentation*, "Creating a Study."

5.3.2.4 Automating the Update of Activity Completion Criteria

5.3.2.4.1 Track Activity Batch Process

The Track Activity process is a PSUB batch job, which evaluates all activity completion criteria and determines if the patient data entered or updated since the last time the track activity job was run will cause the completion status or date of an activity or visit to change. If new activity completion criteria have become active since the last time the track activity job was run, all existing patient data for the study is examined.

To run the Track Activity process, navigate to **Conduct** > **Data Validation** > **Track Activities**. The system displays the Track Activity Completion window.

If no study context exists when the Track Activities menu is selected, the system prompts you to select a new study.

When you select the **Track Activities** menu item, if the study in context is not enabled for Subject Activity integration, the system displays an error and prompts you to select a new study.

If you want to change the study context while in the Track Activities window, select **Special** > **Click Study** or **Change Study**.

For more information about running, viewing, and scheduling PSUB jobs, see *Oracle® Clinical Conducting a Study* Guide.

5.3.2.4.2 Completion Criteria Processing

The following parameters define how the system will process completion criteria objects.

Completeness Criteria

1. A Received DCI (RDCI) or Received DCM (RDCM) that is marked blank is not considered complete.
2. If the study requires two passes, only an RDCI or RDCM, with the status PASS 2 COMPLETE or BATCH LOADED is considered complete.
3. If the study does not require two passes, only a RDCI or RDCM, with the status PASS 1 COMPLETE, PASS 2 STARTED, PASS 2 PENDING, PASS 2 COMPLETE or BATCH LOADED is considered complete.

Exception Processing

An activity will no longer be considered as complete if:

1. A DCI or DCM that was mapped to an activity is soft deleted.
2. The blank flag is set to Y for a DCI or DCM that was mapped to an activity.
3. A response value for a DCM question that was mapped to an activity is modified to a value other than the one defined in the mapping.
4. A response value for a DCM question that was mapped to an activity is removed leaving the response empty.

If the activity is no longer considered complete due to one of the preceding events, the following information is written to the database queue:

- Activity ID (Note that this is the clinical item that is selected for the Activity name in the Map Activities form)
- Study
- Study Site Code
- Patient
- Visit (Note that this is the clinical item mapped to the Visit)
- Activity Code (can be VISIT or ACTIVITY)
- Blank value for Completed Date

Key Change Processing

1. If the patient key is changed for a RDCI or RDCM that was mapped to an activity, the activity is no longer considered complete for the old patient.
2. If the DCI book for an RDCI is either mapped to a DCI Book Visit or contains a RDCM or response mapped to a DCI Book Visit, the activity is no longer considered complete for the mapped visit.
3. If the visit for a RDCI is either mapped to an activity or contains a RDCM or response mapped to an activity, the activity is no longer considered complete for the old visit.

Updating Activity Queue

If the completed date for an activity has changed since the last time the process was run, the following information is written to the database queue:

- Activity ID (Note that this is the clinical item that is selected for the Activity name in the Map Activities form)
- Study
- Study Site Code
- Patient
- Visit (Note that this is the clinical item mapped to the Visit)
- Activity Code (can be VISIT or ACTIVITY)
- Completed Date

5.3.2.4.3 Defining Completion Criteria

Oracle Clinical provides the means to map Siebel Clinical activities to Oracle Clinical study definition objects. The purpose of this portion of the integration is to let events in Oracle Clinical - collection of specific data - to trigger the completion of associated activities in Siebel Clinical.

For more information about defining activity completion criteria, see *Oracle Clinical Creating a Study*.

5.4 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".

The following are the cross-references for Siebel Clinical to Oracle Clinical:

Table 5–1 Cross-references for Siebel Clinical to Oracle Clinical

REFTABLENAME	COLUMN NAME	DESCRIPTION	USAGE
CLINICALSTUDYSITE	COMMON	Concatenation of row ID of Account and row id of Primary Address for the Protocol Site in Siebel Clinical	Populated when a Site is created in Oracle Clinical during the Study Site integration.
-	SEBLCLIN_01	Concatenation of row ID of Account and row ID of Primary Address for the Protocol Site in Siebel Clinical	-
-	OC_O1	Concatenation of Site ID and Site (user identifier) from Oracle Clinical	-
CLINICALSTUDYINVESTIGATOR	COMMON	Row ID for contact in Siebel Clinical	Populated when an Investigator is created in Oracle Clinical during the Study Site integration
-	SEBLCLIN_01	Row ID for contact in Siebel Clinical	-
-	OC_O1	Concatenation of Investigator ID and Investigator (user identifier) in Oracle Clinical	-
CLINICALSTUDYSTUDYSITE	COMMON	Concatenation of CDMS_STUDY_ID from Protocol in Siebel Clinical and Protocol Site Number	Populated when Study Site is created during Study Site integration.
-	SEBLCLIN_01	Protocol Site Id from Protocol Site in Siebel Clinical	-
-	OC_01	Concatenation of Clinical Study ID and Site ID in Oracle Clinical	-

The following are the cross-references for Oracle Clinical to Siebel Clinical:

Table 5–2 Cross-references for Oracle Clinical to Siebel Clinical

REFTABLENAME	COLUMN NAME	DESCR	USAGE
CLINICALSTUDYSTUDYSITE	COMMON	Concatenation of CDMS_STUDY_ID from Protocol in Siebel Clinical and Protocol Site Number	Used to look up Siebel Clinical Study Site when subjects are created during the Subject Integration.
-	SEBLCLIN_01	Protocol Site ID from Protocol Site in Siebel Clinical	-
-	OC_01	Concatenation of Clinical Study ID and Site ID in Oracle Clinical	-

Table 5–2 (Cont.) Cross-references for Oracle Clinical to Siebel Clinical

REFTABLENAME	COLUMN NAME	DESCR	USAGE
CLINICALSTUDY_ CLINICALSTUDYSUBJECT ID	COMMON	Concatenation of Clinical Study Code (Study) and Patient Position Code (Patient) from Oracle Clinical	Populated when subjects are created during the Subject integration.
-	SEBLCLIN_01	Row id of the Subject in Siebel Clinical	-
-	OC_01	Patient Position ID for Patient Position in Oracle Clinical	-

5.5 Working with Domain Value Maps

Domain value maps (DVMs) are a standard feature of the Oracle SOA suite, which enables 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.

The following are the DVMs for Study, Subject, and Visit Synch: Siebel Clinical and Oracle Clinical PIP:

Table 5–3 DVMs for Study, Subject and Visit Synch: Siebel Clinical and Oracle Clinical PIP

DVM Type	DVM Column Name	Comments
COUNTRY	COMMON, SEBLCLIN_01, OC_O1	This maps the country codes between Siebel Clinical and Oracle Clinical.
STATE	COMMON, SEBLCLIN_01, OC_O1	This maps the state codes between Siebel Clinical and Oracle Clinical.
CLINICALSTUDYSUBJEC T_STATUS	COMMON, SEBLCLIN_01, OC_01	This maps the patient status defined by customers in Oracle Clinical to the Subject status in Siebel Clinical.

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".

5.6 Configuring Email Notification

To configure email notification, see *Oracle® Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack, Setting Up Error Handling.*

5.7 Extending the PIP

5.7.1 Enabling Calling Out To Your Own Web Services

This integration lets you call external web services from all the ABCS services.

Figure 5–1 shows various invocation points for external web services from a Requester ABCS - UpdateClinicalStudySEBLHealthSciencesReqABCSImpl or UpdateClinicalStudyOCHealthSciencesReqABCSImpl.

Figure 5–1 Calling Web Services From Requester ABCS

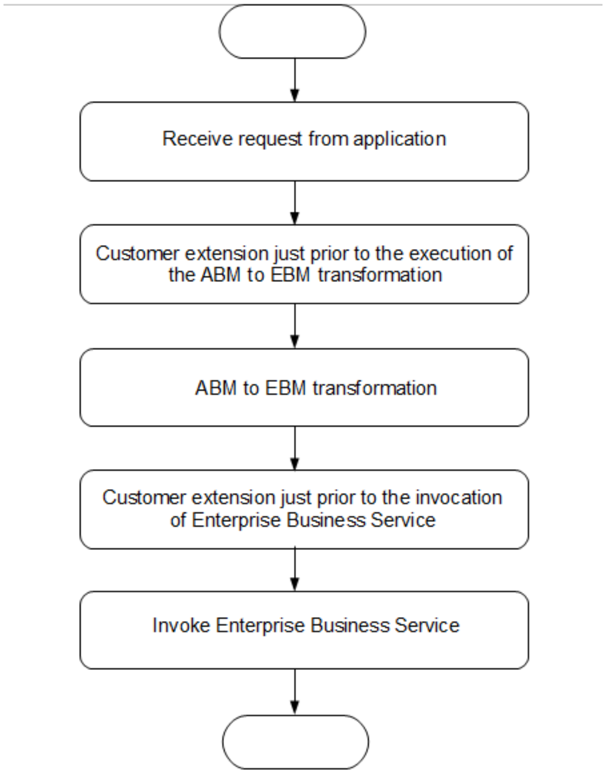
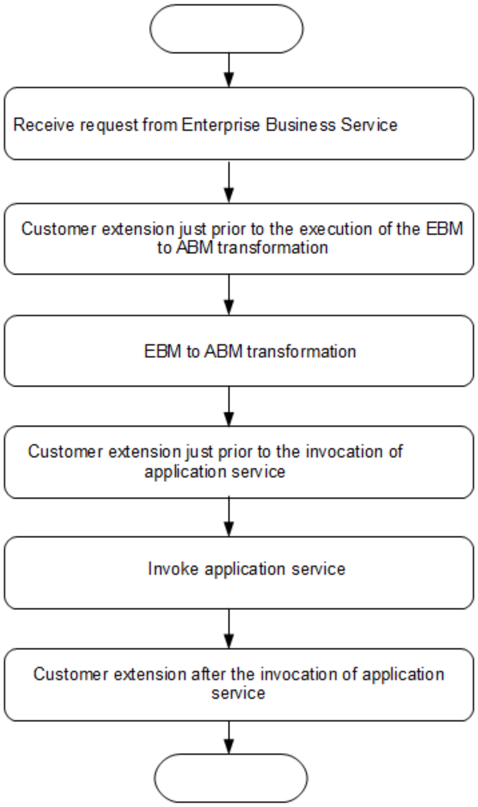

Receive request from application
Customer extension just prior to the execution of the ABM to EBM transformation
ABM to EBM transformation
Customer extension just prior to the invocation of Enterprise Business Service
Invoke Enterprise Business Service

Figure 5–2 shows invocation points for external web services for Provider ABCS - UpdateClinicalStudyOCHealthSciencesProvABCSImpl or UpdateClinicalStudySEBLHealthSciencesProvABCSImpl.

Figure 5–2 Calling Web Services From Provider ABCS


Receive request from Enterprise Business Service
Customer extension just prior to the execution of the EBM to ABM transformation
EBM to ABM transformation
Customer extension just prior to the invocation of application service
Invoke application service
Customer extension after the invocation of application service

To invoke an external web service in an ABCS, perform the following:

1. Open `<AIA_HOME>/aia_instances/<instance_name>/AIAMetaData/config/AIAConfigurationProperties.xml`.
2. Locate the **ServiceConfiguration** section of the ABCS that is being extended.

   For example, if you want Siebel Requester ABCS to call an external web service, go to **Service Configuration** with service name as **{http://xmlns.oracle.com/ABCSImpl/SiebelClinical/Core/UpdateClinicalStudySEBLHealthSciencesReqABCSImpl/V1}UpdateClinicalStudySEBLHealthSciencesReqABCSImpl**.
3. Based on your requirement for control logic, set one of the following extension points to True.

   In a requester ABCS, enable one of the following:

   - PreXformABMtoEBM - Invokes an external web service before ABM to EBM transformation.
   - PreInvokeEBS - Invokes an external web service before invoking EBS.

   In a provider ABCS, enable one of the following:

   - PreXformEBMtoABM - Invokes an external web service before EBM to ABM transformation.
   - PreInvokeABS - Invokes an external web service before invoking Application Business Service (ABS).
   - PostInvokeABS - Invokes an external web service after invoking ABS.

4. Save AIAConfigurationProperties.xml.
5. Upload it to the MDS using the following steps:
   a. Navigate to the `$AIA_INSTANCE/config` directory.
   b. Copy `UpdateMetaDataDp.xml` to a backup file.
   c. Edit `UpdateMetaDataDp.xml` file as follows:

      ```
      <?xml version="1.0" standalone="yes"?>
      <DeploymentPlan component="Metadata" version="3.0">
      <Configurations>
      <UpdateMetadata wlserver="fp">
      <fileset dir="${AIA_INSTANCE}/AIAMetaData">
      <include name="config/AIAConfigurationProperties.xml"/>
      </fileset>
      </UpdateMetadata>
      </Configurations>
      </DeploymentPlan>
      ```
   d. Navigate to the `$AIA_INSTANCE/bin` directory.
   e. Execute the following commands:

      For Windows: execute `aiaenv.bat`

      ```
      ant -f %AIA_HOME%\Infrastructure\Install\config\UpdateMetaData.xml
      ```

      For Linux: `source aiaenv.sh`

      ```
      ant -f $AIA_HOME/Infrastructure/Install/config/UpdateMetaData.xml
      ```
   f. Wait till you see a build successful result.
6. Open the extension service concrete WSDL for the ABCS that is being extended. For example, if Siebel Requester ABCS is being extended to call an external web service, open the file `<AIA_HOME>/AIAMetaData/AIAComponents/ExtensionServiceLibrary/SiebelClinical/UpdateClinicalStudySEBLHealthSciencesReqABCSImplExtensionConcrete.wsdl`.
7. Locate the soap:address section of the WSDL towards the end of the file.
8. Replace the Mirror Servlet location (`http://soaserverhost:soaserverport/MirrorServlet/mirror`) with the actual web service endpoint.
9. Save the concrete WSDL.
10. Upload it to the MDS using the following steps:
    a. Navigate to the `$AIA_INSTANCE/config` directory.
    b. Copy `UpdateMetaDataDp.xml` to a backup file.
    c. Edit `UpdateMetaDataDp.xml` 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="AIAComponents/ExtensionServiceLibrary/SiebelClinical/UpdateClinicalSt
       udySEBLHealthSciencesReqABCSImplExtensionConcrete.wsdl"/>
       </fileset>
       </UpdateMetadata>
       </Configurations>
       ```

```
</DeploymentPlan>
```

d. Navigate to the `$AIA_INSTANCE/bin` directory.

e. Execute the following commands:

For Windows: execute `aiaenv.bat`

```
ant -f %AIA_HOME%\Infrastructure\Install\config\UpdateMetaData.xml
```

For Linux: `source aiaenv.sh`

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

f. Wait till you see a build successful result.

5.7.2 Adding Custom Transformations

5.7.2.1 Overview

To add custom transformation on fields that are already mapped and transformed in the PIP, you must first identify the transformation that must be extended and customized. You can use custom transformation constructs to enable or disable the custom transformations based on properties in the AIAConfigurationProperties.xml file. You can also use this file to define the custom transformations.

5.7.2.2 Customizing Enterprise Business Object Fields for Study Site Flow from Siebel Clinical to Oracle Clinical

This section discusses the following

- Section 5.7.2.2.1, "Adding Custom Transformations - An Example"
- Section 5.7.2.2.2, "Uploading AIAConfigurationProperties.xml to MDS"
- Section 5.7.2.2.3, "Deploying OC Provider Changes to Server"

5.7.2.2.1 Adding Custom Transformations - An Example

SC-OC integration does not let custom transformations be added to the field already mapped in the core transformation xsl file. There are some fields such as, investigator address which are mapped to be blank in the core xsl. There can be scenario where you want to put the site address as the investigator address in OC. To do this, you need to customize the transformation.

As an example, you can extend OC Investigator EBM to ABM Transformation and add custom transformation in the CreateInvestigator flow. The following figure displays the existing elements in this flow as delivered by the integration and the additional constructs. This diagram is a snapshot of the OCProvider.bpel file in Oracle JDeveloper.

Figure 5–3 Create Investigator ABM

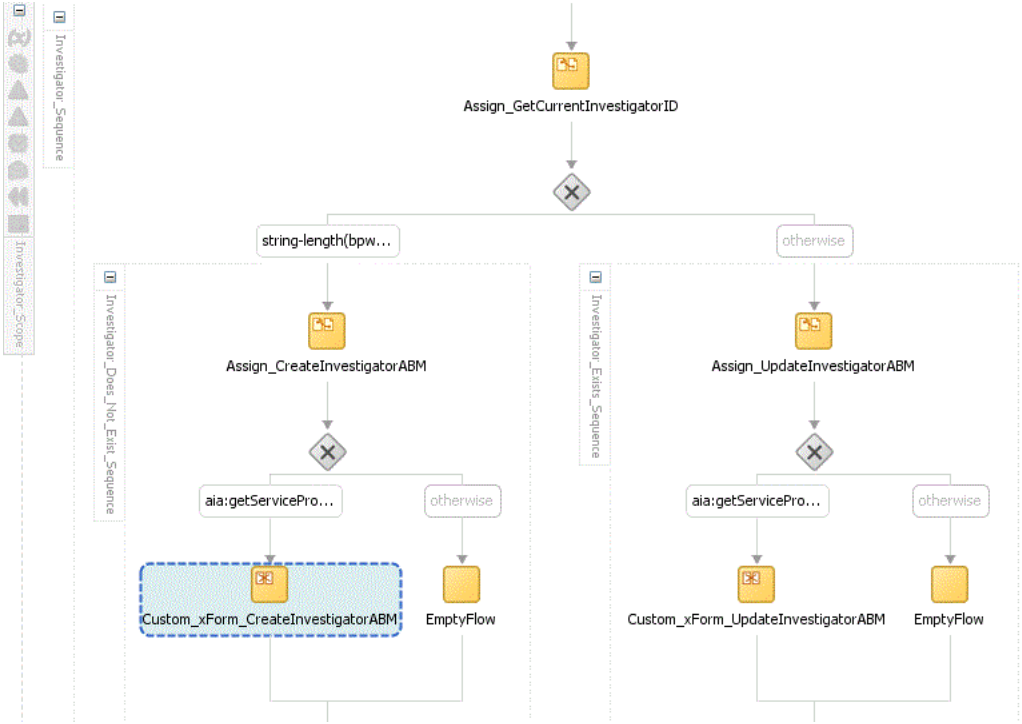

Investigator_Scope
Investigator_Sequence
Assign_GetCurrentInvestigatorID
string-length(bpw...
otherwise
Investigator_Does_Not_Exist_Sequence
Investigator_Exists_Sequence
Assign_CreateInvestigatorABM
Assign_UpdateInvestigatorABM
aia:getServicePro...
otherwise
aia:getServicePro...
otherwise
Custom_xForm_CreateInvestigatorABM
EmptyFlow
Custom_xForm_UpdateInvestigatorABM
EmptyFlow

Figure 5–4 Create Site ABM

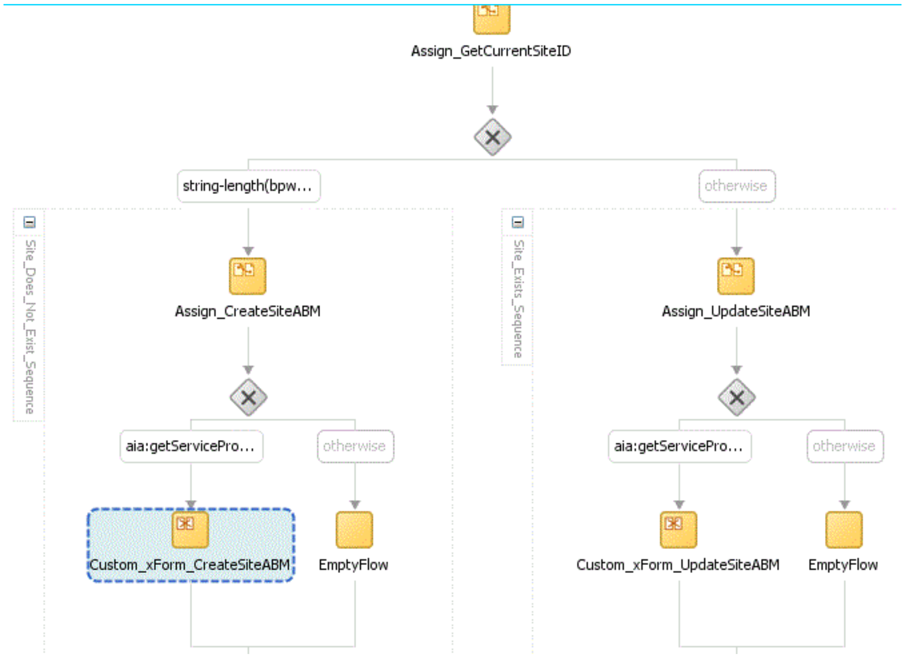
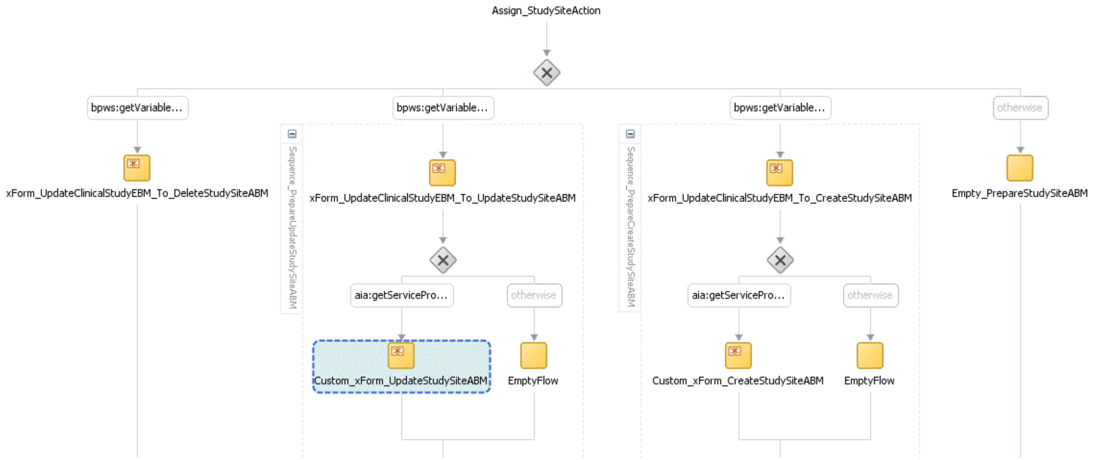


Figure 5–5 Create Study Site ABM


Assign_StudySiteAction
bpws:getVariable...
bpws:getVariable...
bpws:getVariable...
otherwise
xForm_UpdateClinicalStudyEBM_To_DeleteStudySiteABM
Sequence_PrepareUpdateStudySiteABM
xForm_UpdateClinicalStudyEBM_To_UpdateStudySiteABM
Sequence_PrepareCreateStudySiteABM
xForm_UpdateClinicalStudyEBM_To_CreateStudySiteABM
Empty_PrepareStudySiteABM
aia:getServicePro...
otherwise
aia:getServicePro...
otherwise
Custom_xForm_UpdateStudySiteABM
EmptyFlow
Custom_xForm_CreateStudySiteABM
EmptyFlow

To invoke the custom transformation, you must set the property in AIAConfigurationProperties.xml. In the above example, the following properties have been added:

Table 5–4 New Properties

Properties	Enables custom transformations to...
CustomXForm.InvestigatorABM	Investigator ABM (Figure 5–3)
CustomXForm.SiteABM	Site ABM (Figure 5–4)
CustomXForm.StudySiteABM	Study Site ABM (Figure 5–5)

Check if the properties in the Table 5–4 are present in the AIAConfigurationProperties.xml file. If not, add these properties at the end of the Service Configuration section of OCProvider in the AIAConfigurationProperties.xml file as shown in the following sample code.

> **Note:** These properties are not present in the AIAConfigurationProperties.xml file when you upgrade from release 3.1 to 11.1.

```
<ServiceConfiguration
serviceName="{http://xmlns.oracle.com/ABCSImpl/OracleClinical/Industry/HealthScien
ces/UpdateClinicalStudyOCHealthSciencesProvABCSImpl/V1}UpdateClinicalStudyOCHealth
SciencesProvABCSImpl">
<Property name="CustomXForm.InvestigatorABM">false</Property>
<Property name="CustomXForm.SiteABM">false</Property>
<Property name="CustomXForm.StudySiteABM">false</Property>
</ServiceConfiguration>
```

Add the actual transformations to the following six transformation files for the respective ABMs:

- Custom_xForm_CreateInvestigatorABM.xsl
- Custom_xForm_CreateStudySiteABM.xsl
- Custom_xForm_UpdateInvestigtorABM.xsl
- Custom_xForm_UpdateSiteABM.xsl

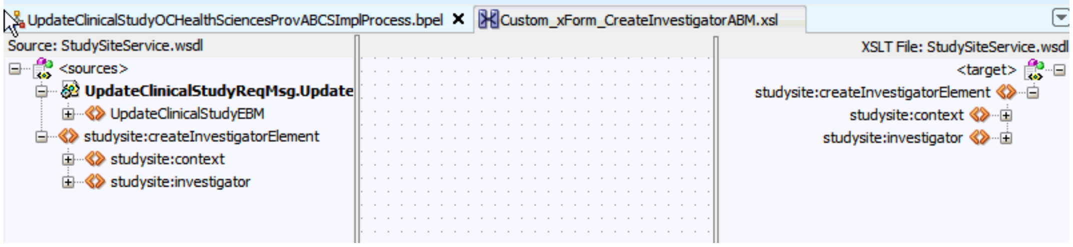
- Custom_xFrom_CreateSiteABM.xsl
- Custom_xFrom_UpdateStudySiteABM.xsl

To add a custom transformation for Investigator address line 1 field, and map Study Site address line 1 to Investigator address line1 field, you have to modify the Create and Update Investigator custom xsl. Perform the following steps:

1. In the AIAConfigurationProperties.xml file, set the value of CustomXForm.InvestigatorABM to *true*.
2. Upload the updated AIAConfigurationProperties.xml file to MDS. For more information, see Section 5.7.2.2.2.
3. To modify the Create Investigator transformation, open the Custom_xForm_CreateInvestigatorABM.xsl file.

   Figure 5–6 displays the Custom_xForm_CreateInvestigatorABM.xsl file.

Figure 5–6 The Custom_xForm_CreateInvestigatorABM.xsl File



UpdateClinicalStudyOCHealthSciencesProvABCSImplProcess.bpel
Custom_xForm_CreateInvestigatorABM.xsl

Source: StudySiteService.wsdl
<sources>
UpdateClinicalStudyReqMsg.Update
UpdateClinicalStudyEBM
studysite:createInvestigatorElement
studysite:context
studysite:investigator

XSLT File: StudySiteService.wsdl
<target>
studysite:createInvestigatorElement
studysite:context
studysite:investigator

   By default, there are no mappings defined in this file.

4. Perform the following steps to create a copy of the Target Investigator ABM:

   a. Right-click on the transformation file middle pane and choose **Auto Map Preferences**. The Auto Map Preferences screen is displayed.

   b. Ensure that the following options are selected:

      – Match Elements with Similar Names

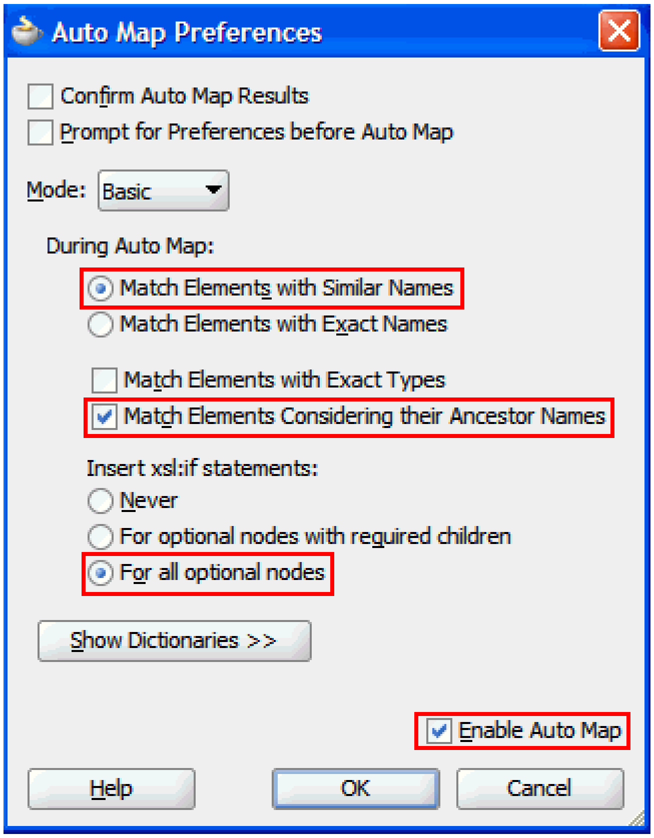
      – Match Elements Considering their Ancestor Names

      – For all optional node

      – Enable Auto Map

      Figure 5–7 displays the Auto Map Preferences dialog box with the selected options.

Figure 5–7 Auto Map Preferences Dialog Box



Auto Map Preferences

☐ Confirm Auto Map Results

☐ Prompt for Preferences before Auto Map

Mode: Basic

During Auto Map:

● Match Elements with Similar Names

○ Match Elements with Exact Names

☐ Match Elements with Exact Types

☑ Match Elements Considering their Ancestor Names

Insert xsl:if statements:

○ Never

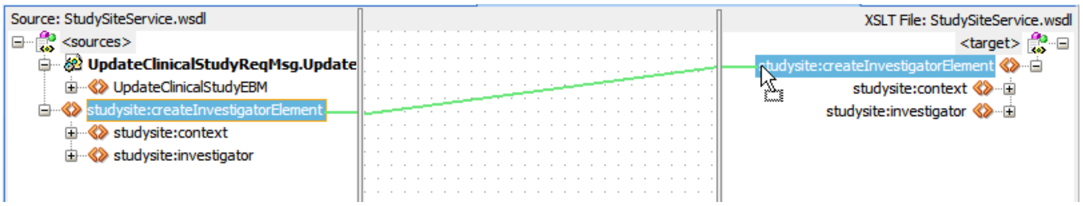
○ For optional nodes with required children

● For all optional nodes

☑ Enable Auto Map

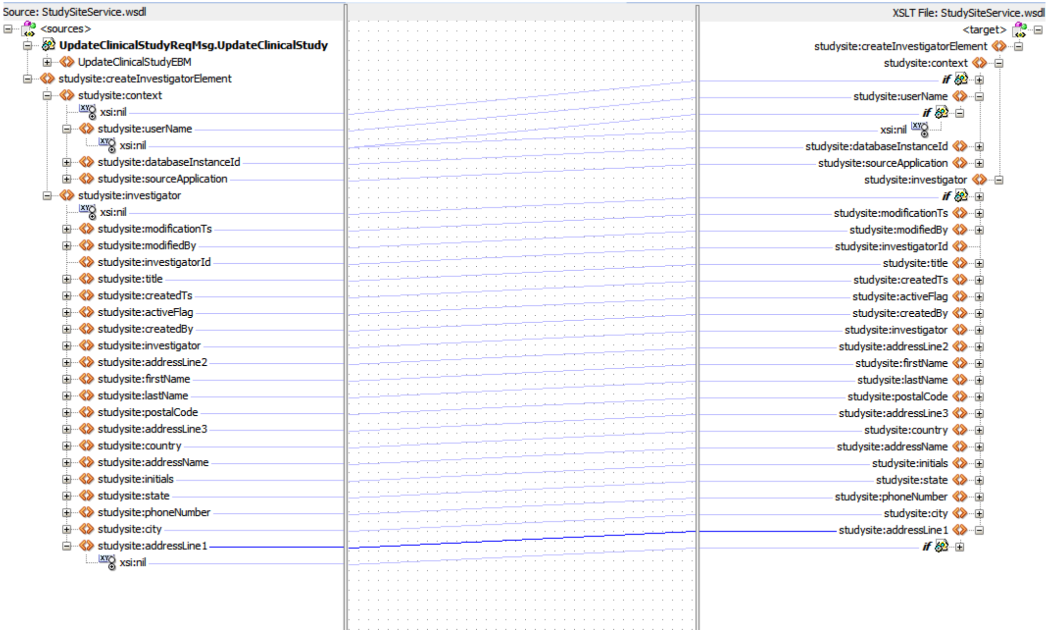
c. Map the *studysite:createInvestigatorElement* in the source to *studysite:createInvestigatorElement* in the target as displayed in the following figures:

Figure 5–8 Mapping studysite:createInvestigatorElement



Source: StudySiteService.wsdl
<sources>
UpdateClinicalStudyReqMsg.Update
UpdateClinicalStudyEBM
studysite:createInvestigatorElement
studysite:context
studysite:investigator
XSLT File: StudySiteService.wsdl
<target>
studysite:createInvestigatorElement
studysite:context
studysite:investigator

Figure 5–9 Mapping studysite:createInvestigatorElement

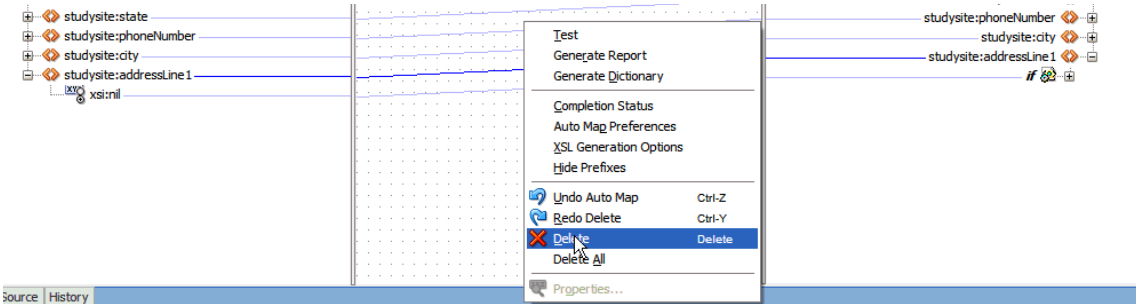


5. Perform the following steps to add the custom mapping to override the mapping for addressLine1:

   a. Expand the nodes on both source and target.

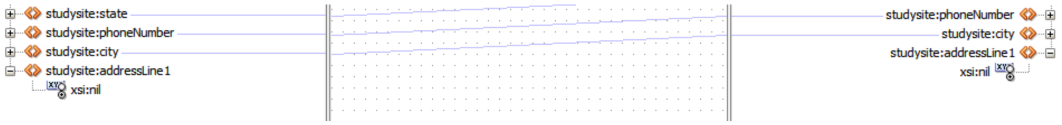
   b. Right-click and select **Delete** to delete the mapping for addressLine1 field as shown in the Figure 5–10:

Figure 5–10 Deleting a Mapping



   c. Delete mapping for corresponding xsi:nil attribute and the additional if node for that xsi:nil attribute.

Figure 5–11 Mappings Deleted

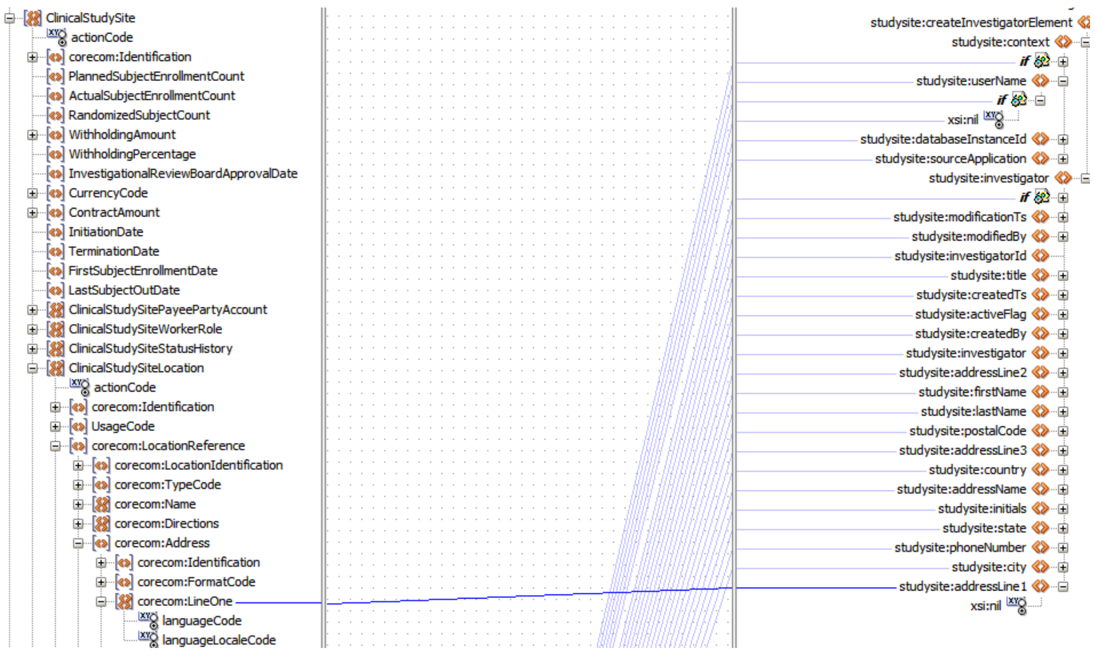


   d. To map the StudySite Address to the Investigator Address, in the source, navigate to following location, and map it to studysite:addressLine1 in the target schema:

$UpdateClinicalStudyReqMsg.UpdateClinicalStudy/ns0:UpdateClinicalStudyEBM/ns0:DataArea/ns0:UpdateClinicalStudy/ns0:ClinicalStudySite/ns0:ClinicalStudySiteLocation/corecom:LocationReference/corecom:Address/corecom:LineOne

Figure 5–12 displays the result.

Figure 5–12 Mapping StudySite Address to Investigator Address



6. Similarly, modify the Update Investigator xsl.
7. Deploy the changes to OC Provider to the server. For more information, see Section 5.7.2.2.3.
8. Test the custom transformation.

5.7.2.2.2 Uploading AIAConfigurationProperties.xml to MDS

To upload the AIAConfigurationProperties.xml file to MDS, perform the following steps:

1. Make changes to AIAConfigurationProperties.xml in $AIA_INSTANCE/AIAMetaData/config.
2. Add the following entry in UpdateMetaDataDP.xml in $AIA_INSTACE/config:

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

3. Navigate to `<AIA_INSTANCE_HOME>/bin` and run the following command as per your platform:
   - On Linux: `source aiaenv.sh`
   - On Windows: `aiaenv.bat`
4. Execute the following command to upload to MDS:
   - On Linux: `ant -f $AIA_HOME/Infrastructure/Install/config/UpdateMetaData.xml`

- On Windows: `ant -f %AIA_ HOME%\Infrastructure\Install\config\UpdateMetaData.xml`

5.7.2.2.3 Deploying OC Provider Changes to Server

Perform the following steps to deploy OC Provider changes to server:

1. Create a custom deployment plan (for example, UpdateOCProviderDP.xml) as follows:

```
<DeploymentPlan component="StudySubjectVisitSyncSCandOC" version="3.0">
<PreInstallScript>
</PreInstallScript>
<Configurations>
<EndpointConfigurator target-server="pips.StudySubjectVisitSyncSCandOC"
dir="${AIA_HOME}">
</EndpointConfigurator>
</Configurations>
<Deployments>
<Composite compositeName="UpdateClinicalStudyOCHealthSciencesProvABCSImpl"
compositedir="${AIA_
HOME}/services/core/OracleClinical/ProviderABCS/UpdateClinicalStudyOCHealthScie
ncesProvABCSImpl" revision="1.0" wlserver="pips.StudySubjectVisitSyncSCandOC"
action="deploy" overwrite="true"/>
</Deployments>
<PostInstallScript>
</PostInstallScript>
</DeploymentPlan>
```

2. Navigate to `<AIA_Instance>/bin` and run the following command as per your platform to configure the installation environment:
   - On Linux: `source aiaenv.sh`
   - On Windows: `aiaenv.bat`
3. At the command prompt, execute the following command to deploy OC Provider:
   - On Linux:
   ```
   ant -f $AIA_
   HOME/Infrastructure/Install/AID/AIAInstallDriver.xml
   -DDeploymentPlan=$AIA_HOME/pips/StudySubjectVisitSyncSCandOC/
   DeploymentPlans/UpdateOCProviderDP.xml -DPropertiesFile=$AIA_
   INSTANCE/config/AIAInstallProperties.xml -l $AIA_HOME/pips/
   StudySubjectVisitSyncSCandOC/DeploymentPlans/UpdateOCProvider.log
   ```
   - On Windows:
   ```
   ant -f %AIA_
   HOME%\Infrastructure\Install\AID\AIAInstallDriver.xml
   -DDeploymentPlan=$AIA_HOME\pips\StudySubjectVisitSyncSCandOC\
   DeploymentPlans\UpdateOCProviderDP.xml -DPropertiesFile=%AIA_
   INSTANCE%\config\AIAInstallProperties.xml -l $AIA_HOME\pips\
   StudySubjectVisitSyncSCandOC\DeploymentPlans\UpdateOCProvider.log
   ```

5.7.2.3 Customizing Enterprise Business Object Fields for Subject Activity Flow From Oracle Clinical to Siebel Clinical

Perform the steps in this section to customize EBO fields for subject activity flow.

You may have added fields to Siebel Clinical and want to send them from Oracle Clinical to Siebel. The following example shows an item that exists in the OC message, DCI book name, that does not exist in the EBO to show all the step. If the item you added exists in the EBO, but did not exist in the out-of-the-box Integration object in Siebel, you will only need to do the parts on the Siebel Provider side.

1. Add the fields, which you have added in Siebel that you want to send from OC, to the Clinical Subject External Integration object using Siebel tools.
2. Create Integration Object for this field using Siebel tools.

5.7.2.3.1 Overview of Steps

1. Create backup of files that will be modified during customization.
2. Add new element to Siebel Clinical xsd and upload to MDS.
3. Add new element to Custom EBO xsd and upload to MDS.
4. Map the DCIBookName to the new element in EBM in the UpdateClinicalStudyOCHealthSciencesReqABCSImpl Custom XSL file.
5. Map the DCIBookName to the new element in EBM in the UpdateClinicalStudySEBLHealthSciencesProvABCSImpl Custom XSL file.
6. Create a custom deployment plan to deploy composites on the soa server.
7. Test the Extension field.

5.7.2.3.2 Steps for Customizing

Step 1: Creating Backups Before Customization

Create a backup copy of the following files on soa_server before customizing:

- <AIA_HOME>/AIAMetaData/AIAComponents/ApplicationObjectLibrary/SiebelClinical/V1/schemas/ClinicalSubjectInternal.xsd
- <AIA_HOME>/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/Custom/EBO/ClinicalStudy/V1/CustomClinicalStudyEBO.xsd
- <AIA_HOME>/services/core/SiebelClinical/ProviderABCS/UpdateClinicalStudySEBLHealthSciencesProvABCSImpl/xsl/XformUpdateClinicalStudyReqMsgtoUpdateUpsert_InputAppReqMsg_Custom.xsl
- <AIA_HOME>/services/core/OracleClinical/RequesterABCS/UpdateClinicalStudyOCHealthSciencesReqABCSImpl/xsl/Xform_ClinicalStudyABMRepMsg_to_HealthSciencesClinicalStudyEBSEBMReqMsg_Custom.xsl

Step 2: Adding New Element to Siebel xsd

In this example, to map DCIBookName element in the Siebel side located at **ClinicalSubject > DCIBookName** after **WithdrawnReason**, perform the following steps:

> **Note:** You must maintain the order of the element. Ensure that you add the new element in the same position you added to the Integration Object.

1. Start Oracle JDeveloper in the local environment.
2. Copy the following file from soa_server file system to the local file system and open it using Oracle JDeveloper.

   <AIA_HOME>/AIAMetaData/AIAComponents/ApplicationObjectLibrary/SiebelClinical/V1/schemas/ClinicalSubjectInternal.xsd

3. To map, expand the element ClinicalSubject, and locate the WithdrawnReason element.
4. Right-click WithdrawnReason and perform insert after element (WithdrawnReason > element).
5. Name this element with the same name as it is created in Siebel tool of type xsd:string and save the changes.
6. Copy the modified xsd from local file system to soa_server file system at the following location:

   <AIA_HOME>/AIAComponents/ApplicationObjectLibrary/SiebelClinical/V1/schemas.
7. From soa_server, perform the following steps to upload the xsd file to MDS:
   1. Update the fileset element in the UpdateMetaDataDP.xml file (at <AIA_INSTANCE_HOME>/config) to point to the location of the xsd file as shown below:

      ```
      <fileset dir="${AIA_HOME}/AIAMetaData">
      <include name="AIAComponents/ApplicationObjectLibrary/
      SiebelClinical/V1/schemas/ClinicalSubjectInternal.xsd"/>
      </fileset>
      ```
   2. Navigate to `<AIA_HOME>/aia_instances/<instance>/bin` and perform the following:

      On Linux: `source aiaenv.sh`

      On Windows: `aiaenv.bat`
   3. Upload to MDS using the following command.

      On Linux: `ant -f $AIA_HOME/Infrastructure/Install/config/UpdateMetaData.sql`

      On Windows: `ant -f %AIA_HOME%\Infrastructure\Install\config\UpdateMetaData.sql`
   4. Wait until you see a build successful result.

Step 3: Adding New Element to the Enterprise Business Object

In this example, to add a field **DCIBookName**, perform the following steps:

Extend the component **ClinicalStudyEBO** in the EBO to include this field.

1. Start Oracle JDeveloper.
2. Copy the <AIA_HOME>/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/Custom/EBO/ClinicalStudy/V1/CustomClinicalStudyEBO.xsd file from soa_server file system to the local file system.
3. Open the following file in the local file system using Oracle JDeveloper.

   <AIA_HOME>/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/Custom/EBO/ClinicalStudy/V1/CustomClinicalStudyEBO.xsd

   Each type name is a place holder for the custom field it extends. For example, CustomClinicalStudySubjectType is the custom field place holders for ClinicalStudySubject business component in the ClinicalStudy EBO.
4. Right-click on **CustomClinicalStudySubjectType** and select **Insert** inside complexType > Sequence.

5. Right-click the new sequence and select **Insert** inside **Sequence > element** to add a new element to the sequence.

6. Name the element **DCIBookName**.

7. Set the data type for the newly added element.

8. Go to the source view of the xsd and find the DCIBookName element.

9. Enter the type of the element. In the following example, the element type is StringType. The CustomClinicalStudySubjectType tag should appear as follows in the source view:

   ```
   <xsd:complexType name="CustomClinicalStudySubjectType">
       <xsd:sequence minOccurs="0">
         <xsd:element name="DCIBookName" type="corecom:StringType" minOccurs="0"/>
       </xsd:sequence>
   </xsd:complexType>
   ```

10. Save the xsd file.

11. Copy the modified xsd from local file system to the soa_server file system at the following location:

    <AIA_HOME>/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Industry/HealthSciences/Custom/EBO/ClinicalStudy/V1/

12. From the soa_server, perform the following steps to upload the xsd file to MDS:

    a. Update <AIA_INSTANCE_HOME>/config/UpdateMetaDataDP.xml to have the following fileset entry:

       ```
       <fileset dir="${AIA_HOME}/AIAMetaData">
       <include name="AIAComponents/EnterpriseObjectLibrary/
       Industry/HealthSciences/Custom/EBO/ClinicalStudy/V1/CustomClinicalStudyEBO.
       xsd"/>
       </fileset>
       ```

    b. Navigate to `<AIA_HOME>/aia_instances/<instance>/bin` and perform the following:

       On Linux: `source aiaenv.sh`

       On Windows: `aiaenv.bat`

    c. Upload to MDS using the following command:

       On Linux: `ant -f $AIA_HOME/Infrastructure/Install/config/UpdateMetaData.sql`

       On Windows: `ant -f %AIA_HOME%\Infrastructure\Install\config\UpdateMetaData.sql`

    d. Wait until you see a build successful result.

Step 4: Verifying if Custom EBO xsd is Uploaded to MDS

Prerequisite

Oracle JDeveloper resource palette must be configured to browse SOA_MDS database for a given database.

1. In Oracle JDeveloper, open the Resource Palette and navigate to the CustomClinicalStudyEBO location.

2. Double-click the files to open and verify the changes.

3. Navigate to core ClinicalStudyEBO.xsd to verify if the changes to the CustomClinicalStudyReportEBO.xsd are reflected correctly in the core EBO. Locate the CustomClinicalStudySubjectType element in the xsd.
4. Expand the element and navigate to the end to locate the Custom element.

Step 5: Mapping the DCIBookName to the New Element in UpdateClinicalStudyOCHealthSciencesReqABCSImpl

Prerequisite

Oracle JDeveloper must be set up with the following configurations:

- Resource palette, to browse SOA_MDS database for a given database.
- jws application, to which jprs can be added.

1. Obtain the following **UpdateClinicalStudySEBLHealthSciencesProvABCSImpl** and **UpdateClinicalStudyOCHealthSciencesReqABCSImpl** from application server to the local Oracle JDeveloper environment.
2. Copy composite projects from the file system of the soa server:

   <AIA_HOME>/services/core/SiebelClinical/ProviderABCS and <AIA_HOME>/services/ core/OracleClinical/RequesterABCS to the local Jdeveloper environment.
3. In Oracle JDeveloper, click **Open**, and navigate to UpdateClinicalStudyOCHealthSciencesReqABCSImpl.jpr.
4. Expand the project and under the xsl folder, open Xform_ClinicalStudyABMRepMsg_to_HealthSciencesClinicalStudyEBSEBMReqMsg_Custom.xsl.
5. In the Oracle JDeveloper design view, map the existing field **DCIBookName** in OC ABM to the newly added field in the ClinicalStudyEBM.
6. Navigate to the **Source** tab.

   Note that the template contains the mapping you created under the "<xsl:template match="/">" element.
7. There are multiple templates provided in the custom xsl. Cut the content enclosed in <coreclinicalstudy:Custom> tag and paste it in the appropriate template.

   For this example, use ClinicalStudySubjectType_ext template.
8. Remove the content enclosing <xsl:template match="/"> tag that is auto generated.
9. Save the changes to custom xsl.

> **Note:** You can modify the xsl using other text editors. Ensure that the name spaces (for example, ebo, ns0) are assigned appropriately.

Step 6: Mapping the DCIBookName to the New Element in UpdateClinicalStudySEBLHealthSciencesProvABCSImpl

Prerequisite

Oracle JDeveloper must be set up with the following configurations:

- Resource palette, to browse SOA_MDS database for a given database.
- jws application, to which jprs can be added.

1. In Oracle JDeveloper, click **Open**, and navigate to **UpdateClinicalStudySEBLHealthSciencesProvABCSImpl.jpr**.
2. Expand the project and under the xsl folder, open XformUpdateClinicalStudyReqMsgtoUpdateUpsert_InputAppReqMsg_Custom.xsl.
3. In the Oracle JDeveloper design view, map the custom field **DCIBookName** in ClinicalStudyEBM to **DCIBookName** in Siebel ABM.
4. Navigate to the Source view of the custom xsl.

   Note that there are multiple templates provided in the custom xsl. Design view adds the mapping created in step 2 to a default template <xsl:template match="/">.
5. Cut and paste the content to the correct template to modify the default template. For example, **< DCIBookName >** should be moved to the **ClinicalSubjectType_ext** template.

   For example,

   ```
   <xsl:template name="ClinicalSubjectType_ext">
   <xsdLocal1:DCIBookName>
   <xsl:value-of select="/ns0:UpdateClinicalStudyEBM/ns0:DataArea/
   ns0:UpdateClinicalStudy/ns0:ClinicalStudySubject/ns0:Custom/coreclinicalstudycu
   st:DCIBookName"/>
   </xsdLocal1:DCIBookName>
   </xsl:template>
   ```
6. Remove the content surrounding <xsl:template match="/"> that is auto generated.
7. Save the changes to the custom xsl.

Step 7: Deploying the Composites

1. On the application server, ensure that **UpdateClinicalStudyOCHealthSciencesReqABCSImpl** and **UpdateClinicalStudySEBLHealthSciencesProvABCSImpl** are updated with customizations.

   This involves copying the modified custom xsls to the application folders of the server $AIA_HOME/services/core/SiebelClinical/ProviderABCS/**UpdateClinicalStudySEBLHealthSciencesProvABCSImpl/xsl** and $AIA_HOME/services/core/OracleClinical/RequesterABCS/**UpdateClinicalStudyOCHealthSciencesReqABCSImpl/xsl**.
2. Open the $AIA_HOME/ pips/StudySubjectVisitSyncSCandOC/DeploymentPlans/StudySubjectVisitSyncSCandOCCustomDP.xml file.
3. Replace <Deployments> </Deployments> with the following:

   ```
   <Deployments>

   <CompositecompositeName="UpdateClinicalStudyOCHealthSciencesReqABCSImpl"
   compositedir="${AIA_HOME}/services/core/OracleClinical/
   RequesterABCS/UpdateClinicalStudyOCHealthSciencesReqABCSImpl" revision="1.0"
   wlserver="pips.StudySubjectVisitSyncSCandOC" action="deploy" overwrite="true"/>

   <Composite compositeName="UpdateClinicalStudySEBLHealthSciencesProvABCSImpl"
   compositedir="${AIA_
   HOME}/services/core/SiebelClinical/ProviderABCS/UpdateClinicalStudySEBLHealthSc
   iencesProvABCSImpl" revision="1.0" wlserver="pips.StudySubjectVisitSyncSCandOC"
   action="deploy" overwrite="true"/>
   ```

```
</Deployments>
```

4. Save the file.

5. Navigate to <AIA_HOME>/aia_instances/<instance>/bin/ and run the following command as per your platform to configure the installation environment:

   - On Linux: `source aiaenv.sh`
   - On Windows: `aiaenv.bat`

6. Run the following custom deployment command for deploying customized artifacts:

   - On Linux:
     ```
     ant -f $AIA_
     HOME/Infrastructure/Install/AID/AIAInstallDriver.xml
     -DDeploymentPlan=$AIA_HOME/pips/StudySubjectVisitSyncSCandOC/
     DeploymentPlans/StudySubjectVisitSyncSCandOCCustomDP.xml
     -DPropertiesFile=$AIA_INSTANCE/config/AIAInstallProperties.xml  -l
     <location and name where you want the log file written>
     ```
   - On Windows:
     ```
     ant -f %AIA_
     HOME%\Infrastructure\Install\AID\AIAInstallDriver.xml
     -DDeploymentPlan=$AIA_HOME\pips\StudySubjectVisitSyncSCandOC\
     DeploymentPlans\StudySubjectVisitSyncSCandOCCustomDP.xml
     -DPropertiesFile=%AIA_INSTANCE%\config\AIAInstallProperties.xml -l
     <location and name where you want the log file written>
     ```

Step 8: Testing the Custom Extension

You must test the custom field you have added.

5.8 Configuration Properties

Table 5–5 describes the configuration properties available for the Study, Subject and Visit Synch: Siebel Clinical –Oracle Clinical PIP:

Table 5–5 Configuration Properties

Service Name	Property Name	Value	Description
UpdateClinicalStudySEBLHealthSciencesReqABCSImpl	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.
	ABCSExtension.PreXformABMtoEBMClinicalStudyABM	false (default), true	Custom extension before the execution of transformation of Application Business Message (ABM) to Enterprise Business Message (EBM).
	ABCSExtension.PreInvokeEBSHealthSciencesClinicalStudyEBSV1EBM	false (default), true	Custom extension just after execution of transformation of ABM to EBM and before the call to the Enterprise Business Service (EBS).
	Routing.HealthSciencesClinicalStudyEBS.UpdateClinicalStudy.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.
	Routing.HealthSciencesClinicalStudyEBS.UpdateClinicalStudy.CAVS.EndpointURI	http://<<soaservername>>:<<port>>/AIAValidationSystemServlet/asyncresponseSimulator	End point URL for CAVS, when RouteToCAVS property is set to true.
UpdateClinicalStudySEBLHealthSciencesProvABCSImpl	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.
	ABCSExtension.PreXformEBMtoABMClinicalStudyEBM	false (default), true	Customer extension just before the execution of the EBM to ABM transformation.
	Routing.Upsert.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.
	Routing.Upsert.CAVS.EndpointURI	http://<<soaservername>>:<<port>>/AIAValidationSystemServlet/syncresponseSimulator	End point URL for CAVS, when RouteToCAVSproperty is set to true.
	Routing.Upsert.SEBLCLIN_01.EndpointURI	${participatingapplications.sc.server.InternetProtocol}${participatingapplications.sc.server.host}:${participatingapplications.sc.server.port}/eai_ ${participatingapplications.sc.server.Language}/start.swe?SWEExtSource=SecureWebService&amp;SWEExtCmd=Execute&amp;WSSOAP=1	End point URL for Siebel Clinical participating application.

Table 5–5 (Cont.) Configuration Properties

Service Name	Property Name	Value	Description
	Routing.Upsert.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.
	ABCSExtension.PreInvokeABSUpsert_InputABM	false (default), true	Customer extension just before the invocation of Upsert call to Siebel.
	ABCSExtension.PostInvokeABSUpsert_InputABM	false (default), true	Customer extension just after invoking Upsert call to Siebel.
UpdateClinicalStudyOCHealthSciencesReqABCSImpl	Default.SystemID	OC_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.
	ABCSExtension.PreXformABMtoEBMClinicalStudyABM	false (default), true	Custom extension before the execution of transformation of ABM to EBM.
	Routing.ClinicalStudyEBSV1.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.
	Routing.ClinicalStudyEBSV1.CAVS.EndpointURI	http://<<soaservername>>:<<port>>/AIAValidationSystemServlet/syncresponseSimulator	End point URL for CAVS, when RouteToCAVS Property is set to true.
	ABCSExtension.PreInvokeEBSHealthSciencesClinicalStudyEBSEBM	false (default), true	Custom extension before the invocation of the EBS.

Table 5–5 (Cont.) Configuration Properties

Service Name	Property Name	Value	Description
UpdateClinicalStudyOCHealthSciencesProvABCSImpl	Default.SystemID	OC_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.
	ABCSExtension.PreXformEBMtoABMClinicalStudyEBM	false (default), true	Customer extension just before the execution of the EBM to ABM transformation.
	Routing.StudySiteServiceSoapHttpPort.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.
	Routing.StudySiteServiceSoapHttpPort.CAVS.EndpointURI	http://<<soaservername>>:<<port>>/AIAValidationSystemServlet/syncresponseSimulator	End point URL to invoke the Oracle Clinical Study Site web service.
	Routing.SiteServiceSoapHttpPort.OC_01.EndpointURI	http://${participatingapplications.ocrdc.server.http.host}:${participatingapplications.ocrdc.server.http.port}/OracleClinical-context-root/SiteServiceSoapHttpPort	End point URL to invoke the Oracle Clinical Site web service.
	Routing.SiteServiceSoapHttpPort.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.
	Routing.InvestigatorServiceSoapHttpPort.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.
	Routing.InvestigatorServiceSoapHttpPort.CAVS.EndpointURI	http://<<soaservername>>:<<port>>/AIAValidationSystemServlet/syncresponseSimulator	End point URL for CAVS, when RouteToCAVS property is set to true.
	Routing.InvestigatorServiceSoapHttpPort.OC_01.EndpointURI	http://${participatingapplications.ocrdc.server.http.host}:${participatingapplications.ocrdc.server.http.port}/OracleClinical-context-root/InvestigatorServiceSoapHttpPort	End point URL to invoke the Oracle Clinical Investigator web service.
	Routing.InvestigatorServiceSoapHttpPort.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.

Table 5–5 (Cont.) Configuration Properties

Service Name	Property Name	Value	Description
	ABCSExtension.PreInvokeABSStudySiteABM	false (default), true	Customer extension just before the invocation of the Study Site Service in Oracle Clinical.
	ABCSExtension.PostInvokeABSStudySiteABM	false (default), true	Customer extension just after invoking the Study Site Service in Oracle Clinical.
	CustomXForm.InvestigatorABM	false (default), true	If set to true, it invokes the custom transformation file for Investigator ABM (Custom_xForm_CreateInvestigatorABM.xsl).
	CustomXForm.SiteABM	false (default), true	If set to true, it invokes the custom transformation file for Site ABM (Custom_xFrom_CreateSiteABM.xsl).
	CustomXForm.StudySiteABM	false (default), true	If set to true, it invokes the custom transformation file for Study Site ABM (Custom_xForm_CreateStudySiteABM.xsl).
AIASessionPoolManager	all_hosts	SEBLCLIN_01 NOSERVER	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 Host ID value.
	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.
	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.
	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.

Table 5–5 (Cont.) Configuration Properties

Service Name	Property Name	Value	Description
	SEBLCLIN_01.EndpointURI	Populated by values specified in the ConfigurationWizard. (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.
	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.
	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.
	SEBLCLIN_01.PredictExpiration_Idle	780000	Indicates the maximum time in milliseconds that a session token can be idle before expiring. Oracle recommends you to set this value to a value lower than the actual maximum idle time configured for the application web server. This value is recommended 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.
	SEBLCLIN_01.PredictExpiration_Age	82800000	Indicates the maximum age in milliseconds that a session token can reach before expiring. Oracle recommends you to 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 1or 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.
	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.
	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.

5.9 Updating Server Information for Siebel Clinical and Oracle Clinical

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.

Also, while running the configuration wizard, if you have entered a wrong port number or want to move the Siebel application to another machine, you can modify the file and upload these changes to MDS.

To modify information about Siebel Clinical, perform the following steps:

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

   a. Navigate to `$AIA_INSTANCE/AIAMetaData/config` directory.

   b. Open the file AIAConfigurationProperties.xml.

   c. Search for SEBLCLIN_01.EndpointURI.

   d. Replace the host name and port, if they have changed.

2. Upload the updated AIAConfigurationProperties.xml file to MDS. Perform the following 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. Navigate to `<AIA_HOME>/aia_instances/<instance>/bin` and perform the following:

   On Linux: `source aiaenv.sh`

   On Windows: `aiaenv.bat`

   c. Upload to MDS using the following command.

   On Linux: `ant -f $AIA_HOME/Infrastructure/Install/config/UpdateMetaData.sql`

   On Windows: `ant -f %AIA_HOME%\Infrastructure\Install\config\UpdateMetaData.sql`

   d. Wait until you see a build successful result.

5.9.1 Changing Passwords Specified During Install

You can change the passwords specified at the time of AIA installation post-install. To change passwords, AIA provides a utility called UpdateStore. AIA stores the passwords in keystore. This utility modifies the existing passwords in the keystore.

To run this utility, the AIAIntallProperties.xml properties file should exist in the corresponding AIAHOME/AIA_INSTANCE/<Instance name>/config folder and the password being modified must exist in the file.

Note: This utility only updates encrypted passwords in AIA installation. It will not change passwords of the SOA server or the database schemas.

To Modify Passwords:

1. Navigate to `<AIA_Instance>/bin` and run the following command as per your platform to configure the installation environment:
   - On Linux: `source aiaenv.sh`
   - On Windows: `aiaenv.bat`
2. Navigate to `AIA_HOME/util/`.
3. Execute the following command:

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

   The Update AIA keystore screen is displayed.
4. Enter the existing user name, existing password, new user name, and new password and the Xpath of the password which you want to change.

   Note: This utility modifies the user name and the password in the AIAInstallProperties.xml. This does not synchronize the AIAInstallProperties.xml in the MDS. You need to do it manually.

   If you want to change the password under the o2c tag, enter the following values in the screen that appears.

   Existing user name:

   Existing password:

   New user name:

   New password:

   XPath: `/properties/participatingapplications/sc/server/eai/password`

   For more information, see *Oracle® Fusion Middleware Installation and Upgrade Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.4.0) Guide*.

5.9.2 Updating Server or Password Information for Oracle Clinical

If the password, host name, or port of OC changes, it impacts OracleClinicalCoreDS and UpdateClinicalStudyOCDS data source. In such a scenario, perform the following steps:

1. Log in to the Weblogic console.
2. On left-hand side navigation hierarchy, navigate to soa_domain > Services > Data Sources
3. Click **OracleClinicalCoreDS**.
4. Click the **Connection Pool** tab.
5. Change the password, host name, and port and click **Save**.

6. Change the password, host name, and port for UpdateClinicalStudyOCDS following the steps from 3 to 5.

5.10 Error Handling

Based on the roles defined for the services, email notifications are sent if a service errors out.

For more information about Oracle Clinical errors, see the Oracle Clinical documentation.

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."

Table 5–6 Error Handling

Problem	Solution
Instances are not getting generated in the AIA layer when Siebel user selects the **Activate for Synchronization** check box.	Perform one of the following: ■ Check if workflow monitor agent is up and running. ■ Check if JNDI properties in <dir>:/WLSJMS has the correct SOA server URL.
UpdateClinicalStudySEBLHealthSciencesReqABCSImplProcess is in faulted state with the following error: The value entered for the State '<to be replace>' does not exist in '<SEBLCLIN_01>,' column for the STATE DVM. Please add the value to the STATE DVM.	Ensure that STATE.dvm has appropriate state codes listed in the SEBLCLIN_01 column. You can add or modify the DVM values using the following: `http://soaserverhost:soaserverport/soa/composer`
UpdateClinicalStudyOCHealthSciencesProvABCSImplProcess is in faulted state with the following error: The value entered for the State '<to be replace>' does not exist in the <OC_01> column for the STATE DVM. Please add the value to the STATE DVM.	Ensure that STATE.dvm has appropriate state codes listed in the OC_01 column. You can add or modify the DVM values using the following: `http://soaserverhost:soaserverport/soa/composer`
Email notifications are not coming.	Perform one of the following: ■ Ensure the email addresses are entered correctly in the Oracle User Messaging Service standalone user interface: `http://<soa-host>:<soa-port>/sdpmessaging/userprefs-ui` ■ Ensure that the messaging channel name you enter corresponds to an error handling user role name you have created. When logging into Oracle User Messaging Service, use the credentials of user role name not the credentials of WebLogic admin user.

Updates to AIAConfigurationProperties.xml are not reflected in the actual business flow.	Perform one of the following:
	■ Ensure that any changes to AIAConfigurationProperties.xml are uploaded to MDS.
	■ Update file set element at `$AIA_INSTANCE/config/UpdateMetaDataDP.xml` to point to the location of AIAConfigurationProperties.xml. Use the following commands to upload the file to the MDS:
	For Windows:
	`%AIA_INSTANCE%\bin\aiaenv.bat`
	`ant -f \%AIA_HOME%\Infrastructure\Install\config\UpdateMetaData.xml`
	For Linux:
	`source $AIA_INSTANCE/bin/aiaenv.sh`
	`ant -f /$AIA_HOME/Infrastructure/Install/config/UpdateMetaData.xml`

Table 5–6 (Cont.) Error Handling

Problem	Solution
Instance is created in the OC AQ clin_study_queue_tbl table but not picked by UpdateClinicalStudyOCAQConsumer.	Check if dequeue is enabled on OC AQ. If not, you can enable it by executing the following command on the RXC schema of OC DB: `EXECUTE DBMS_AQADM.START_QUEUE(queue_name=> 'RXC.CLINICAL_STUDY_QUEUE', enqueue=> true, dequeue => true);`
Failed to invoke end component oracle.pharma.oc.model.services.server.webservice.SiteServer (POJO), operation=createSite. ■ Failed to invoke method ■ OC-68540 - ERR - State not found	Check that the state in the Site address is defined as a region of type state within a region of type country for the country passed in the site address.
The requested visit could not be located in the database.	Check that the activity name defined in OC matches exactly with the clinical item defined in SC.
Error running subprocess 'SWI LS Clinical Subject Inbound - Activity' at step 'Send to Activity Flow'.(SBL-BPR-00183)	This error occurs when a blank completion date is sent for an activity which was already marked as completed in SC. An investigator or site may have been paid for an activity or visit that did not occur. You must manually adjust this in SC.

6 Configuring Multiple Oracle Clinical Database Instances

This chapter discusses:

- Section 6.1, "Overview"
- Section 6.2, "Configuring Additional Oracle Clinical Instances"

6.1 Overview

If you have more than one instances of Oracle Clinical (OC), you can use the following guidance in configuring multiple OC environments and conditional routing of messages from Siebel Clinical (SC) to either of the OC instances.

6.2 Configuring Additional Oracle Clinical Instances

Perform the following steps to configure additional OC instances:

1. Update AIA System Registry. Perform the following steps:

   a. Navigate to the AIA Console URL: http://<server name>:<portnumber>/AIA.

   b. Log in with the admin user name.

   c. Navigate to **Setup** > **Systems**.

   d. Click **Create** to create an additional row for each instance of OC to be configured. In this example, one additional instance of OC named OC_02 is added (assuming the original instance of OC is OC_01).

   e. Enter values in the following mandatory fields:

Table 6–1 Mandatory Fields

Field	Description
Internal ID	The value should be same as the value returned when you run the following query in second Oracle Clinical database instance: `select *from global_name`
System Code	A logical value. For example, OC_02
System Type	Enter Oracle Clinical.

2. Modify the DVM and Xref entries.

The Domain Value Maps (DVMs) and Cross References (XRefs) are stored under **$AIA_HOME/AIAMetaData/** in dvm and xref folders respectively. You need to modify the following DVMs and XRefs:

- STATE.dvm
- COUNTRY.dvm
- CLINICALSTUDYSUBJECT_STATUS.dvm
- CLINICALSTUDYSTUDYSITE.xref
- CLINICALSTUDYSITE.xref
- CLINICALSTUDY_CLINICALSTUDYSUBJECTID.xref
- CLINICALSTUDYINVESTIGATOR.xref

In the XRefs, add an additional column for OC_02 (second instance of OC and any other OC instances, as you may require). In the DVMs, add additional column name, and alter to add the column values appropriately.

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

COUNTRY.dvm

```
<?xml version='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="OC_01"/>
<column name="OC_02"/>
<column name="SEBL_01"/>
<column name="ARGUS_01"/>
</columns>
<rows>
<row>
<cell>Please populate</cell>
<cell/>
<cell/>
<cell/>
<cell/>
<cell/>
</row>
</rows>
</dvm>
```

CLINICALSTUDYINVESTIGATOR.xref

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Upgraded by Xref Upgrade Utility 1.0 -->
<xref xmlns="http://xmlns.oracle.com/xref">

<table name="CLINICALSTUDYINVESTIGATOR">
<description></description>
<columns>
<column name="COMMON"/>
<column name="SEBLCLIN_01"/>
<column name="OC_01"/>
<column name="OC_02"/>
</columns>
```

```
</table>
</xref>
```

To upload updated DVMs and XRefs to MDS, perform the following:

a. Navigate to the `$AIA_INSTANCE/config` directory.

b. Copy `UpdateMetaDataDp.xml` to a backup file.

c. Edit `UpdateMetaDataDp.xml` 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="xref/CLINICALSTUDYINVESTIGATOR.xref"/>
<include name="dvm/COUNTRY.dvm"/>
</fileset>
</UpdateMetadata>
</Configurations>
</DeploymentPlan>
```

d. Navigate to the `$AIA_INSTANCE/bin` directory.

e. Execute the following command:

For Linux: `source aiaenv.sh`

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

For Windows: `aiaenv.bat`

```
ant -f %AIA_HOME%\Infrastructure\Install\config\UpdateMetaData.xml
```

f. Wait till you see a build successful result.

Note: MetaData gets deleted when you uninstall Foundation Pack. For specific instruction on how to clean the MDS after you uninstall Foundation Pack, see *Oracle Fusion Middleware Installation and Upgrade Guide for Oracle Application Integration Architecture Foundation Pack, Cleaning the MDS*.

Note: 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 the SOA Composer.

3. Update AIAInstallProperties.xml.

a. Update the AIAInstallProperties.xml file in the $AIA_INSTANCE_HOME/config to add the following OC_02 database entry as a child node of /properties/participatingapplications.

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

```
<oc2>
<version>4.6.0.52</version>
<db>
<jdbc-url>{Connect-url}</jdbc-url>
<host>{host-name}</host>
<port>{port}</port>
```

```
<internal_id>{SID}</internal_id>
<username>{username}</username>
<password>{password}</password>
</db>
<server>
<http> <host>{soa server host name}</host>
<port>{soa server port}</port>
</http>
</server>
</oc2>
```

b. Encrypt the AIAInstallProperties password by running the following commands either through XTerm or VNC viewer:

– `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:

– User name: rxc

– Password: for rxc schema of OC_02 database

– Path: Xpath of the OC_02 database password field in the AIAInstallProperties.xml. For example: /properties/participatingapplications/oc2/db/password

This updates the encrypted password in the MDS as well as in the $AIA_INSTANCE_HOME/config file.

4. Create OC_02 database data source.

Use the Oracle WebLogic console to create the OC_02 DB database data source. As an example, a data source OracleClinicalCore2DS is created with jndi value as jdbc/OracleClinicalCore2DS pointing to the OC_02 database. You should deploy your new data source to the managed SOA server instance.

a. Log in to the WebLogic 11g server console as the WebLogic user.

b. Click **Services** under Domain Structure (in the left-side of the screen), and then **Data Sources**.

The system displays all existing data sources. The New button becomes active.

c. Click **New**.

The screen displays three options.

d. For non-RAC:

i. Select the **Generic Data Source** option.

ii. In the Create Screen, name the Datasource as `OracleClinicalCore2DS`.

iii. In the JNDI Name column, enter `jdbc/OracleClinicalCore2DS`.

Note: Do not change any other default values.

iv. Click **Next**.

v. Click **Next**.

vi. Click **Next** and enter the database name, host name, and port of your Oracle Clinical 2 database instance. You must enter the user name as `rxa_ws` and password for this user.

vii. Click **Next**.

viii. Click **Test Configuration**. Ensure the connectivity is established.

ix. Click **Next**.

x. Select your SOA server as the target. For example, `soa_server1`.

xi. Click **Finish**.

The Data Source is created and deployed on the target server.

e. For RAC:

i. Select the **Multi Data Sources** option.

ii. In the Create Screen, name the Datasource as `OracleClinicalCore2DS`.

iii. In the JNDI Name column, enter `jdbc/OracleClinicalCore2DS`.

Note: Do not change any other default values.

iv. Click **Next**.

v. Select the SOA server and click **Next**.

vi. Retain the default, **Non-XA Driver**, and click **Next**.

vii. Click **Create New Data Source**.

viii. In the Create Screen, name the Datasource as `OracleClinicalCore2DS-rac0`.

where, `rac0` is the first RAC node and the number should be incremented for each node.

ix. In the JNDI Name column, enter `jdbc/OracleClinicalCore2DS-rac0`.

where, `rac0` is the first RAC node and the number should be incremented for each node.

Note: Do not change any other default values.

x. Click **Next**.

xi. Enter `Oracle` as the database type. For the driver type, enter `Oracle Driver (Thin)` for RAC server-Instance connection Version 10, 11.

xii. Deselect **Supports Global Transactions** and click **Next**.

xiii. Click **Next** and enter the database name, host name, and port of your Oracle Clinical 2 database Instance. You must enter the user name as `rxa_ws` and password for this user.

xiv. Click **Next**.

xv. Test the configuration and click **Next**.

xvi. Select your SOA server as the target. For example, `soa_server1`.

xvii. Click **Finish**.

The Data Source is created and deployed on the target server.

xviii. Repeat steps e.vii to e.xvi for each node of the RAC instance.

f. Ensure the RAC data source are attached to the Multi Data Source created in step e.a.

g. Check the list of the Data Sources whether this data source is created successfully.

5. Customize the OracleClinical.ear.

Locate the original OracleClinical.ear that is shipped with Oracle Clinical. This .ear is installed as OracleClinical application in the Oracle WebLogic console.

a. Extract the OracleClinical.ear file to any location (for example: D:\OracleClinicalEar folder). It contains a META-INFfolder and an OracleClinical-web.war file.

b. Open the OracleClinical-web.war using WinZip or similar tool.

c. Navigate to OracleClinical-web.war\WEB-INF\classes\oracle\pharma\oc\model\services\common\.

d. Open bc4j.xcfg for editing and change the JDBC DataSource to the jndi location of the Data Source defined in the step 4. You need to make this change at four places in the file. A sample configuration change is shown below:

Original:

<Custom JDBCDataSource="jdbc/OracleClinicalCoreDS"/>

Change to:

<Custom JDBCDataSource="jdbc/OracleClinicalCore2DS"/>

e. Navigate to D:\OracleClinicalEar\META-INF folder. Modify the application.xml and change the context-root value to a new value that will be used for the second OC instance. Following is a sample file code:

```
<?xml version ='1.0' encoding ='windows-1252'?> <!DOCTYPE application
PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application 1.3//EN"
"http://java.sun.com/dtd/application_1_3.dtd"> <application>
<display-name>OracleClinicalSecond-WS</display-name> <module> <web>
<web-uri>OracleClinical-web.war</web-uri>
<context-root>OracleClinical-second-context-root</context-root> </web>
</module> </application>
```

f. Re-package the contents of OracleClinical folder to an .ear file. You can either use WinZip or a similar tool or the following command from the OracleClinical directory.

i. Execute the following command to zip the OracleClinical-web file:

```
jar -cvf OracleClinical-web.war META-INF WEB-INF
```

ii. Delete the OracleClinical-web directory.

The contents of D:\OracleClinicalEar folder are:

META-INF

OracleClinical-web.war

iii. Execute the following command to zip the D:\OracleClinicalEar folder:

```
jar -cvfm OracleClinical.ear META-INF/MANIFEST.MF
META-INF/application.xml OracleClinical-web.war.
```

The OracleClinical.ear file is generated.

The jar command requires the jre in the system path. If the command is not recognized, ensure to include the jre/bin folder in the system path. (<Oracle_Home>/Middleware/<jdk>/jre/bin).

For example, use the following commands to set the system path:

Unix: `export PATH=<absolute path for jdk/jre/bin>:$PATH`

Windows: `set PATH=< absolute path for jdk/jre/bin>;%PATH%`

g. Deploy the newly created .ear file to the Oracle WebLogic server.

i. Navigate to the Oracle WebLogic console and click **Deployments** in the left navigation bar.

ii. Click **Install**.

iii. In the **Path** field, provide the path where the new ear file is uploaded and click **Next**.

iv. Select the **Install this deployment as an application** and click **Next**.

v. Select the SOA server instance and click **Next**.

vi. Change the deployment name so that it is unique in the deployment list on the WebLogic server console.

vii. Test the deployment by accessing one of the following URLs:

http://{server-name}:{port}/{context-root as defined in step 'c'}/StudySiteServiceSoapHttpPort

http://{server-name}:{port}/{context-root as defined in step 'c'}/SiteServiceSoapHttpPort

http://{server-name}:{port}/{context-root as defined in step 'c'}/InvestigatorServiceSoapHttpPort

vii. Verify that WDSL link is accessible.

where,

server-name refers to the Fusion Middleware (FMW) server name

port refers to the SOA port

viii. Click **Finish**.

6. Modify the service configurations.

a. Navigate to ${AIA_HOME}/aia_instances/inst1/AIAMetaData/config and open AIAConfigurationProperties.xml for modification.

b. Modify the service configuration for OC Provider ABCS.

The service configurations contain the WSDL to locate the services of various OC instances. Each instance of OC configured in Systems Registry in AIA console has an entry in the service configuration section.

c. Look for Service Configuration section of UpdateClinicalStudyOCHealthSciencesProvABCSImpl.

d. There is a default System ID defined. This System ID is used in case there is no Target System defined in the EBM message header. This may happen when the routing rules specifically do not set any Target ID, or no routing rules match. Set this according to your requirements.

e. There are configurations for each of the OC Web services for investigator, site, and study site (locate and inspect properties such as Routing.StudySiteServiceSoapHttpPort.OC_01.EndpointURI).

You need to add similar properties for each of the OC instance.

See the highlighted sections in the following sample configuration for reference. For each of the OC instances, the WSDL to locate these services should be unique. Note that the system code that is set in the AIA System Console is used to differentiate the OC instance for which the service should be called. Also, the WSDL location should reflect the correct context root as defined for the corresponding OC instance in the step 5.e.

The following is a sample modified configuration:

```
<ServiceConfiguration
serviceName="{http://xmlns.oracle.com/ABCSImpl/OracleClinical/Industry/Heal
thSciences/UpdateClinicalStudyOCHealthSciencesProvABCSImpl/V1}UpdateClinica
lStudyOCHealthSciencesProvABCSImpl">
<Property name="Default.SystemID">OC_01</Property>
<Property
name="ABCSExtension.PreXformEBMtoABMClinicalStudyEBM">false</Property>
<Property
name="ABCSExtension.PostXformABMtoEBMClinicalStudyEBM">false</Property>
<Property
name="Routing.StudySiteServiceSoapHttpPort.RouteToCAVS">false</Property>
<Property name="Routing.StudySiteServiceSoapHttpPort.CAVS.EndpointURI">
http://{server-name}:{soa server
port}/AIAValidationSystemServlet/syncresponsesimulator</Property>
<Property name="Routing.StudySiteServiceSoapHttpPort.OC_01.EndpointURI">
http://{server-name}:{soa server
port}/OracleClinical-context-root/StudySiteServiceSoapHttpPort</Property>
<Property name="Routing.StudySiteServiceSoapHttpPort.OC_
02.EndpointURI">http://{server-name}:{soa server
port}/OracleClinical-second-context-root/StudySiteServiceSoapHttpPort</Prop
erty>
<Property
name="Routing.StudySiteServiceSoapHttpPort.MessageProcessingInstruction.Env
ironmentCode">PRODUCTION</Property>
<Property
name="Routing.SiteServiceSoapHttpPort.RouteToCAVS">false</Property>
<Property name="Routing.SiteServiceSoapHttpPort.CAVS.EndpointURI">
http://{server-name}:{soa server
port}/AIAValidationSystemServlet/syncresponsesimulator</Property>
<Property name="Routing.SiteServiceSoapHttpPort.OC_01.EndpointURI">
http://{server-name}:{soa server
port}/OracleClinical-context-root/SiteServiceSoapHttpPort</Property>
<Property name="Routing.SiteServiceSoapHttpPort.OC_02.EndpointURI">
http://{server-name}:{soa server
port}/OracleClinical-second-context-root/SiteServiceSoapHttpPort</Property>
<Property
name="Routing.SiteServiceSoapHttpPort.MessageProcessingInstruction.Environm
entCode">PRODUCTION</Property>
<Property
name="Routing.InvestigatorServiceSoapHttpPort.RouteToCAVS">false</Property>
<Property name="Routing.InvestigatorServiceSoapHttpPort.CAVS.EndpointURI">
http://{server-name}:{soa-server-port}/AIAValidationSystemServlet/syncrespo
nsesimulator</Property>
<Property name="Routing.InvestigatorServiceSoapHttpPort.OC_01.EndpointURI">
http://{server-name}:{soa server
port}/OracleClinical-context-root/InvestigatorServiceSoapHttpPort</Property
```

```
>
<Property name="Routing.InvestigatorServiceSoapHttpPort.OC_02.EndpointURI">
http://{server-name}:{soa server
port}/OracleClinical-second-context-root/InvestigatorServiceSoapHttpPort</P
roperty>
<Property
name="Routing.InvestigatorServiceSoapHttpPort.MessageProcessingInstruction.
EnvironmentCode">PRODUCTION</Property>
<Property name="ABCSExtension.PreInvokeABSStudySiteABM">false</Property>
<Property name="ABCSExtension.PostInvokeABSStudySiteABM">false</Property>
</ServiceConfiguration>
```

f. Once the changes to AIAConfigurationProperties file are complete, upload the file to MDS. See Section 5.7.2.2.2.

The following is a sample UpdateMetaDataDP.xml:

```
<?xml version="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>
```

7. Create the lookup table.

To route the message from Siebel to different Oracle Clinical databases, you need to define the correct routing rule. This example routes the message for a given study to a particular OC instance. This requires creating a lookup table. If you are using a different criteria, name the column accordingly.

a. Run the following script in the AIA schema of the AIA database:

```
CREATE TABLE RouteLookup (Study VARCHAR2(50) NOT NULL UNIQUE,
TargetID VARCHAR2(50) NOT NULL)
```

b. Insert Study and TargetID mappings into this table using SQL insert statements. The following is an example:

```
INSERT INTO RouteLookup (Study, TargetID) VALUES ('TESTSTUDY01',
'OC_01');
```

```
COMMIT;
```

c. Note the values in the insert statements. The first value is the study name as per the target OC system. The second value is the target instance name, which must be same as one of the system code values entered in the AIA Console System Registry screen in the step 2.

8. Create the XSL file. The following is a sample XSL file. This code maps the study in the input EBM message to the Study column in the Routelookup table and fetches the corresponding Target ID. In case the Study is not mapped, the Target ID is set to the default Target ID specified for OC Provider in the AIAServiceConfiguration file.

```
<?xml version="1.0" encoding="UTF-8" ?>
<?oracle-xsl-mapper
 <!-- SPECIFICATION OF MAP SOURCES AND TARGETS, DO NOT MODIFY. -->
 <mapSources>
 <source type="WSDL">
 <schema
location="oramds:/apps/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibra
ry/Industry/HealthSciences/EBO/ClinicalStudy/V1/HealthSciencesClinicalStudyEBSV
1.wsdl"/>
 <rootElement name="UpdateClinicalStudyEBM"
namespace="http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/ClinicalStudy/V1"
```

```
/>
 </source>
 </mapSources>
 <mapTargets>
 <target type="WSDL">
 <schema
location="oramds:/apps/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibra
ry/Industry/HealthSciences/EBO/ClinicalStudy/V1/HealthSciencesClinicalStudyEBSV
1.wsdl"/>
 <rootElement name="UpdateClinicalStudy"
namespace="http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/ClinicalStudy/V1"
/>
 </target>
 </mapTargets>
 <!-- GENERATED BY ORACLE XSL MAPPER 11.1.1.4.0(build 110106.1932.5682) AT [TUE
MAY 31 17:40:32 IST 2011]. -->
?>
<xsl:stylesheet version="1.0"
 xmlns:svcdoc="http://xmlns.oracle.com/Services/Documentation/V1"
 xmlns:ns1="http://xmlns.oracle.com/EnterpriseServices/ClinicalStudy/V1"

xmlns:aia="http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.
AIAFunctions"
 xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"

xmlns:xp20="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.fun
ctions.Xpath20"

xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
 xmlns:ebs="http://xmlns.oracle.com/EnterpriseServices/Core/ClinicalStudy/V1"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:ebo="http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/ClinicalStudy/V1"

xmlns:coreclinicalstudyabcs="http://xmlns.oracle.com/ABCSImpl/OracleClinical/In
dustry/HealthSciences/UpdateClinicalStudyOCHealthSciencesProvABCSImpl/V1"
 xmlns:bpm="http://xmlns.oracle.com/bpmn20/extensions"

xmlns:ns0="http://xmlns.oracle.com/EnterpriseObjects/Core/Custom/EBO/ClinicalSt
udy/V1"
 xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"

xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:ns3="http://xmlns.oracle.com/EnterpriseObjects/Core/Custom/Common/V2"
 xmlns:ora="http://schemas.oracle.com/xpath/extension"

xmlns:socket="http://www.oracle.com/XSL/Transform/java/oracle.tip.adapter.socke
t.ProtocolTranslator"
 xmlns:ns2="http://schemas.xmlsoap.org/ws/2003/03/addressing"

xmlns:mhdr="http://www.oracle.com/XSL/Transform/java/oracle.tip.mediator.servic
e.common.functions.MediatorExtnFunction"

xmlns:oraext="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.f
unctions.ExtFunc"

xmlns:dvm="http://www.oracle.com/XSL/Transform/java/oracle.tip.dvm.LookupValue"
 xmlns:hwf="http://xmlns.oracle.com/bpel/workflow/xpath"
xmlns:med="http://schemas.oracle.com/mediator/xpath"
```

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns:ids="http://xmlns.oracle.com/bpel/services/IdentityService/xpath"
 xmlns:ns5="urn:oasis:names:tc:xacml:2.0:policy:schema:cd:04"
 xmlns:xdk="http://schemas.oracle.com/bpel/extension/xpath/function/xdk"

xmlns:xref="http://www.oracle.com/XSL/Transform/java/oracle.tip.xref.xpath.XRef
XPathFunctions"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:bpelext="http://schemas.oracle.com/bpel/extension"
 xmlns:corecom="http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2"
 xmlns:ns6="http://xmlns.oracle.com/Services/Documentation/V1.0"
 xmlns:ns4="urn:oasis:names:tc:xacml:2.0:context:schema:cd:04"
 xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
exclude-result-prefixes="xsi xsl svcdoc ns1 ebs ebo ns0 soap ns3 ns2 ns5 xsd
corecom ns4 coreclinicalstudyabcs plnk bpelext ns6 aia bpws xp20 bpel bpm ora
socket mhdr oraext dvm hwf med ids xdk xref ldap">

<xsl:variable name="TargetSystemID"

select="/ebo:UpdateClinicalStudyEBM/corecom:EBMHeader/corecom:Target/corecom:ID
"/>
 <xsl:variable name="StudyID"

select="/ebo:UpdateClinicalStudyEBM/ebo:DataArea/ebo:UpdateClinicalStudy/coreco
m:Identification/corecom:ID"/>
 <xsl:template match="@*|node()">
 <xsl:copy>
<xsl:copy-of select="@*"/>
 <xsl:apply-templates/>
 </xsl:copy>
 </xsl:template>
<xsl:template
match="/ebo:UpdateClinicalStudyEBM/corecom:EBMHeader/corecom:Target/corecom:ID"
>
<xsl:choose>
<xsl:when
test="oraext:lookup-table('ROUTELOOKUP','STUDY',$StudyID,'TARGETID','jdbc/AIADa
taSource')">
<corecom:ID><xsl:value-of
select="oraext:lookup-table('ROUTELOOKUP','STUDY',$StudyID,'TARGETID','jdbc/AIA
DataSource')"/></corecom:ID>
</xsl:when>
<xsl:otherwise>
<corecom:ID><xsl:value-of
select="aia:getServiceProperty('{http://xmlns.oracle.com/ABCSImpl/OracleClinica
l/Industry/HealthSciences/UpdateClinicalStudyOCHealthSciencesProvABCSImpl/V1}Up
dateClinicalStudyOCHealthSciencesProvABCSImpl','Default.SystemID',true())"/>
</corecom:ID>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:stylesheet>
```

a. Create an XSL file Xform_UpdateClinicalStudyEBM_to_UpdateClinicalStudy.xsl with the above code in the ${AIA_HOME}/pips/StudySubjectVisitSyncSCandOC/EBS/ClinicalStudy/HealthSciencesClinicalStudyEBS/xsl directory.

You may select another file name, but ensure that the name matches the path given in the <transform> clause in the EBS routing as explained in the step 10.

If mapping is done on an attribute other than study ID, a corresponding Xpath should be used in the definition of variable StudyID in the XSL code.

If the column name was changed from Study to any other value in the route lookup table (Step 7), an appropriate column name should be used in second parameter passed to oraext:lookuptable function in the sample code.

9. Modify the routing rules in EBS.

Open ${AIA_HOME}/pips/StudySubjectVisitSyncSCandOC/EBS/ClinicalStudy/HealthSciencesClinicalStudyEBS/HealthSciencesClinicalStudyEBS.mplan in any text editor. You may need to change the file permission from read-only to writeable.

a. Locate <operation name="UpdateClinicalStudy">. There are various routings defined case wise. Change the code as indicated for Case_1 of this operation.

Important: You must not copy and paste the whole code from the following code. You must add the section to the correct position.

Original Code:

```
<case name="Case_1_HealthSciencesClinicalStudyEBS.UpdateClinicalStudy"
executionType="direct">
<condition language="xpath"
xmlns:ebo="http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/ClinicalStudy
/V1"
xmlns:corecom="http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2"
xmlns:aia="http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xp
ath.AIAFunctions">

<expression>($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/corecom:
EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRO
DUCTION' or
not($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/corecom:EBMHeader
/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and
($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/corecom:EBMHeader/co
recom:Target/corecom:ApplicationTypeCode='Oracle Clinical' or
((not($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/corecom:EBMHead
er/corecom:Target/corecom:ID/text()) or
($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/corecom:EBMHeader/co
recom:Target/corecom:ID/text()='')) and
aia:getSystemType($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/cor
ecom:EBMHeader/corecom:Sender/corecom:ID)!='Oracle Clinical'))</expression>
 </condition>
 <action>
 <assign>
 <copy target="$out.UpdateClinicalStudy"
expression="$in.UpdateClinicalStudyEBM"/>
 </assign>
 <invoke reference="UpdateClinicalStudyOCHealthSciencesProvABCS"
operation="UpdateClinicalStudy">
 </invoke>
 </action>
 </case>
```

New Code:

```
<case name="Case_1_HealthSciencesClinicalStudyEBS.UpdateClinicalStudy"
executionType="direct">
 <condition language="xpath"
```

```
xmlns:ebo="http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/ClinicalStudy
/V1"
xmlns:corecom="http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2"
xmlns:aia="http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xp
ath.AIAFunctions">

<expression>((($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/coreco
m:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='P
RODUCTION' or
not($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/corecom:EBMHeader
/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and
(((not($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/corecom:EBMHea
der/corecom:Target/corecom:ID/text()) or
($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/corecom:EBMHeader/co
recom:Target/corecom:ID/text()='')) and
aia:getSystemType($in.UpdateClinicalStudyEBM/ebo:UpdateClinicalStudyEBM/cor
ecom:EBMHeader/corecom:Sender/corecom:ID)!='Oracle
Clinical'))))</expression>
 </condition>
 <action>
<transform>
<part name="$out.UpdateClinicalStudy" function="xslt(xsl/Xform_
UpdateClinicalStudyEBM_to_UpdateClinicalStudy.xsl,
$in.UpdateClinicalStudyEBM)"/>
</transform>
 <assign/>
 <invoke reference="UpdateClinicalStudyOCHealthSciencesProvABCS"
operation="UpdateClinicalStudy">
</invoke>
 </action>
 </case>
```

   b. Ensure the name selected for the XSL in the step 8.a matches with the transform clause in the new code.

10. You can deploy the changes to EBS to your environment using an EBS deployment plan. Create a new deployment plan StudySubjectVisitSyncSCandOCEBSDP.xml at <AIA_HOME>/pips/StudySubjectVisitSyncSCandOC/DeploymentPlans and enter content similar to the following example:

```
<DeploymentPlan component="StudySubjectVisitSyncSCandOC" version="3.0">
<PreInstallScript/>
<Configurations>
<EndpointConfigurator target-server="pips.StudySubjectVisitSyncSCandOC"
dir="${AIA_HOME}"> </EndpointConfigurator>
</Configurations>
<Deployments>
<Composite compositeName="HealthSciencesClinicalStudyEBS" compositedir="${AIA_
HOME}/pips/StudySubjectVisitSyncSCandOC/EBS/ClinicalStudy/HealthSciencesClinica
lStudyEBS" revision="1.0" wlserver="pips.StudySubjectVisitSyncSCandOC"
action="deploy" overwrite="true" />
</Deployments>
<PostInstallScript>
</PostInstallScript>
</DeploymentPlan>
```

   a. Deploy the modified EBS composite using the ant script.

   b. Navigate to `$AIA_INSTANCE_HOME}/bin/` and execute the following command:

      For Linux: `source aiaenv.sh`

      For Windows: `aiaenv.bat`

c. Execute the following deployment command:

The following is a sample of the `ant` script.

For Linux:
```
ant -f <AIA_
HOME>/Infrastructure/Install/AID/AIAInstallDriver.xml
-DDeploymentPlan=<AIA_
HOME>/pips/StudySubjectVisitSyncSCandOC/DeploymentPlans/StudySubjec
tVisitSyncSCandOCEBSDP.xml -DPropertiesFile=<AIA_HOME>/aia_
instances/<aia_instance>/config/AIAInstallProperties.xml -l <AIA_
HOME>/pips/StudySubjectVisitSyncSCandOC/DeploymentPlans/StudySubjec
tVisitSyncSCandOCEBS.log
```

For Windows:
```
ant -f <AIA_
HOME>\Infrastructure\Install\AID\AIAInstallDriver.xml
-DDeploymentPlan=<AIA_
HOME>\pips\StudySubjectVisitSyncSCandOC\DeploymentPlans\StudySubjec
tVisitSyncSCandOCEBSDP.xml -DPropertiesFile=<AIA_HOME>\aia_
instances\<aia_instance>\config\AIAInstallProperties.xml -l <AIA_
HOME>\pips\StudySubjectVisitSyncSCandOC\DeploymentPlans\StudySubjec
tVisitSyncSCandOCEBS.log
```

d. Restart the SOA Server.

e. Verify the EBS routing by testing the Site (SC to OC) flow for both OC instances.

11. Configure additional AQConsumer.

For the Subject Flow (OC to SC) to work, there should be an AQConsumer instance per OC instance listening to the message queue. To configure an additional instance of AQ Consumer, create a copy of the folder <AIA_HOME>/services/core/OracleClinical/AdapterServices/UpdateClinicalStudyOCAQConsumer in the same location such as, UpdateClinicalStudyOC2AQConsumer. Make changes to the following files as indicated:

a. Update composite.xml:

Table 6–2 Updating composite.xml

Old Code	New Code
`composite name="UpdateClinicalStudyOCAQConsumer"`	`composite name="UpdateClinicalStudyOC2AQConsumer"`
`service name="UpdateClinicalStudyOCAQConsumer"`	`service name="UpdateClinicalStudyOC2AQConsumer"`
`<svcdoc:ConnectionFactory>eis/AQ/UpdateClinicalStudyOCDS</svcdoc:ConnectionFactory>`	`<svcdoc:ConnectionFactory>eis/AQ/UpdateClinicalStudyOC2</svcdoc:ConnectionFactory>`
`<source.uri>UpdateClinicalStudyOCAQConsumer</source.uri>`	`<source.uri>UpdateClinicalStudyOC2AQConsumer</source.uri>`

b. Update ClinicalStudyOCAQConsumer_aq.jca:

Table 6–3 Updating ClinicalStudyOCAQConsumer_aq.jca

Old Code	New Code
`adapter-config name="UpdateClinicalStudyOCAQConsumer"`	`adapter-config name="UpdateClinicalStudyOC2AQConsumer"`

Table 6–3 (Cont.) Updating ClinicalStudyOCAQConsumer_aq.jca

Old Code	New Code
`connection-factory location="eis/AQ/UpdateClinicalStudyOCDS"`	`connection-factory location="eis/AQ/UpdateClinicalStudyOC2"`
`UIConnectionName="OCDB_sun6x13"`	`UIConnectionName="OCDB_<OC2DB_Name>"`

In UIConnectionName, replace <OC2DB_Name> with the actual value of OC2DB SID. Ensure that the connection-factory value matches with UIConnectionName.

c. Update ClinicalStudyOCAQConsumer.wsdl:

Table 6–4 Updating ClinicalStudyOCAQConsumer.wsdl:

Old Code	New Code
`wsdl:definitions name="UpdateClinicalStudyOCAQConsumer"`	`wsdl:definitions name="UpdateClinicalStudyOC2AQConsumer"`

12. Deploy the new AQConsumer. Create a new deployment plan StudySubjectVisitSyncSCandOC2AQDP.xml at <AIA_HOME>/pips/StudySubjectVisitSyncSCandOC/DeploymentPlans and enter content similar to the following example:

```
<DeploymentPlan component="StudySubjectVisitSyncSCandOC" version="3.0">
<PreInstallScript/> <Configurations> <Datasource
name="UpdateClinicalStudyOC2DS" jndiLocation="jdbc/UpdateClinicalStudyOC2"
action="create" database="participatingapplications.oc2.db" xa-enabled="true"
wlserver="pips.StudySubjectVisitSyncSCandOC"/><ManagedServer
wlserver="pips.StudySubjectVisitSyncSCandOC" action="shutdown"
failonerror="true"/> <AqAdapter
connection-instance-jndi="eis/AQ/UpdateClinicalStudyOC2"
datasource-jndi="jdbc/UpdateClinicalStudyOC2" xa-enabled="true" action="create"
wlserver="pips.StudySubjectVisitSyncSCandOC" /><ManagedServer
wlserver="pips.StudySubjectVisitSyncSCandOC" action="start"
failonerror="true"/> </Configurations><Deployments><Composite
compositeName="UpdateClinicalStudyOC2AQConsumer" compositedir="${AIA_
HOME}/services/core/OracleClinical/AdapterServices/UpdateClinicalStudyOC2AQCons
umer" revision="1.0" wlserver="pips.StudySubjectVisitSyncSCandOC"
action="deploy"
/></Deployments><PostInstallScript></PostInstallScript></DeploymentPlan>
```

a. Ensure that the AqAdapter connection-instance-jndi is exactly the same as the connection factory jndi mentioned in composite.xml and UpdateClinicalStudyOCAQConsumer_aq.jca.

b. Ensure that the datasource jndi mentioned in AQAdapter tag matches with the one in DataSource tag and is unique across the system.

c. Ensure that the DataSource name is unique across the WebLogic data sources.

d. Ensure that the value of the attribute compositedir is the location of the new folder that you have created and modified the files in that.

e. Ensure to update the compositeName to keep it unique per OC AQConsumer instance deployed.

f. Source `aiaenv.sh`.

```
Source {$AIA_INSTANCE_HOME}/bin/aiaenv.sh
```

g. Run the deployment plan using the following ant script:

```
ant -f <AIA_HOME>/Infrastructure/Install/AID/AIAInstallDriver.xml
-DDeploymentPlan=<AIA_
HOME>/pips/StudySubjectVisitSyncSCandOC/DeploymentPlans/StudySubjectVisitSy
ncSCandOC2AQDP.xml -DPropertiesFile=<AIA_HOME>/aia_instances/<aia_
instance>/config/AIAInstallProperties.xml -l <AIA_
HOME>/pips/StudySubjectVisitSyncSCandOC/DeploymentPlans/StudySubjectVisitSy
ncSCandOC2.log
```

13. Login to OC2 database as rxc user and execute the following command:

```
EXECUTE DBMS_AQADM.START_QUEUE(queue_name => 'CLINICAL_STUDY_QUEUE',dequeue =>
true,enqueue => true);
commit;
```

14. Restart the SOA server.
15. Verify the AQ Consumer deployment by testing the Subject (OC to SC) flow from the new OC instance.

7

Using the Resubmission Utility

This chapter discusses:

- Section 7.1, "Overview"
- Section 7.2, "Executing the Resubmission Utility for Study Site Flow"
- Section 7.3, "Executing the Resubmission Utility for Subject Activity Flow"

7.1 Overview

When a message cannot be delivered to a service or component in the flow of a global transaction, the message is rolled back to the appropriate source milestone. This source milestone corresponds to an Oracle Advanced Queue or JMS topic. The message will be persisted until it can be resubmitted for delivery to the service or component. At the same time, a fault is raised by the Error Handling framework and, if enabled, error notifications and Oracle BPM Worklist tasks regarding the fault are created to alert administrators.

Once the fault is notified, the most natural course of action is for the administrator to bring up the failed service or component. Once the service or component is back up and running, the administrator can use the *Message Resubmission Utility* to recover the faulted message from the source milestone.

The Message Resubmission Utility changes the state of the faulted message to the Ready state, enabling it to be picked up by the consumer process.

7.2 Executing the Resubmission Utility for Study Site Flow

To execute the resubmission utility for Study Site Flow (Siebel Clinical to Oracle Clinical):

1. Navigate to the following location:

   `<AIA_HOME>/util/AIAMessageResubmissionUtil`

2. Open the **ResubmissionParams.properties** file.
3. Enter the values for host name, admin.port, soa.port, user name, password.
4. Set the value of jms.aq to `false`.
5. Enter the value for jms.moduleName as `AIAJDBCJMSModule`.
6. Enter the value for jms.resourceCFJndi as `jms/aia/AIA_SiebelClinical_ClinicalStudyJMSQueueCF`.

7. Enter the value for jms.errorResourceCFJndi as `jms/aia/AIA_SiebelClinical_ClinicalStudyJMSQueueCF`.
8. Enter the value for resource type as `1`.
9. Enter the value for resourceName as `AIA_SiebelClinical_ClinicalStudyJMSQueue`.
10. Enter the value for messageID.

    You can get the value of messageID from the payload, the <corecom:SenderMessageI=D> field of EBM payload and append 'ID:<' before the ID and '>' at the end. For example, ID:<163824.1330947408709.0>.
11. Save the changes and close the ResubmissionParams.properties file.
12. Navigate to `<AIA_HOME>/aia_instances/<AIA_instance_name>/bin` and perform the following:
    - On Linux: `source aiaenv.sh`
    - On Windows: `aiaenv.bat`
13. Execute the following commands at the command prompt:
    - On Linux: `cd $AIA_HOME/util/AIAMessageResubmissionUtil`
    - On Windows: `cd %AIA_HOME%\util\AIAMessageResubmissionUtil`
14. Run the following command from the command prompt:

    ```
    ant -f MessageResubmit.xml -l MessageResubmit.log
    ```

    The following is a sample ResubmissionParams.properties for site flow:

    ```
    # Password parameters will be fetched from the command prompt, based on
    security guidelines.
    jms.app.host name=myhost.domain.com
    jms.app.admin.port=7072
    jms.app.soa.port=8072
    jms.app.userName=weblogic
    jms.app.password=<password for weblogic>
    jms.aq=false
    jms.moduleName=AIAJDBCJMSModule
    #Resource Connection Factory jndi name
    jms.resourceCFJndi=jms/aia/AIA_SiebelClinical_ClinicalStudyJMSQueueCF
    #Error Resource Connection Factory jndi name
    jms.errorResourceCFJndi=jms/aia/AIA_SiebelClinical_ClinicalStudyJMSQueueCF
    #QUEUE/TOPIC/RESEQUENCER - 1/2/3
    resourceType=1
    #queueName/topicName/routingServiceName
    resourceName=AIA_SiebelClinical_ClinicalStudyJMSQueue
    #messageID/groupID
    messageID=ID:<163824.1330947408709.0>
    ```

7.3 Executing the Resubmission Utility for Subject Activity Flow

To execute the resubmission utility for Subject activity flow (Oracle Clinical to Siebel Clinical):

1. Navigate to the following location:

    ```
    <AIA_HOME>/util/AIAMessageResubmissionUtil
    ```
2. Open the ResubmissionParams.properties file.

3. Enter the values for host name, admin.port, soa.port, user name, password.
4. Set the value of jms.aq to `true`.
5. Append `#` at the start of the jms.moduleName entry to mark it as a comment.
6. Do not enter values for jms.resourceCFJndi and jms.errorResourceCFJndi. Leave their values blank.
7. Enter the value for resource type as `1`.
8. Enter the value for resourceName as `CLINICAL_STUDY_QUEUE`.
9. Enter the value for messageID. For example, DFF91B414F3D602AE0440021287E64CE

   You can get the value of messageID from the payload, the <corecom:SenderMessageID> field of EBM payload.
10. Enter the value of aq.resourceTableName as `CLIN_STUDY_QUEUE_TBL`.
11. Enter the value of aq.db.jdbcURL as `jdbc:oracle:thin@<hostname:port:instance_name>`.
12. Enter the value of aq.db.userName as `rxc`.
13. Enter the rxc schema password for   aq.db.password.
14. Save the changes and close the ResubmissionParams.properties file.
15. Navigate to `$AIA_INSTANCE/bin/` and execute the following command:

    For Linux: `source aiaenv.sh`

    For Windows: `aiaenv.bat`
16. Execute the following deployment command:

    For Linux: `cd $AIA_HOME/util/AIAMessageResubmissionUtil ant -f MessageResubmit.xml -l MessageResubmit.log`

    For Windows: `cd %AIA_HOME%\util\AIAMessageResubmissionUtil ant -f MessageResubmit.xml -l MessageResubmit.log`

```
# Password parameters will be fetched from the command prompt, based on
security guidelines.
jms.app.hostName=myhost.domain.com
jms.app.admin.port=7072
jms.app.soa.port=8072
jms.app.userName=weblogic
jms.app.password=<password for weblogic>
jms.aq=true
#jms.moduleName=AIAJDBCJMSModule
#Resource Connection Factory jndi name
jms.resourceCFJndi=
#Error Resource Connection Factory jndi name
jms.errorResourceCFJndi=
#QUEUE/TOPIC/RESEQUENCER - 1/2/3
resourceType=1
#queueName/topicName/routingServiceName
resourceName=CLINICAL_STUDY_QUEUE
#messageID/groupID
messageID=BA2BB8280EED5340E0436E6A950A9DC1
#queueTableName/topicTableName
aq.resourceTableName=CLIN_STUDY_QUEUE_TBL
aq.db.driverName=oracle.jdbc.driver.OracleDriver
aq.db.jdbcURL=myhost.domain.com:1521:ocdb
aq.db.userName=rxc
aq.db.password=<password>
```

Index