Siebel Field Service Integration to Oracle Real-Time Scheduler Implementation Guide

Siebel Innovation Pack 2013

Version 8.1/8.2

September 2013

ORACLE

Contents

1 What's New in This Release

What's New in Siebel Field Service Integration to Oracle Real-Time Scheduler Implementation Guide, Version 8.1/8.2

No new features have been added to this guide for this release. This guide has been updated to reflect only product name changes.

What's New in Siebel Field Service Integration to Oracle Real-Time Scheduler Implementation Guide, Version 8.1, Rev. A

Table 1 lists changes described in this version of the documentation to support release 8.1 of the software.

Table 1. New Product Features in Siebel Field Service Integration to Oracle Real-Time Scheduler Implementation Guide, Version 8.1, Rev. A

Topic	Description
"Business Process Task Flow" on page 10	Updated topic. Includes support for Siebel Mobile connected applications.
"Design Assumptions for Mastering Data" on page 14	Updated topic. Includes support for Siebel Mobile connected applications.
"Mapping a Shift Status From Siebel CRM to Oracle Real-Time Scheduler" on page 108	Updated topic. Includes support for Siebel Mobile connected applications.
"Mapping a Break Status from Siebel CRM to Oracle Real-Time Scheduler" on page 109	Updated topic. Includes support for Siebel Mobile connected applications.

What's New in Siebel Field Service Integration to Oracle Real-Time Scheduler Implementation Guide, Version 8.2

Table 2 lists changes described in this version of the documentation to support release 8.2 of the software.

Table 2. New Product Features in Siebel Field Service Integration to Oracle Real-Time Scheduler Implementation Guide, Version 8.2

Topic	Description
"Adding User Credentials to the Credential Store" on page 96	New topic. You must configure this integration so that it can authenticate with Siebel Inbound Web Services.
"Ant Commands You Can Use to Compile and Deploy Composites" on page 101	New topic. To compile and deploy a composite, you can use different ant commands.
Customizing the Appointment Booking Flow to Support Error Messages in Multiple Languages	New topic. You can configure the connector to display error messages in multiple languages.
Chapter 10, "Extending the Connector"	New chapter. This chapter describes how to extend the connector.

2 Overview of Siebel Field Service Integration to Oracle Real-Time Scheduler

This chapter describes the direct process integration for Oracle's Siebel Field Service Integration to Oracle Real-Time Scheduler. It includes the following topics:

Direct Process Integration Overview

The direct process integration for Siebel Field Service Integration to Oracle Real-Time Scheduler helps Siebel Field Service customers achieve their service goals by allowing them to use important Oracle Real-Time Scheduler features.

NOTE: The procedures in this guide assume that you do not use left-hand navigation. However, you can set up left-hand navigation. For more information about left-hand navigation and about implementing it, see *Siebel Fundamentals for Siebel Open UI*.

Oracle Real-Time Scheduler is a software package that enables real time scheduling. It includes scheduling features, such as cost optimized appointments, street level routing with integrated mapping, and continuous optimization until the point of dispatch. These features provide benefits, such as labor and vehicle cost savings, Service Level Agreement (SLA) adherence, and increased customer satisfaction.

Integrating Siebel Field Service with Oracle Real-Time Scheduler provides important appointment booking and scheduling capabilities to Siebel Field Service users, and reduces implementation effort and risk. Siebel field Service Integration to Oracle Real-Time Scheduler supports the following functionality:

- Book, rebook, and cancel appointments.
- Book appointments according to customer SLA.
- Assign technicians according to the required job skill.
- Allow the technician to receive and send updates.
- Provide real-time synchronization of employee data.
- Retrieve geocodes for addresses.
- Perform a phased migration to Oracle Real-Time Scheduler if you currently use Siebel Scheduler.

Business Process Task Flow

Figure 1 illustrates a functional overview of the scheduling and dispatch task flow.

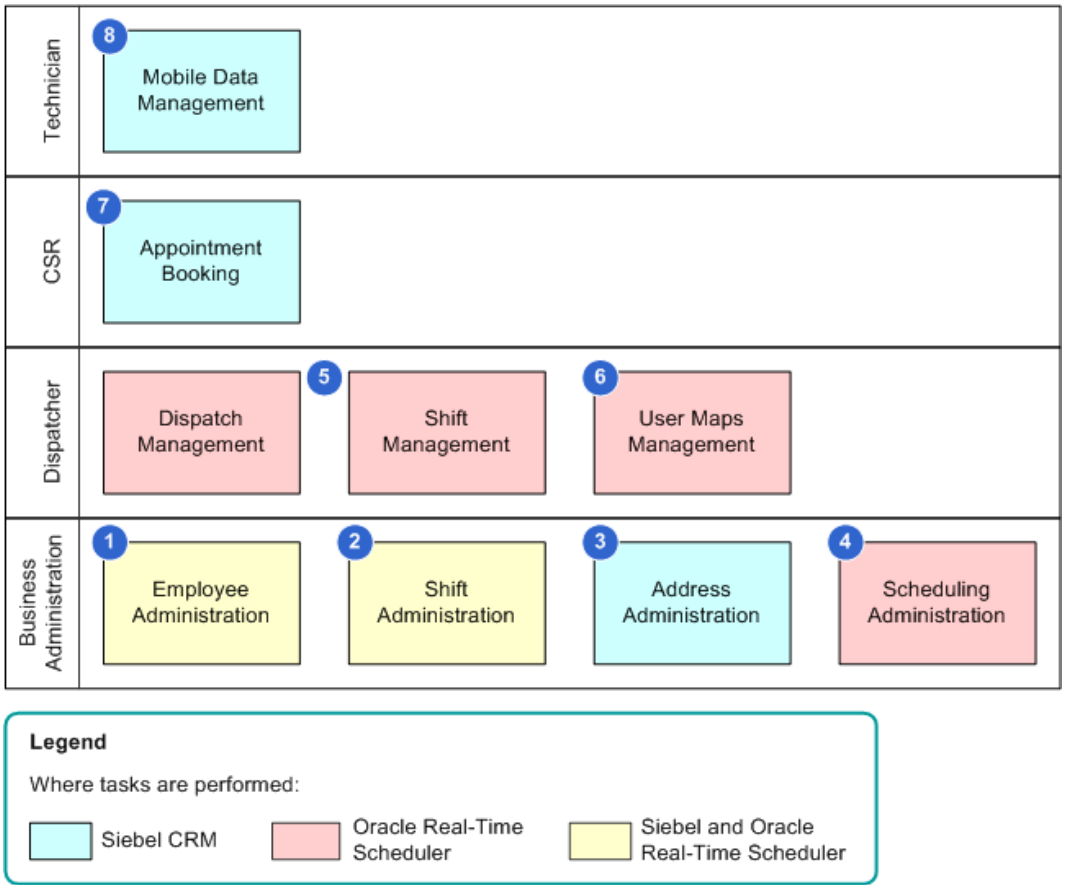

Figure 1. Functional Overview of the Scheduling and Dispatch Task Flow

Explanation of Callouts

The scheduling and dispatch task flow includes the following tasks:

1. **Employee administration.** Allows you to create and update employee data. You use the Siebel client to create and maintain the employee data that Field Service requires. Oracle Fusion Middleware synchronizes this information with Oracle Real-Time Scheduler. You then use the Oracle Real-Time Scheduler client to enter data that is specific to Oracle Real-Time Scheduler, such as employee cost profile mapping.
2. **Shift administration.** A subprocess that maintains schedules and exceptions. You must do the following:
   - Manually create the schedule record in Siebel CRM.
   - Associate the Schedule with Employees or Service Region.

3 **Address administration.** A subprocess that manages address data, including updating the latitude and longitude for each address that Oracle Real-Time Scheduler requires. This work includes retrieving geocodes for multiple existing addresses. It also includes creating new addresses and retrieving geocodes for these new addresses.

4 **Scheduling administration.** A subprocess that includes administration tasks that are specific to the Oracle Real-Time Scheduler scheduling module. You use the Oracle Real-Time Scheduler client to perform these tasks. Example tasks include managing slot groups, cost profiles, and so forth.

5 **Dispatch and shift management.** Includes tasks that are related to viewing and manually managing shifts and associated entities. You use the Oracle Real-Time Planner client to perform these tasks.

6 **User map management.** Includes tasks that are associated with using a map to view and manage the location of a technician.

7 **Appointment booking.** Includes tasks that are related to booking and canceling an appointment. You use the Siebel CRM client to perform these tasks.

8 **Mobile data management.** A subprocess that handles data synchronization between the Siebel wireless application or Siebel Mobile connected application that the technician uses, the Siebel Server, and Oracle Real-Time Scheduler.

Application Process Flows That This Integration Supports

Figure 2 illustrates the application process flows that this integration supports.

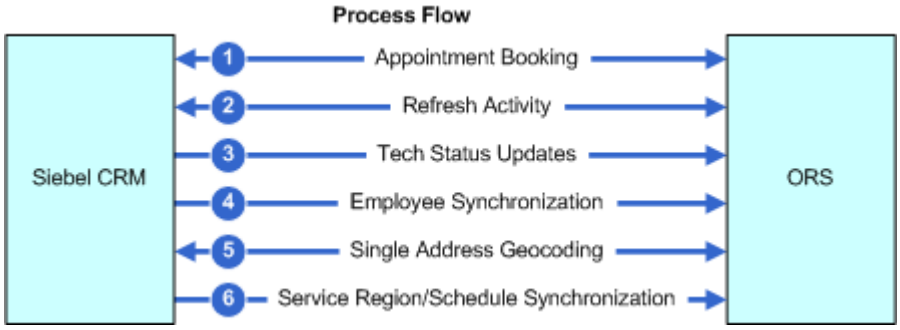

Figure 2. Application Process Flows That This Integration Supports

Explanation of Callouts

This integration supports the following application process flows:

1 **Appointment booking.** Includes the following features:

- Book an appointment
- Confirm an appointment
- Cancel an appointment
- Schedule a contract

- Rescheduling a contract
- Book an appointment that occurs beyond the scheduling horizon

2 **Refresh activity.** Gets the following items from Oracle Real-Time Scheduler:

- Activity status
- Activity owner
- Planned start
- Planned end

3 **Tech status updates.** Handles status updates from the technician for the following items:

- Activity
- Shift
- Breaks
- Periods of unavailability (POUs). For example, if a team meeting is scheduled for a technician, then Oracle Real-Time Scheduler creates a POU to represent the team meeting.

4 **Employee synchronization.** Handles real-time synchronization of administrative data. This synchronization includes the transfer of employee records along with employee addresses, and so forth.

5 **Single address geocoding.** Uses the Oracle Real-Time Scheduler Geocode Server to retrieve a geocode for a single address from the Siebel database in real time. If the Geocode Server returns multiple geocodes, then Siebel CRM displays these multiple geocodes. *Geocoding* is the process of finding latitude and longitude coordinates from geographic data, such as street addresses or postal codes.

6 **Schedule Synchronization.** Handles real-time synchronization of administrative data. This synchronization includes the transfer of schedule records, schedule hours, and schedule exceptions.

Data Flows That This Integration Supports

Figure 3 illustrates data flows that this integration supports.

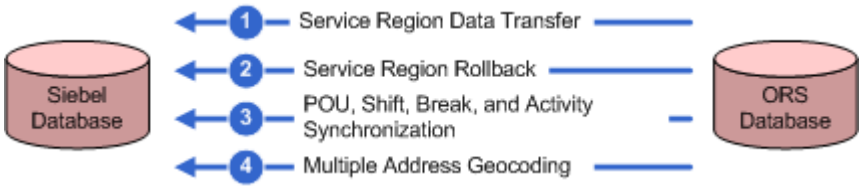

Figure 3. Data Flows That This Integration Supports

Explanation of Callouts

This integration supports the following data flows:

1. **Service region data transfer.** Transfers employees and activities associated with a service region from Siebel CRM to Oracle Real-Time Scheduler (ORS). If you currently use Siebel Scheduler, then you can use Service Region Data Transfer flow to do the following:
   - Perform a phased migration of service regions from the Siebel Scheduler to Oracle Real-Time Scheduler.
   - Migrate new service regions from Siebel CRM to Oracle Real-Time Scheduler.
2. **Service region rollback.** Inactivates all crews and cancels activities for the chosen service region in Oracle Real-Time Scheduler. Siebel Scheduler then schedules activities for this service region.

   Siebel Scheduler refers to the existing Scheduling module in Siebel CRM that uses the iLog engine. For more information, see *Siebel Field Service Guide*.
3. **POU, shift, break, and activity synchronization.** Synchronizes activity, shift, break, and POU data from Oracle Real-Time Scheduler to Siebel CRM.
4. **Multiple address geocoding.** Retrieves geocodes for multiple Siebel addresses and writes geocodes directly back to the Siebel database.

In this integration, a service region is a defined grouping of service engineers and the activities that they perform. For more information, see *Siebel Field Service Guide*.

A *service region data transfer* is a process that transfers employee and activity data for a service region from Siebel CRM to Oracle Real-Time Scheduler. This capability provides a way for the Oracle Real-Time Scheduler engine to schedule the service region instead of the Siebel Scheduler.

Benefits of Using This Integration

The Oracle Fusion Middleware architecture supports the data flows described in Figure 2 on page 11 and Figure 3 on page 12, and integrates Siebel CRM and Oracle Real-Time Scheduler. This integration makes Oracle Real-Time Scheduler Web services available as Oracle Fusion Middleware composites. The Siebel application uses these composites to call Oracle Real-Time Scheduler. This integration provides the following benefits:

- Seamless integration between Siebel CRM and Oracle Real-Time Scheduler Scheduler
- Loose coupling between components
- High availability and scalability of the Siebel application
- High error resilience and advanced fault handling mechanisms
- Monitoring services

Design Assumptions for Mastering Data

Table 3 describes how data must be mastered.

Table 3. Design Assumptions for Mastering Data

Entity	Operation	Master	Synchronization Direction	Descriptions
Activity	Create	Siebel CRM	One way	Not applicable
	Update	Siebel CRM and Oracle Real-Time Scheduler	Two way	Not applicable
	Delete	Not applicable	Not applicable	Not allowed
Employee	Create	Siebel CRM	One way	Not applicable
	Update			Not applicable
	Delete	Not applicable	Not applicable	Not allowed
Employee Addresses	Link or unlink	Siebel CRM	One way	Not applicable
Employee Skills	Link or unlink			Not applicable
Addresses	Create			Not applicable
	Update			Not applicable
	Delete	Not applicable	Not applicable	Not applicable
POU	Create	Siebel CRM	One way	Not applicable
	Update	Siebel CRM and Oracle Real-Time Scheduler	Two way	
	Delete	Not applicable	Not applicable	
Break	Create	Siebel CRM	One way	This integration sends the break actual start, actual end date, and break status from a Siebel wireless application or Siebel Mobile connected application to Oracle Real-Time Scheduler.
	Update	Siebel CRM and Oracle Real-Time Scheduler	Two way	Not applicable
	Delete	Not applicable	Not applicable	

Table 3. Design Assumptions for Mastering Data

Entity	Operation	Master	Synchronization Direction	Descriptions
Shift Template	Create	Siebel CRM	One way	Not applicable
	Update			
	Delete	Not applicable		Not applicable
Leaves	Create	Siebel CRM	Not applicable	Not applicable
	Update			
	Delete			
Slot Group	Create	Oracle Real-Time Scheduler	Replication	This integration does not synchronize slot groups to Siebel CRM. You must manually recreate the slot group in Siebel CRM as a LOV.
	Update			
	Delete			
Slots	Create		No synchronization occurs.	Siebel CRM does not require this data.
	Update			
	Delete			
Cost Profile	Create			
	Update			
	Delete			
Service Region	Create	Siebel CRM	One Way	Oracle Real-Time Scheduler does not require this data.
	Update			
	Delete			

Documentation Sources for Configuring Oracle Real-Time Scheduler

This topic describes documentation sources that provide information for Siebel Field Service Integration to Oracle Real-Time Scheduler.

Oracle Real-Time Scheduler Documentation

For more information about tasks you perform in Oracle Real-Time Scheduler, see the documentation for Oracle Real-Time Scheduler. You can obtain this documentation through Oracle Software Delivery Cloud, under the Oracle Utilities Applications product pack. Some of the relevant documentation titles for Oracle Real-Time Scheduler include the following:

- *Oracle Real-Time Scheduler V2.1.0 Release Notes*
- *Oracle Real-Time Scheduler V2.1.0 Quick Install Guide*
- *Oracle Real-Time Scheduler V2.1.0 Install Documentation*
- *Oracle Real-Time Scheduler V2.1.0 User Documentation*
- *Oracle Real-Time Scheduler V2.1.0 Supplemental Documentation*

Also see the readme files and release notes for Oracle Real-Time Scheduler.

Oracle Fusion Middleware and Oracle SOA Suite Documentation

For more information about Oracle Fusion Middleware and Oracle SOA Suite, see the following documentation on Oracle Technology Network:

- *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite*
- *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*
- *Oracle Fusion Middleware Introduction to Oracle WebLogic Server*
- *Oracle Fusion Middleware Installation Guide for Oracle WebLogic Server*
- *Enterprise Deployment Guide for Oracle SOA Suite*

More Resources

The following resources are available on My Oracle Support:

- Oracle Application Integration Architecture: Product-to-Guide Index
- Known Issues and Workarounds
- Release notes
- Documentation updates

3 Appointment Booking Integration

This chapter describes the appointment booking process integration. It includes the following topics:

Overview of Appointment Booking Integration

The appointment booking process includes the following flows:

- **Get and Confirm Appointment.** Makes an appointment request, displays the available slots, and then confirms one of the slots.
- **Reschedule an Appointment.** Rebooks a previously confirmed appointment.
- **Cancel an Appointment.** Cancels a confirmed appointment.
- **Update an Appointment.** Updates activity details.
- **SLA Based Appointment (Contract Scheduling).** Inserts an appointment into an optimized schedule to meet the SLA Service Level Agreement (SLA) requirements of the customer.
- **Get Latest Appointment Details**. Gets the latest appointment details from Oracle Real-Time Scheduler.
- **Book Appointment for Child Activity.** Makes an appointment request for a bounded job.

A *composite* is a deployment unit of the Service-Oriented Architecture (SOA) composite architecture that Oracle SOA Suite 11g supports. A composite includes various components, such as Mediator, BPEL flow, Adapters, and so forth. Siebel Field Service Integration to Oracle Real-Time Scheduler creates a single SOA composite to perform tasks that are related to the Appointment Booking System (ABS). The entry point to this composite is an outbound Web service call from Siebel CRM.

The Web Service Definition Language (WSDL) file defines the operations and the expected message formats. The service that is available is the entry point that Siebel CRM uses to target the Siebel CRM Web service request. Separate operations are defined within the WSDL to book an appointment, cancel an appointment, and so forth.

A single Business Process Execution Language (BPEL) process handles requests that occur in this composite. The mediator sets the OPERATION_TYPE global variable in the BPEL process in one of the following ways:

- To Book_Appt if the Web service operation is Book_Appointment
- To Confirm_Appt if the Web service operation is Confirm_Appointment

- And so forth for other Web service operations

This integration handles these requests so that it can use a single Business Process Execution Language (BPEL) process for these tasks and so that it can place a switch inside the BPEL that calls the appropriate Oracle Real-Time Scheduler Web service.

For more information, see Appendix A, "Integration Services."

Integration Flows That Book an Appointment

This topic describes the integration flows in the Appointment Booking System composite. It includes the following topics:

For more information, see "Hook Points That Book an Appointment" on page 83.

Integration Flow That Books or Reschedules an Appointment

Figure 4 illustrates the integration flow to book an appointment. The flow to reschedule an appointment is same as the flow to book an appointment. The is a synchronous request and reply flow. It uses Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP) for all interactions with Siebel CRM and Oracle Real-Time Scheduler. For more information, see "Mapping Definitions That Book, Confirm, Insert, or Reschedule an Appointment" on page 127.

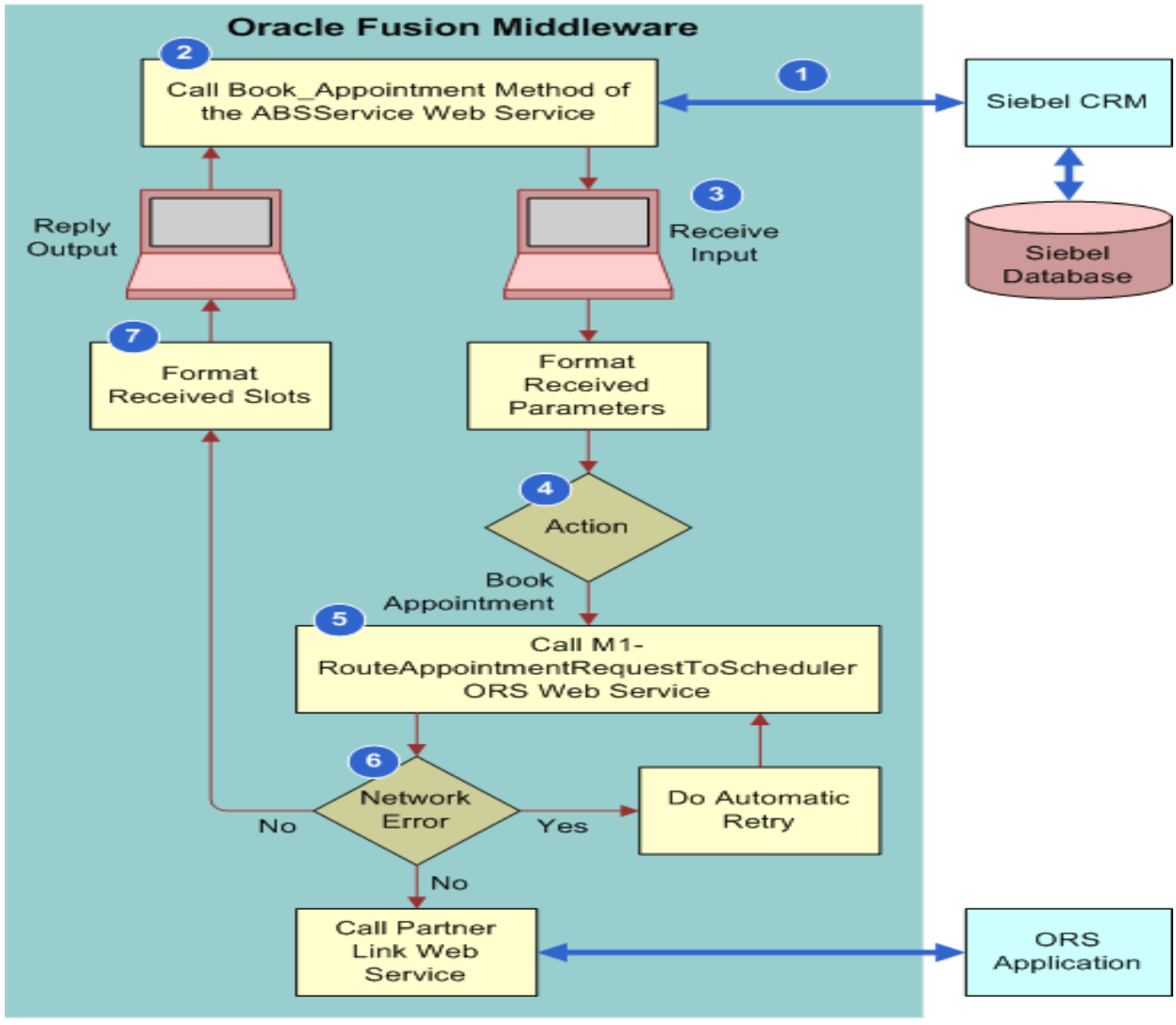

Figure 4. Integration Flow to Book or Reschedule an Appointment

Explanation of Callouts

The integration flow to book or reschedule an appointment includes the following steps:

1. Receives input from Siebel CRM.
2. Siebel CRM makes a Web service call to the Book_Appointment method of the ABSService Web service.
3. The mediator forwards the request to the BPEL flow.
4. The decision diamond represents a case statement. If the flow is a book appointment flow, then this integration performs the flow illustrated in Figure 4. For other flows, see the appropriate topic in this book.
5. To get slots from Oracle Real-Time Scheduler, the BPEL flow calls the M1-RouteAppointmentRequestToScheduler Web service method of Oracle Real-Time Scheduler.
6. If there is a network error, then this integration automatically attempts to communicate again with the Web service. A network error indicates that this integration cannot establish communication with the Oracle Real-Time Scheduler Web service.
7. The BPEL flow formats the slots that Oracle Real-Time Scheduler returns and sends them back to Siebel CRM.

For more information, see Appendix A, "Integration Services.".

How This Integration Locks Assignments and Schedules

This integration locks assignments and schedules in the following ways:

- If the Lock Assignment flag is set in the Web service input that the BPEL flow receives, then this integration sets the ROW_ID of the employee at the Allowed Crew Name field while sending the request to Oracle Real-Time Scheduler.
- If the Lock Schedule flag is set in the input, then this integration does not set the period of the request from the earliest start field and latest start field that the input specifies. Instead, it uses the planned start and planned end fields of the input. This integration uses a tolerance of plus or minus ten minutes for the duration of the time period.

Domain Value Map That Books or Reschedules an Appointment

A domain value map (DVM) is an SOA utility that facilitates the mapping of a list of values from one application to another application. In this integration, the Siebel application provides source values to Oracle Real-Time Scheduler. For example, assume the State value in a list of values includes the following values:

- CA in Siebel CRM
- California in Oracle Real-Time Scheduler

The DVM maps CA to California.

Table 4 describes the DVM that occurs in the book and reschedule appointment flow. This integration assumes that all DVMs are stored on the *Oracle Meta Data Services (MDS)*. The Oracle MDS is a shared area in Oracle Fusion Middleware. It can reference a database or files. To support access to the same DVMs and cross references for multiple composites, this integration assumes Oracle MDS references a database. For more information, see "Configuring Domain Value Maps" on page 105.

Table 4. DVM That Books or Reschedules an Appointment

DVM	Description
SBL_ORS_Parameter_Definition.dvm	This integration uses the following DVM parameters: ■ ORS-No of Slots ■ ORS-No of Slots-<SRV_REGION_NAME> If SRV_REGION_NAME is missing, then this integration uses only ORS-No of Slots. ■ ORS-Request Slot MaxCost ■ ORS-Request Slot MaxCost-<SRV_REGION_NAME> If SRV_REGION_NAME is missing, then this integration uses only ORS-Request Slot MaxCost. For more information, see "IMapping Run-Time Parameters" on page 115.
DVM-SBL_ORS_Endpoint_Definition.dvm	This integration uses the Appointment Booking DVM parameter.

Integration Flow That Confirms an Appointment

The integration flow to confirm an appointment is the same as the flow to book or reschedule an appointment illustrated in Figure 4 on page 19 except for the following differences:

- Siebel CRM makes a Web service call to the Confirm_Appointment method of the ABSService Web service.
- To confirm the chosen slot, the BPEL flow calls the M1-MaintainActivityByHost Web service method of Oracle Real-Time Scheduler.
- The BPEL flow sends the reply that it receives from Oracle Real-Time Scheduler to Siebel CRM. The reply from Oracle Real-Time Scheduler can be a success or failure.

For more information, see Appendix A, "Integration Services" and "Mapping Definitions That Manage an Appointment" on page 127.

DVM That Confirms an Appointment

Table 5 describes the DVM to confirm an appointment. For more information, see "Configuring Domain Value Maps" on page 105.

Table 5. DVM That Confirms an Appointment

DVM	Description
SBL_ORS_Parameter_Definition.dvm	This integration uses the following DVM parameters: ■ ORS-Confirm Slot MaxCost ■ ORS-Confirm Slot MaxCost-<SRV_REGION_NAME> If SRV_REGION_NAME is missing, then this integration uses only ORS- Confirm Slot MaxCost. ■ ORS-FindShift ■ ORS-FindShift-<SRV_REGION_NAME> If SRV_REGION_NAME is missing, then this integration uses only ORS-FindShift.
DVM-SBL_ORS_Endpoint_Definition.dvm	This integration uses the Activity Maintenance DVM parameter.

Integration Flow That Cancels an Appointment

The integration flow to cancel an appointment is the same as the flow to confirm an appointment described in "Integration Flow That Confirms an Appointment" on page 21 except for the following differences:

- Siebel CRM calls the Cancel_Appointment method.
- To cancel the appointment in Oracle Real-Time Scheduler, the BPEL flow calls the M1-ActivityStatusUpdatesFromHost Web service method of Oracle Real-Time Scheduler.

For more information, see Appendix A, "Integration Services" and "Mapping Definitions That Synchronize a Status" on page 129.

DVM That Cancels an Appointment

Table 6 describes the DVM to cancel an appointment. For more information, see "Configuring Domain Value Maps" on page 105.

Table 6. DVM That Cancels an Appointment

DVM	Description
SBL_ORS_Endpoint_Definition.dvm	This integration uses StatusUpdatesFromHost DVM parameter.

Integration Flow That Schedules an Appointment According to a Contract

The integration flow to schedule an appointment according to a contract is the same as the flow to confirm an appointment described in "Integration Flow That Confirms an Appointment" on page 21 except for the following differences:

- Siebel CRM calls the Upsert_Activity method.
- To insert the activity in Oracle Real-Time Scheduler, the BPEL flow calls the M1-MaintainActivityByHost Web service method of Oracle Real-Time Scheduler.

For more information, see Appendix A, "Integration Services.".

DVM That Schedules an Appointment According to a Contract

Table 7 describes the DVM to schedule an appointment according to a contract. For more information, see "Configuring Domain Value Maps" on page 105.

Table 7. DVM That Schedules an Appointment According to a Contract

DVM	Description
SBL_ORS_Endpoint_Definition.dvm	This integration uses Activity Maintenance DVM parameter.

Integration Flow That Updates and Inserts an Appointment

The integration flow to update and insert an appointment is the same as the flow to confirm an appointment described in "Integration Flow That Confirms an Appointment" on page 21 except for the following differences:

- Siebel CRM calls the Upsert_Activity method. *Upsert* means to perform an update operation and an insert operation.

- To insert the activity in Oracle Real-Time Scheduler, the BPEL flow calls the M1-MaintainActivityByHost Web service method.

DVM That Updates and Inserts an Appointment

The DVM to update and insert an appointment is the same as the DVM to schedule an appointment according to a contract. For more information, see "DVM That Schedules an Appointment According to a Contract" on page 23.

Output XSD That Updates and Inserts an Appointment

The output XML Schema Definition (XSD) to update and insert an appointment is the same as the output XSD it uses to confirm an appointment. For more information, see Appendix A, "Integration Services.".

Integration Flow That Retrieves Activity Details fromOracle Real-Time Scheduler

The details of an appointment continue to change in Oracle Real-Time Scheduler even after the customer service representative (CSR) confirms the appointment. Details include Crew assigned to the activity, planned start, and so forth. If the CSR requires the current details of the activity, then the CSR must click Refresh Activity in Siebel CRM.

The integration flow to retrieve activity details is the same as the flow to book or reschedule an appointment illustrated in Figure 4 on page 19 except for the following differences:

- Siebel CRM calls the Retrieve_Activity_State method of the ActivityStateRetrieval Web service.
- To get the latest details on the activity from Oracle Real-Time Scheduler, the BPEL flow calls the M1-ActivityStatusInquiry Web service method.
- The BPEL flow formats the reply that it receives from Oracle Real-Time Scheduler and then sends this reply to Siebel CRM.

This integration uses the ActivityLatestStateRetrieval_EBF project to get appointment details.

For more information, see the following topics:

- Appendix A, "Integration Services"
- "Hook Points That Update Employee Data" on page 87

DVM That Gets Appointment Details

Table 8 describes the DVM to get appointment details. This integration assumes that all DVMs are stored on the Oracle MDS. For more information, see "Configuring Domain Value Maps" on page 105.

Table 8. DVM That Gets Appointment Details

DVM	Description
SBL_ORS_Endpoint_Definition.dvm	This integration uses the ActivityStatusInquiry DVM parameter.
ORS_to_SBL_ActivityStatusMapping.dvm	Used to extract the Siebel CRM equivalent of an Oracle Real-Time Scheduler status. For more information, see "Configuring Domain Value Maps" on page 105.

Integration Flow That Books an Appointment for a Child Activity

Users must associate an already booked activity, which will be the parent activity for the current activity, and set the minimum offset duration (within which the child activity has to be scheduled from parent activity) and the maximum offset (within which the child activity has to be scheduled from parent activity) activity. The minimum offset and maximum offset are in minutes and considered from the Start time of the Parent activity.

When you click of Book Appointment, Siebel CRM makes a Web service call to the Upsert_Appointment method of the ABSService Web service. The mediator forwards the request to the BPEL flow to insert the activity in Oracle Real-Time Scheduler. The BPEL flow then calls the M1-MaintainActivityByHost Web service method from the Oracle Real-Time Scheduler. Instead of displaying the slots, Siebel CRM will display a confirmation message while booking an appointment for Child Activity.

Web Services That Book an Appointment

This topic describes Web services that book an appointment.

Outbound Web Service That Books an Appointment

Table 9 describes the outbound Web service for the Appointment Booking System.

Table 9. Outbound Web Service for the Appointment Booking System

Web Service Name	Proxy Business Service Name
ABSWebService	ABSWebServiceSoap

The Data Type property is Integration Object for all method arguments of the ABSWebService Web service that book an appointment.

Table 10 describes the arguments of the Book_Appointment Web service method.

Table 10. Arguments for the Book_Appointment Web Service Method

Name	Integration Object Name	Type
Book_Appointment_IN:in	Abssiebel	Input
Book_Appointment_OUT:out	BOOK_APPT_RESPONSE	Output

Table 11 describes the arguments of the Cancel_Appointment Web service method.

Table 11. Arguments of the Cancel_Appointment Web Service Method

Name	Integration Object Name	Type
Cancel_Appointment_IN:in	Abssiebel	Input
Cancel_Appointment_OUT:out	CANCEL_APPT_RESPONSE	Output

Table 12 describes the arguments of the Confirm_Appointment Web service method.

Table 12. Arguments of the Confirm_Appointment Web Service Method

Name	Integration Object Name	Type
Confirm_Appointment_IN:in	Abssiebel	Input
Confirm_Appointment_OUT:out	CONFIRM_APPT_RESPONSE	Output

Table 13 describes the arguments of the Contract_Scheduling Web service method.

Table 13. Arguments of the Contract_Scheduling Web Service Method

Name	Integration Object Name	Type
Contract_Scheduling_IN:in	Abssiebel	Input
Contract_Scheduling_OUT:out	CONFIRM_APPT_RESPONSE	Output

Table 14 describes the arguments of the Upsert_Activity Web service method.

Table 14. Arguments of the Upsert_Activity Web Service Method

Name	Integration Object Name	Type
Upsert_Activity_IN:in	Abssiebel	Input
Upsert_Activity_OUT:out	UPSERT_ACTIVITY_RESPONSE	Output

Integration Objects That Book an Appointment

Table 15 describes the integration objects that book an appointment.

Table 15. Integration Objects for the Appointment Booking System

Integration Object Name	Business Object	Type
Abssiebel	Action	Hierarchy
BOOK_APPT_RESPONSE	Not applicable	XML
CONFIRM_APPT_RESPONSE		
CANCEL_APPT_RESPONSE		
UPSERT_ACTIVITY_RESPONSE		

Table 16 describes the properties of the Abssiebel integration object.

Table 16. Properties of the Abssiebel Integration Object

Integration Component	Business Component	Parent Integration Component
Action	Action	Not applicable
Action_Service Address	CUT Address	Action
FS Activity Skill	FS Activity Skill	Action
FS Activity Skill Item	FS Activity Skill Item	FS Activity Skill Item

Table 17 describes the properties of the BOOK_APPT_RESPONSE integration object.

Table 17. Properties of the BOOK_APPT_RESPONSE Integration Object

Integration Component	Business Component	Parent Integration Component
CHS_REPLY	CHS_REPLY	BOOK_APPT_RESPONSE

Table 17. Properties of the BOOK_APPT_RESPONSE Integration Object

Integration Component	Business Component	Parent Integration Component
ERROR	ERROR	BOOK_APPT_RESPONSE
RESULT	RESULT	BOOK_APPT_RESPONSE

Table 18 describes the properties of the CANCEL_APPT_RESPONSE integration object.

Table 18. Properties of the CANCEL_APPT_RESPONSE Integration Object

Integration Component	Business Component	Parent Integration Component
CANCEL_APPT_RESPONSE	CANCEL_APPT_RESPONSE	Not applicable
ERROR	ERROR	CANCEL_APPT_RESPONSE
RESULT	RESULT	CANCEL_APPT_RESPONSE

Table 19 describes the properties of the CONFIRM_APPT_RESPONSE integration object.

Table 19. Properties of the CONFIRM_APPT_RESPONSE Integration Object

Integration Component	Business Component	Parent Integration Component
CONFIRM_APPT_RESPONSE	CONFIRM_APPT_RESPONSE	Not applicable
ERROR	ERROR	CONFIRM_APPT_RESPONSE
RESULT	RESULT	CONFIRM_APPT_RESPONSE

Table 20 describes the properties of the UPSERT_ACTIVITY_RESPONSE integration object.

Table 20. Properties of the UPSERT_ACTIVITY_RESPONSE Integration Object

Integration Component	Business Component	Parent Integration Component
UPSERT_ACTIVITY_RESPONSE	UPSERT_ACTIVITY_RESPONSE	Not applicable
ERROR	ERROR	UPSERT_ACTIVITY_RESPONSE
RESULT	RESULT	UPSERT_ACTIVITY_RESPONSE

4 How the Connector Synchronizes Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

This chapter describes how the connector synchronizes shift, break, period of unavailability (POU), or activity data from Oracle Real-Time Scheduler to Siebel CRM. It includes the following topics:

Overview of How the Connector Synchronizes Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

During a search to find the best solution to minimize costs, Oracle Real-Time Scheduler continually changes certain activity parameters. Oracle Real-Time Scheduler performs these changes on the following items:

- Status column
- Time fields
- Owner Id field

Siebel Field Service Integration to Oracle Real-Time Scheduler does the following:

- If the activity status is set to Dispatched or Inactive (Cancelled Activity by Scheduler) in Oracle Real-Time Scheduler, then this integration synchronizes activity details to Siebel CRM. When this integration synchronizes the Cancelled Activity By Scheduler, the Activity Status in Siebel CRM will be updated to Cancelled and the Status Reason from Oracle Real-Time Scheduler is mapped to the Error Description field in Siebel CRM to distinguish the Activity being Cancelled by Scheduler compared to the Activity being Cancelled by a customer service representative directly in Siebel CRM.
- If a shift is created, updated, or deleted in Oracle Real-Time Scheduler, then this integration sends shift data to Siebel CRM.
- If a break is dispatched in Oracle Real-Time Scheduler, then this integration sends break data to Siebel CRM.
- If a POU is dispatched in Oracle Real-Time Scheduler, then this integration sends POU data to Siebel CRM.

This integration performs this work without physical intervention from a CSR (customer service representative).

When an Activity is dispatched or a Shift, Break, or POU is created or updated, Oracle Real-time Scheduler invokes a Web Service to pass the relevant data to the integration component. The data is sent to Siebel CRM from the integration component by invoking Web Services.

Integration Flow That Synchronizes Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

Figure 5 on page 32 illustrates the integration flow to synchronize the status for a shift, break, POU, or activity from Oracle Real-Time Scheduler to Siebel CRM.

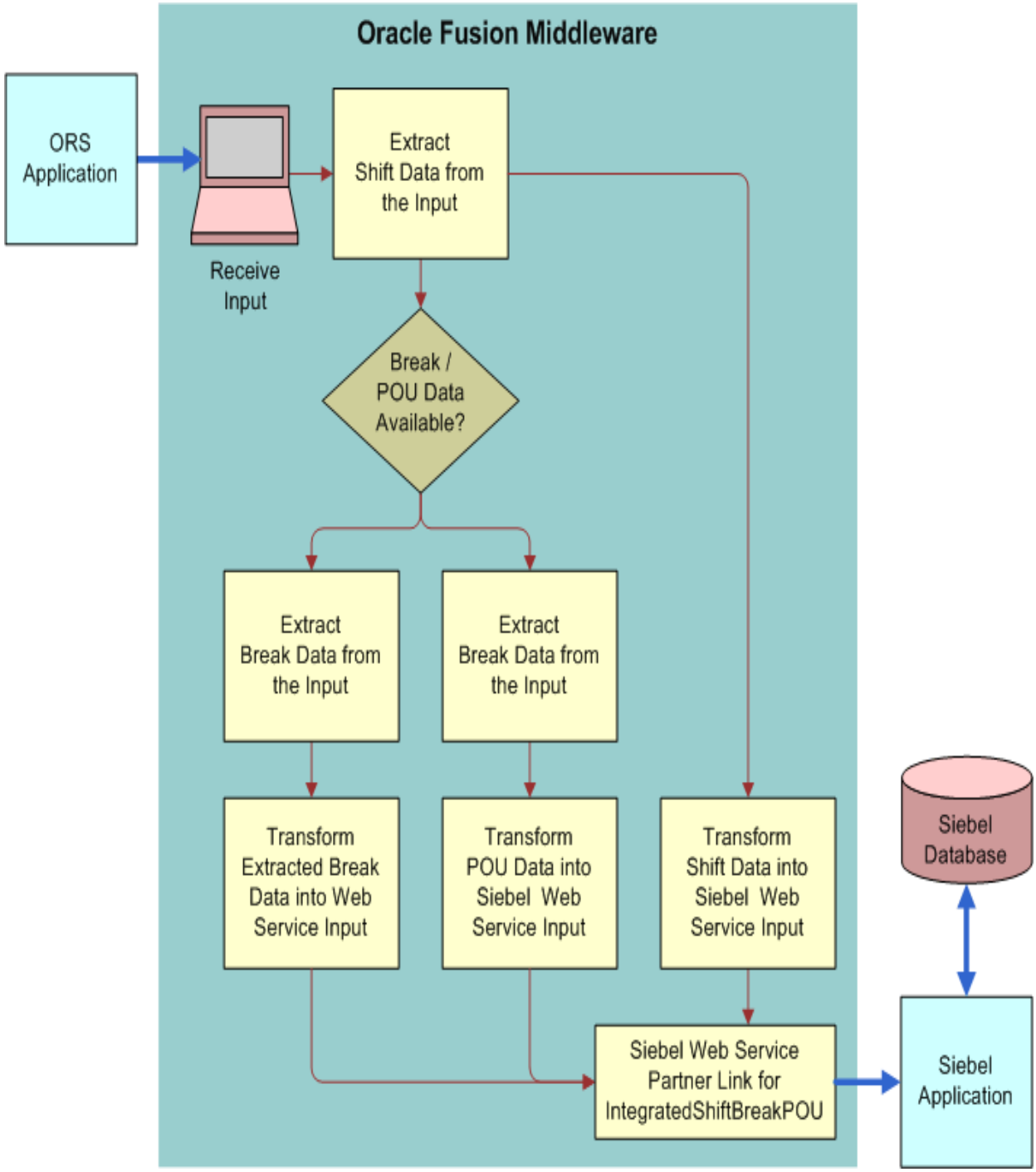

Figure 5. Integration Flow to Synchronize the Status for a Shift, Break, POU, or Activity from Oracle Real-Time Scheduler to Siebel CRM

Figure 6 illustrates the integration flow to synchronize the status for an activity from Oracle Real-Time Scheduler to Siebel CRM.

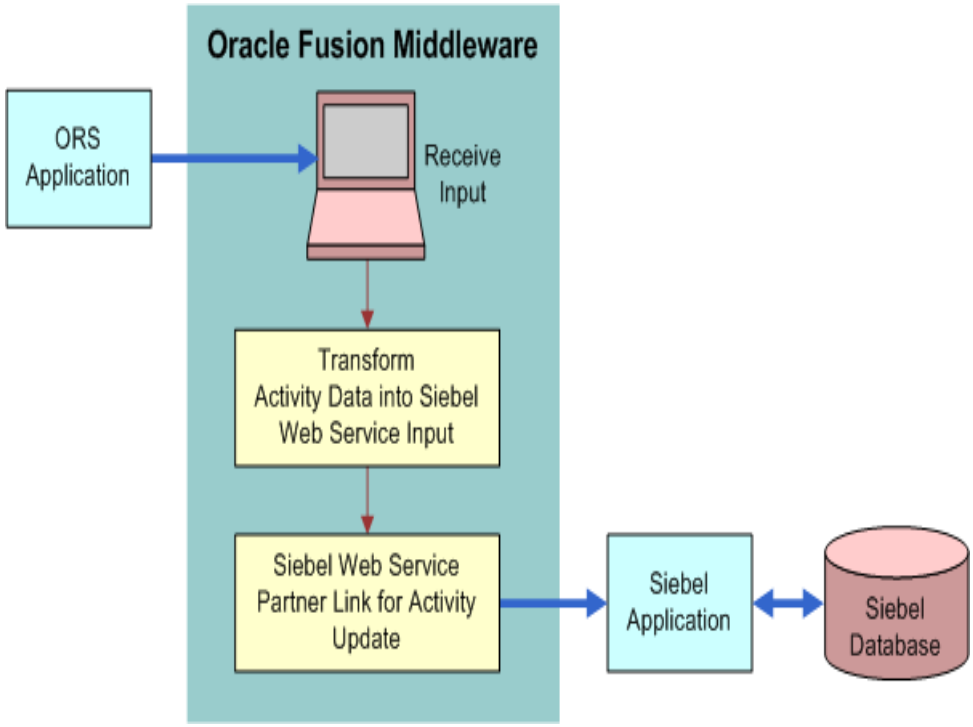

Figure 6. Integration Flow to Synchronize the Status for an Activity from Oracle Real-Time Scheduler to Siebel CRM

DVM That Synchronizes Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

Table 21 describes the domain value map (DVM) to synchronize statuses from Oracle Real-Time Scheduler with Siebel CRM. For more information, see "Mapping an Activity Status From Oracle Real-Time Scheduler to Siebel CRM" on page 107.

Table 21. DVM That Synchronizes Statuses from Oracle Real-Time Scheduler to Siebel CRM

DVM	Description
ORS_SBL_ActivityStatusMapping.dvm	Used to extract the Siebel Status equivalent to the Oracle Real-Time Scheduler status.

Web Services That Synchronize Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

This topic describes Siebel Web services that synchronize activity and related data from Oracle Real-Time Scheduler to Siebel CRM.

Inbound Web Service That Synchronizes Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

Table 22 describes the inbound Web service that synchronizes activity and related data from Oracle Real-Time Scheduler to Siebel CRM.

Table 22. Inbound Web Service That Synchronizes Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

Web Service Name	Proxy Business Service Name
IntegratedPOUShiftBreakORSWS	IntegratedPOUShiftBreakORSWS
ActivitySyncORSWS	ActivitySyncORSWS

Table 23 describes the inbound Web service ports that synchronize activity and related data from Oracle Real-Time Scheduler to Siebel CRM.

Table 23. Inbound Web Service Ports That Synchronize Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

Name	Type	Business Service or Business Process Name
IntegratedPOUShiftBreakORSWS	Business Service	IntegratedPOUShiftBreakORSWS
POUUpsert	POUUpsert	ORS POU Inbound Upsert Workflow

Table 24 describes the properties of the Insert, InsertOrUpdate, and Delete Web service methods.

Table 24. Properties of the Insert, InsertOrUpdate, and Delete Web Service Methods

Method Argument Name	Data Type	Integration Object Name	Type
SiebelMessage	Integration Object	Integrated Shift Break POU	Input and Output

Table 25 describes the properties of the Update Web service method.

Table 25. Properties of the Update Web Service Method

Method Argument Name	Data Type	Integration Object Name	Type
SiebelMessage	Integration Object	Abssiebel	Input and Output

Table 26 describes the inbound Web service ports that synchronize activity and related data from Oracle Real-Time Scheduler to Siebel CRM.

Table 26. Inbound Web Service Ports That Synchronize Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

Name	Type	Business Service
ActivitySyncORSWS	Business Service	ActivitySyncORSWS

Integration Objects That Synchronize Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

Table 27 describes the integration objects that synchronize activity and related data from Oracle Real-Time Scheduler to Siebel CRM.

Table 27. Integration Objects That Synchronize Activity and Related Data from Oracle Real-Time Scheduler to Siebel CRM

Integration Object Name	Business Object	Type
Integrated Shift Break POU	Integrated Shift Break POU	Siebel Business Object
STATUS_UPDATE_RESPONSE	Not applicable	XML

Table 28 describes the properties of the Integrated Shift Break POU integration object.

Table 28. Properties of the Integrated Shift Break POU Integration Object

Integration Component	Business Component	Parent Integration Component
Integrated Shift Break POU	Integrated Shift Break POU	Not applicable

Table 29 describes the properties of the STATUS_UPDATE_RESPONSE integration object.

Table 29. Properties of the STATUS_UPDATE_RESPONSE Integration Object

Integration Component	Business Component	Parent Integration Component
STATUS_UPDATE_RESPONSE	STATUS_UPDATE_RESPONSE	Not applicable
RESULT	RESULT	STATUS_UPDATE_RESPONSE
ERROR	ERROR	STATUS_UPDATE_RESPONSE

5 How the Connector Synchronizes Activity and Related Data from Siebel CRM to Oracle Real-Time Scheduler

This chapter describes how the connector synchronizes the status of a shift, break, period of unavailability (POU), or activity from Siebel CRM to Oracle Real-Time Scheduler. It includes the following topics:

Overview of How the Connector Synchronizes a Status from Siebel CRM to Oracle Real-Time Scheduler

The field technician can send the following updates to Oracle Real-Time Scheduler:

- Activity status updates, such as Acknowledged, Enroute, In Progress, Completed, and so forth
- Shift log on and log off
- Break start and break complete
- POU start and POU complete

Siebel CRM sends all status updates that relate to a shift, break, POU, or activity to Oracle Real-Time Scheduler through a single composite. Siebel Field Service Integration to Oracle Real-Time Scheduler uses separate Business Process Execution Language (BPEL) processes within the composite to synchronize a shift, break, POU, or activity.

For more information, see Appendix A, "Integration Services."

Integration Flows That Synchronize a Status from Siebel CRM to Oracle Real-Time Scheduler

This topic describes integration flows that synchronize activity and related data from Siebel CRM to Oracle Real-Time Scheduler. It includes the following topics:

■ "Integration Flow That Synchronizes Activity and Related Data from Siebel CRM to Oracle Real-Time Scheduler" on page 40

Integration Flow That Synchronizes a Shift, Break, or POU Status from Siebel CRM to Oracle Real-Time Scheduler

The technician sends updates on the shift start time, stop time, and so forth. To optimize Activities, Siebel CRM must immediately send these changes to Oracle Real-Time Scheduler. For more information, see "Hook Points That Synchronize a Shift, Break, or POU Status" on page 85.

Integration Flow That Synchronizes a Shift Status from Siebel CRM to Oracle Real-Time Scheduler

Figure 7 illustrates the integration flow to synchronize a shift status from Siebel CRM to Oracle Real-Time Scheduler. This flow uses Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP) for all interactions with Siebel CRM and Oracle Real-Time Scheduler.

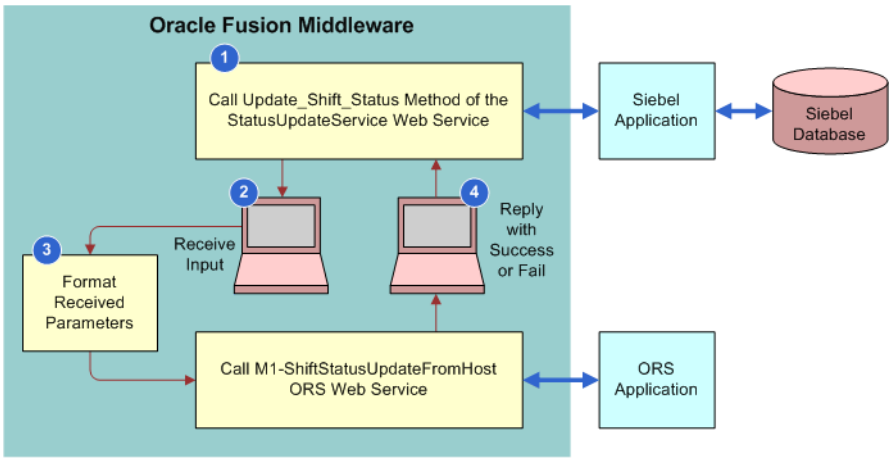

Figure 7. Integration Flow to Synchronize a Shift Status from Siebel CRM to Oracle Real-Time Scheduler

Explanation of Callouts

The integration flow to synchronize a shift status from Siebel CRM to Oracle Real-Time Scheduler includes the following steps:

1. If the status update is for a shift, then Siebel CRM makes a Web service call to the Update_Shift_Status method of the StatusUpdatesService Web service.
2. The mediator forwards the request to the BPEL flow that handles the shift status updates.
3. The BPEL flow calls the M1-ShiftStatusUpdateFromHost Web service method.

4 Oracle Real-Time Scheduler sends a success or fail reply. The output of a status update is success or failure. Oracle Real-Time Scheduler sends the result synchronously to Siebel CRM.

Integration Flow That Synchronizes a Break Status from Siebel CRM to Oracle Real-Time Scheduler

The integration flow to synchronize a break status from Siebel CRM to Oracle Real-Time Scheduler is the same as the flow illustrated in Figure 7 on page 38, with the following differences:

- If the status update is for a SHIFT_BREAK, then Siebel CRM makes a Web service call to the Update_Break_Status method of the StatusUpdatesService Web service.
- The mediator forwards the request to the BPEL flow that handles break status updates.
- The BPEL flow calls the M1-ActivityMaintenance web service method to update the Break Status in Oracle Real-Time Scheduler.

Integration Flow That Synchronizes a POU Status from Siebel CRM to Oracle Real-Time Scheduler

The integration flow to synchronize a POU status from Siebel CRM to Oracle Real-Time Scheduler is the same as the flow illustrated in Figure 7 on page 38, with the following differences:

- If the status update is for a POU, then Siebel CRM makes a Web service call to the Update_POU_Status method of the StatusUpdatesService Web service.
- The mediator forwards the request to the BPEL flow that handles POU status updates.
- The BPEL flow calls the M1-ActivityMaintenance web service method to update the POU Status in Oracle Real-Time Scheduler.

WSDL Operation That Synchronizes a Shift, Break, or POU Status from Siebel CRM to Oracle Real-Time Scheduler

The following Web Service Definition Language (WSDL) operations synchronize a shift, break, or POU status from Siebel CRM to Oracle Real-Time Scheduler:

- Update_Shift_Status for shifts
- Update_Break_Status for breaks
- Update_POU_Status for POUs

XML Schema Definition (XSD) defines which Extensible Markup Language (XML) structure that this integration must follow when it sends the XML request during the Web service call. The input XSD is the same for all WSDL operations. For more information, see Appendix A, “Integration Services.”

Integration Flow That Synchronizes Activity and Related Data from Siebel CRM to Oracle Real-Time Scheduler

The integration flow to synchronize activity and related data from Siebel CRM to Oracle Real-Time Scheduler is the same as the flow illustrated in Figure 7 on page 38, with the following differences:

- If the status update is for activity data, then Siebel CRM makes a Web service call to the Update_Activity_Status method of the StatusUpdatesService Web service.
- The mediator forwards the request to the BPEL flow that handles status updates for activities.
- Instead of formatting received parameters, this flow uses a DVM to determine the status. For more information, see "DVM That Synchronizes Activity and Related Data from Siebel CRM to Oracle Real-Time Scheduler" on page 40.
- The BPEL flow calls the M1-ActivityStatusUpdatesFromHost Web Service.

DVM That Synchronizes Activity and Related Data from Siebel CRM to Oracle Real-Time Scheduler

Because the status in Siebel CRM does not necessarily map to the same status in Oracle Real-Time Scheduler, this integration keeps the mapping in a DVM in the middle tier. This integration chooses the Web service after it transforms the status to an Oracle Real-Time Scheduler status.

Table 30 describes the DVM to synchronize activity and related data from Siebel CRM to Oracle Real-Time Scheduler. For more information, see "Configuring Domain Value Maps" on page 105.

Table 30. DVM That Synchronizes Activity and Related Data from Siebel CRM to Oracle Real-Time Scheduler

DVM	Description
SBL_ORS_Endpoint_Definition.dvm	This integration uses the ActivityStatusUpdates DVM parameter.
Siebel_to_ORS_ActivityStatusMapping.dvm	Used to extract the Oracle Real-Time Scheduler equivalent of a Siebel status.

WSDL Operation That Synchronizes Activity and Related Data from Siebel CRM to Oracle Real-Time Scheduler

Siebel Field Service Integration to Oracle Real-Time Scheduler uses the Update_Activity_Status WSDL operation to synchronize activity and related data from Siebel CRM to Oracle Real-Time Scheduler.

For more information, see Appendix A, "Integration Services."

Web Services That Synchronize a Status from Siebel CRM to Oracle Real-Time Scheduler

This topic describes Siebel Web services that synchronize activity and related data from Siebel CRM to Oracle Real-Time Scheduler.

Outbound Web Service That Synchronizes Activity and Related Data from Siebel CRM to Oracle Real-Time Scheduler

Table 31 describes the outbound Web service that synchronize activity and related data from Siebel CRM to Oracle Real-Time Scheduler.

Table 31. Outbound Web Service That Synchronize Activity and Related Data from Siebel CRM to Oracle Real-Time Scheduler

Web Service Name	Proxy Business Service Name
StatusUpdatesWebServiceSoap	StatusUpdatesWebServiceSoap

The Data Type property is Integration Object for all method arguments of the StatusUpdatesWebServiceSoap Web service that synchronize activity and related data from Siebel CRM to Oracle Real-Time Scheduler.

Table 32 describes the arguments of the Update_Activity_Status Web service method.

Table 32. Arguments of the Update_Activity_Status Web Service Method

Name	Integration Object Name	Type
ACTIVITY_UPDATE_IN:in	ORSActivityStatus	Input
ACTIVITY_UPDATE_OUT:out	STATUS_UPDATE_RESPONSE	Output

Table 33 describes the arguments of the Update_Break_Status Web service method.

Table 33. Arguments of the Update_Break_Status Web Service Method

Name	Integration Object Name	Type
BREAK_UPDATE_IN:in	Integrated Shift Break POU	Input
BREAK_UPDATE_OUT:out	STATUS_UPDATE_RESPONSE	Output

Table 34 describes the arguments of the Update_POU_Status Web service method.

Table 34. Arguments of the Update_POU_Status Web Service Method

Name	Integration Object Name	Type
POU_UPDATE_IN:in	Integrated Shift Break POU	Input
POU_UPDATE_OUT:out	STATUS_UPDATE_RESPONSE	Output

Table 35 describes the arguments of the Update_Shift_Status Web service method.

Table 35. Arguments of the Update_Shift_Status Web Service Method

Name	Integration Object Name	Type
SHIFT_UPDATE_IN:in	Integrated Shift Break POU	Input
SHIFT_UPDATE_OUT:out	STATUS_UPDATE_RESPONSE	Output

6 How the Connector Updates Employee Data in Real Time

This chapter describes how the connector updates employee data in real time. It includes the following topics:

Overview of How the Connector Updates Employee Data in Real Time

The integration synchronizes employee data in real time from Siebel CRM to Oracle Real-Time Scheduler for service regions that are scheduled in Oracle Real-Time Scheduler. This synchronization occurs in the following situations:

- A user or a process adds a new employee record in Siebel CRM under a service region which is already synchronized to Oracle Real-Time Scheduler.
- A user or a process updates an existing employee record in Siebel CRM which is already synchronized to Oracle Real-Time Scheduler.

Siebel Field Service Integration to Oracle Real-Time Scheduler sets up each flow on a single Business Process Execution Language (BPEL) process and uses a switch to identify the flow according to the operation it calls in the Web Service Definition Language (WSDL). For information about setup flows for administrative data for this integration, see "Integration Flows That Update Employee Data in Real Time" on page 44.

This integration uses the AdminDataManagement_EBF project to update employee data in real time.

For more information, see Appendix A, "Integration Services."

Integration Flows That Update Employee Data in Real Time

This topic describes integration flows that update employee data in real time. It includes the following topics:

For more information, see "Hook Points That Update Employee Data" on page 87.

Integration Flow That Updates or Inserts an Employee Record

If Oracle Real-Time Scheduler is optimizing a service region, and if a new employee record is added to the service region, or if an update is made to an existing employee record, then Siebel CRM makes a Web service call to Oracle Fusion Middleware. This middleware creates or updates the employee record in Oracle Real-Time Scheduler.

Figure 8 on page 45 illustrates the integration flow to update or insert an employee record. This flow uses Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP) for all interactions with Siebel CRM and Oracle Real-Time Scheduler.

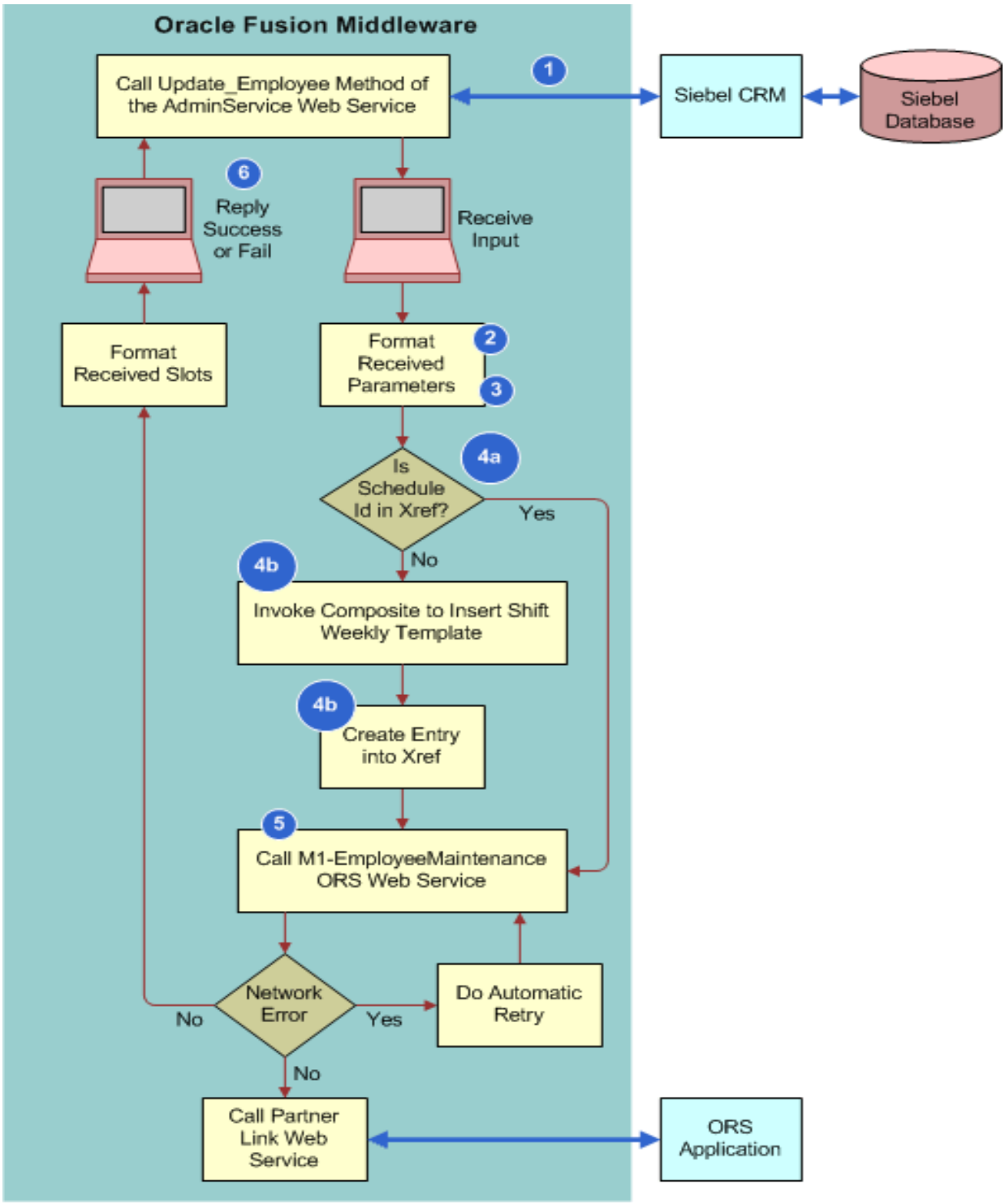

Figure 8. Integration Flow to Update or Insert an Employee Record

Explanation of Callouts

To transfer an employee record, a significant amount of processing occurs in the composite. For brevity, this topic describes an overview of this flow. To transfer an employee record, Siebel Field Service Integration to Oracle Real-Time Scheduler does the following:

1. Receives input from Siebel CRM.
2. Determines if the employee record is associated with a valid start and end location:
   - If the address association is present on the employee record, then it continues to Step 3.
   - If a valid start or end location are not present, then it sends a reply that the start address or end address is not valid, and then immediately exits this process. This integration assumes the following:
     - It is sufficient to check if the employee is associated with a start address and end address, and that it is not necessary to examine the address for authenticity.
3. Determines if the employee time zone is present in the input data, and then does one of the following:
   - If the employee time zone is present, then it uses this time zone value.
   - If the employee time zone is not present, then it uses the time zone from the service region.

   For more information, see "How Oracle Real-Time Scheduler Handles the Time Zone" on page 49.
4. Determines if the employee schedule ID is present in the input data, and then does one of the following:
   a. If the employee schedule ID is present, then it refers to the cross reference table to verify whether the Schedule has been transferred to Oracle Real-Time Scheduler.
   b. If not, then the integration will call the Siebel inbound web service to get the Schedule details and transfer the Schedule to Oracle Real-Time Scheduler by invoking another composite. An entry will be made in the Cross Reference table for this Schedule ID.
5. Calls the M1-EmployeeMaintenance Web service method to update the employee in Oracle Real-Time Scheduler.

   If the start date of the scheduling availability of the employee is not present in the request, then the flow sets the employee start date to the current system date before it makes the Web service call to Oracle Real-Time Scheduler.

   The Employee Address to be transferred will be decided based on the Start Shift From and End Shift At fields in the employee record. If the Start Shift From field value is Home, then the integration will transfer the Start Home Address. If the Start Shift From field value is Depot, then the integration will transfer Start Depot Address. Similarly, the End Home Address or End Depot Address values will be transferred based on the End Shift At field value.

6 Sends the output from the M1-EmployeeMaintenance Web Service to Siebel CRM.

Figure 9 illustrates the integration flow to update or insert a Shift Weekly Template. This flow uses Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP) for all interactions with Siebel CRM and Oracle Real-Time Scheduler.

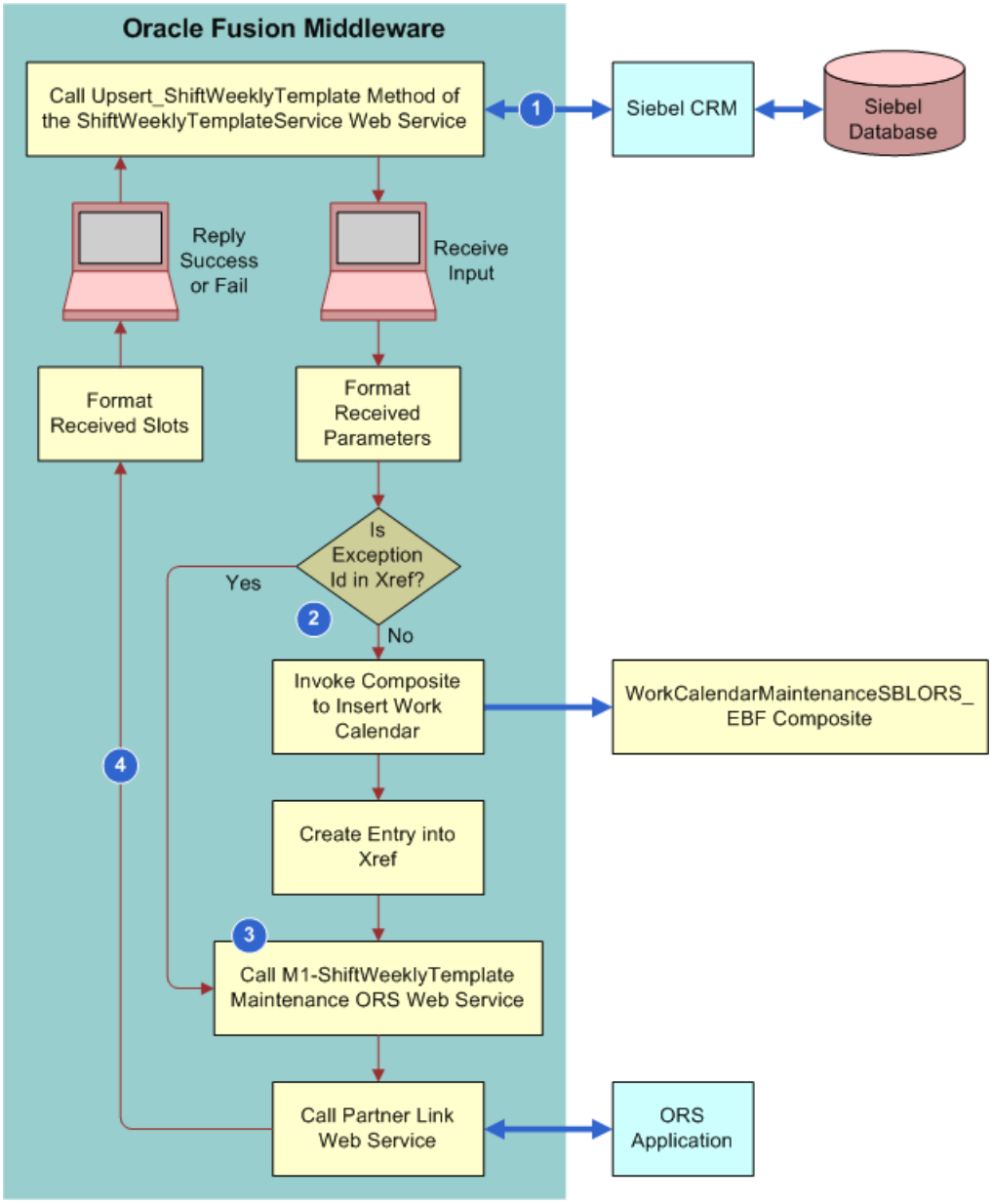

Figure 9. Integration Flow That Updates or Inserts a Shift Weekly Template

Explanation of Callouts

1 Receives input from Siebel CRM.

2 Determines if the Exception Id is present in the input data and then does one of the following:

- If the Exception Id is present, then it refers to the cross reference table to verify whether the Exception has been transferred to Oracle Real-Time Scheduler.
- If not, then the integration will call the Siebel Inbound web service to get the Exception Details along with Exception hours and transfer the Exception to Oracle Real-Time Scheduler by invoking another composite. An entry will be made in the Cross Reference table for this Exception Id.

3 Calls M1-ShiftWeeklyTemplateMaintenance Service to create Shift Weekly Template in Oracle Real-Time Scheduler based on the following logic:

- For each day, integration will consider the Shift Start time of the record with Shift Start Flag set to Y and consider the Shift End Time of the last record of that same day.
- Any time gap between the same day records will be considered as Breaks and the Shift Weekly Template will be updated accordingly at Oracle Real-Time Scheduler.

4 Sends the output from the M1-ShiftWeeklyTemplateMaintenance Web Service to Siebel CRM.

Figure 10 illustrates the integration flow to update or insert a work calendar. This flow uses Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP) for all interactions with Siebel CRM and Oracle Real-Time Scheduler.

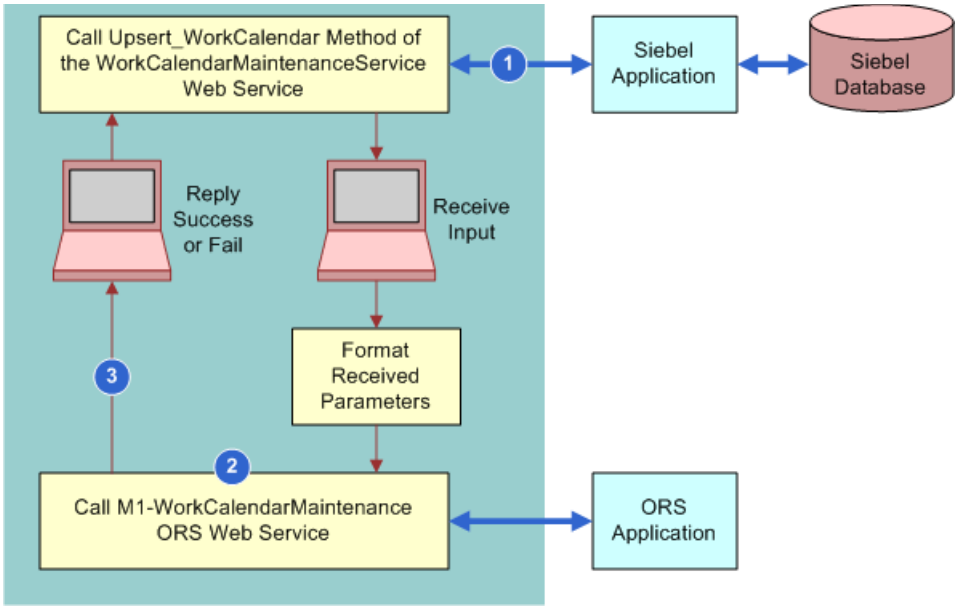

Figure 10. Integration Flow to Update or Insert a Work Calendar

Explanation of Callouts

1. Receives input from Siebel CRM.
2. Calls M1-WorkCalendarMaintenance Service to create a Work Calendar in Oracle Real-Time Scheduler. This integration supports only Non-working exceptions at the Schedule level. Exception Hours with Working Flag set to Y will not be synchronized to Oracle-Real Time Scheduler.
3. Sends the output from the M1-WorkCalendarMaintenance Web Service to Siebel CRM.

How Oracle Real-Time Scheduler Determines Whether or Not to Synchronize a Schedule

The cross reference stores the Schedule ID of those Schedules which are already transferred to Oracle Real-Time Scheduler.

When Siebel Field Service Integration to Oracle Real-Time Scheduler transfers an employee record from Siebel CRM to Oracle Real-Time Scheduler, it determines whether the Schedule ID associated with the Employee exists in the cross reference and then does the following:

- If it finds an entry, then it does not invoke the flow to transfer the Schedule to Oracle Real-Time Scheduler.
- If it does not find an entry, then it does the following:
  - Calls a Siebel Inbound Web Service by passing the Schedule ID as an input parameter to get the details of that Schedule.
  - Invokes the integration flow with the Schedule details received to transfer the Schedule to Oracle Real-Time Scheduler.
  - Writes an entry into the cross reference that states whether this integration created the schedule in Oracle Real-Time Scheduler.

How Oracle Real-Time Scheduler Handles the Time Zone

This integration maps the Standard Abbreviation as well as the Name of the Siebel side Time Zone for the corresponding Time Zone code in the SBL_ORS_TimeZoneMapping.dvm in Oracle Real-Time Scheduler.

How Oracle Real-Time Scheduler Associates an Employee Record with a Service Area

This integration uses the *ORS Default Service Area with Srv regn* parameter in the domain value map (DVM). This integration does the following:

- If you set this parameter to YES, then this integration associates the transferred employee to a Service Area with the same name as the service region.
- If you set this parameter to NO, then this integration does not associate the transferred employee to a Service Area.

- If this parameter is not defined, then this integration will associate the employee with the Service Area that is the same as the Service Region defined in the Preferred Service Area.

How Oracle Real-Time Scheduler Uniquely Identifies an Employee Record

This integration maps the Siebel ROW_ID to the Host External ID in Oracle Real-Time Scheduler to uniquely identify an employee record.

How Oracle Real-Time Scheduler Synchronizes Employee Exceptions

Employee Exceptions are sent along with the Employee details, but they are handled differently in Oracle Real-Time Scheduler based on their type.

- **Non-working Exception Without Address.** Non-working exceptions without any address associated with it will be considered as Leaves in Oracle Real-Time Scheduler.
- **Non-working Exceptions With Address.** Non-working exceptions with an address associated with it are considered as a Meeting type Period of unavailable in Oracle Real-Time Scheduler.
- **Working Exception.** For Working exceptions (with Working flag set to Y in Siebel CRM), new Shifts will be created for that Crew in Oracle Real-Time Scheduler.

How Oracle Real-Time Scheduler Synchronizes Employee Skills

Employee Skills are sent along with the Employee details but they are handled differently in Oracle Real-time Scheduler. Skills will be created as Attributes first in Oracle Real-time Scheduler and then will be associated with an Employee or Crew. If no expertise Level is defined in Siebel CRM for a Skill Item, then this integration will use SBL_Expertise_Code_Definition.dvm to look up the value provided for DEFAULT parameter and pass that skill level to Oracle Real-time Scheduler.

DVM That Updates or Inserts an Employee Record

Table 36 describes the DVM to update or insert an employee record.

Table 36. DVM That Updates or Inserts an Employee Record

DVM	Description
SBL_ORS_Parameter_Definition.dvm	This integration uses the following DVM parameters: ■ ORS-Default Service Area with Srv Regn ■ ORS-Relative Rank of DEFAULT Expertise
SBL_Expertise_Code_Definition.dvm	This integration uses this DVM to get the expertise level for corresponding expertise code from Siebel CRM.

WSDL That Updates or Inserts an Employee Record

This integration uses the Upsert_Employee WSDL operation to update or insert an employee record. For more information, see Appendix A, "Integration Services".

Web Services That Update Employee Data in Real Time

This topic describes Siebel Web services that update employee data in real time.

Outbound Web Service That Updates Employee Data in Real Time

Table 37 describes the outbound Web service that update employee data in real time.

Table 37. Outbound Web Service That Updates Employee Data in Real Time

Web Service Name	Proxy Business Service Name
adminDataMgmtWebService	adminDataMgmtWebServiceSoap

The Data Type property is Integration Object for all method arguments of the adminDataMgmtWebService Web service that update employee data in real time.

Table 38 describes the arguments of the Upsert_Employee Web service method.

Table 38. Arguments of the Upsert_Employee Web Service Method

Name	Integration Object Name	Type
Upsert_Employee_IN:in	EmployeeORS	Input
Upsert_Employee_OUT:out	EmployeeORS	Output

Integration Objects That Update Employee Data in Real Time

Table 39 describes the integration objects that update employee data in real time.

Table 39. Integration Objects That Update Employee Data in Real Time

Integration Object Name	Business Object	Type
EmployeeORS	Employee	Siebel Business Object

Table 40 describes the properties of the EmployeeORS integration object.

Table 40. Properties of the EmployeeORS Integration Object

Integration Component	Business Component	Parent Integration Component
Employee	Employee	Not applicable
Employee Skill	Employee Skill	Employee
Employee Skill Item	Employee Skill Item	Employee Skill
Employee Shift Exception Hour	Employee Shift Exception Hour	Employee

7 How the Connector Retrieves Geocodes

This chapter describes how the connector retrieves geocodes for a single address or for multiple addresses. It includes the following topics:

Overview of How the Connector Retrieves Geocodes

To retrieve geocodes, the connector uses the following composites:

- **Real-time.** Siebel Field Service integration uses this flow to retrieve a geocode for a single address record, which typically occurs when a customer service representative (CSR) creates a new address record. Siebel CRM sends the address details to the ORS Geocode Server through the Oracle Fusion Middleware. Siebel Field Service Integration to Oracle Real-Time Scheduler sends a reply to Siebel CRM that includes a list of matching geocoded addresses. The user can choose one of the records.
- **Batch mode.** Siebel Field Service integration uses this flow to retrieve geocodes for multiple addresses, which typically occurs during initial setup when you must retrieve geocodes for a large number of addresses. Batch mode is an asynchronous flow. It uses the Oracle Fusion Middleware database adapter to directly update geocodes in the Siebel database.

For more information, see Appendix A, "Integration Services."

Integration Flows That Retrieve Geocodes

This topic describes integration flows that retrieve geocodes. It includes the following topics:

For more information, see Appendix A, "Integration Services."

Integration Flow That Retrieves a Geocode for a Single Address in Real-Time

Siebel Field Service integration retrieves a geocode for a single address in real-time when it must send a geocode query for a particular address.

Figure 11 illustrates the integration flow to retrieve a geocode for a single address in real-time. This flow uses Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP) for all interactions with Siebel CRM and Oracle Real-Time Scheduler.

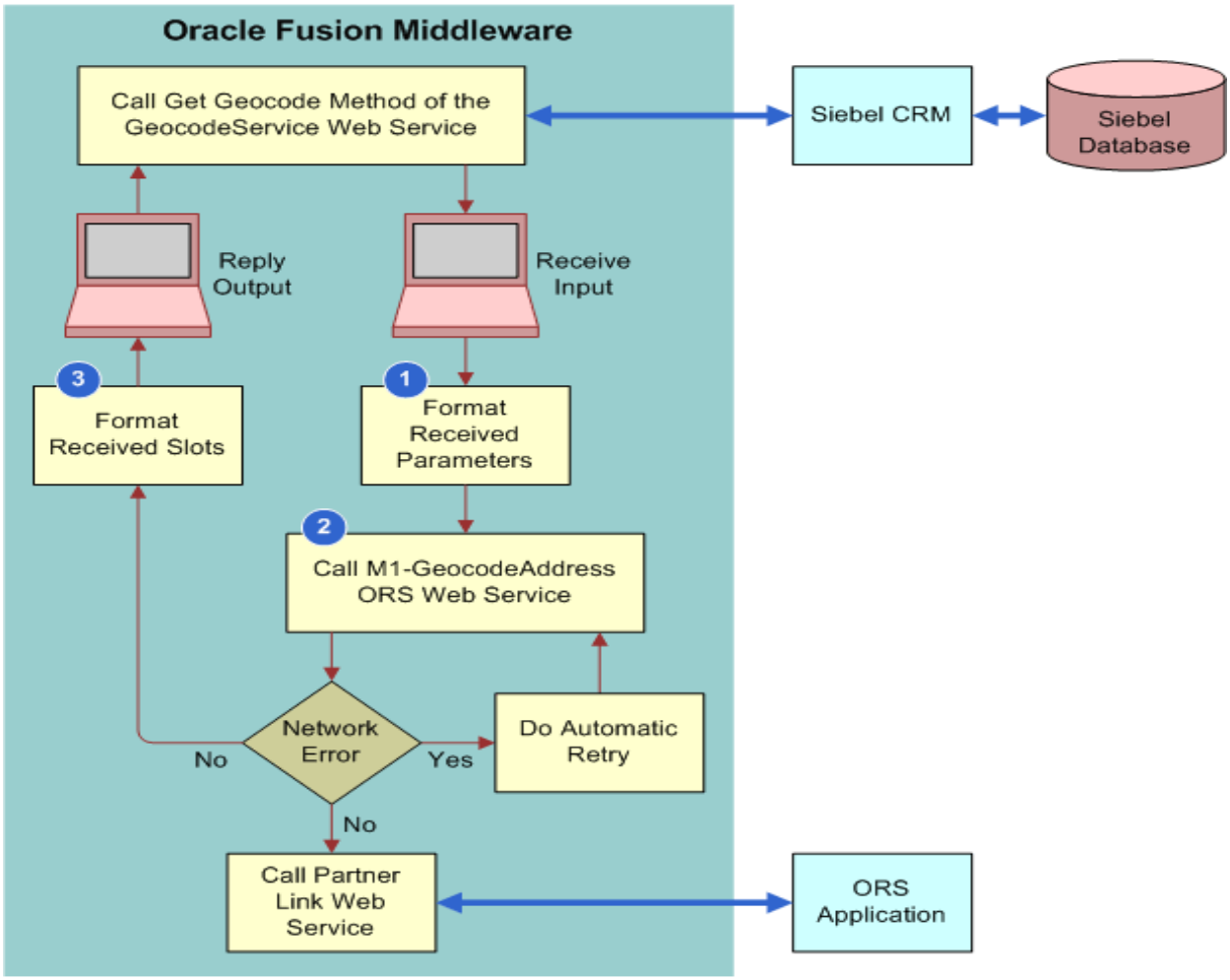

Figure 11. Integration Flow That Retrieves a Geocode for a Single Address in Real-Time

Explanation of Callouts

The integration flow that retrieves a geocode for a single address in real-time includes the following steps:

1. The Business Process Execution Language (BPEL) process transforms an input address to fit the specifications that the M1-GeocodeAddress ORS Web service operation requires.
2. The BPEL flow calls the ORS Web service and then receives the reply. The reply can contain zero, one, or more geocodes depending on the granularity and accuracy of the address.
3. The BPEL flow sends the geocodes it receives as a reply to the Web service call. The BPEL flow does not attempt to directly update the Siebel database record with the geocodes.

This integration uses the RealtimeSingleAddressGeoCoding project to retrieve a geocode for a single address in real-time.

WSDL That Retrieves a Geocode for a Single Address in Real-Time

This integration uses the RealTimeGEOCoding Web Service Definition Language (WSDL) operation to retrieve a geocode for a single address in real-time. For more information, see Appendix A, "Integration Services."

How Siebel CRM Makes Sure Addresses Are Compatible

Some countries use address formats that include districts, provinces, counties, and so forth. To make these addresses compatible with the address format in Oracle Real-Time Scheduler, the BPEL process performs the following mappings before it calls the Oracle Real-Time Scheduler geocode Web service:

- Maps county to state
- Maps province to state
- Maps district to city
- Maps metro area to city

Integration Flow That Retrieves Geocodes for Multiple Addresses

Figure 12 illustrates the integration flow that retrieves geocodes for multiple addresses.

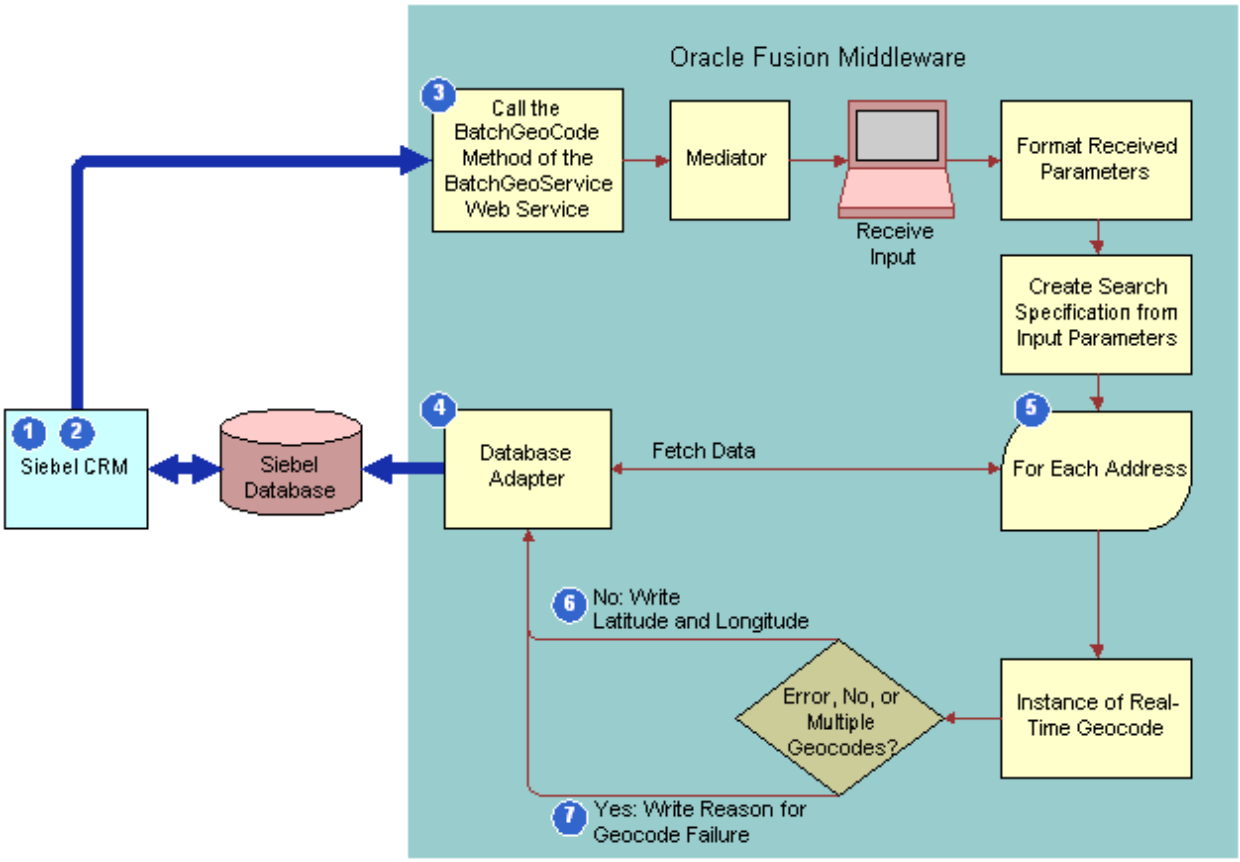

Figure 12. Integration Flow That Retrieves Geocodes for Multiple Addresses

Explanation of Callouts

The integration flow that retrieves geocodes for multiple addresses includes the following steps:

1. In Siebel CRM, the user queries for the addresses for which this integration must retrieve geocodes, and then clicks Bulk Geocode.
2. Siebel CRM changes the value in the Geocode Sync Description field of the address record to Geocoding In Progress.
3. Siebel CRM calls the BatchGeoCode method of the BatchGeoService Web service in Oracle Fusion Middleware. It uses Geocoding In Progress as the input.
4. A database adapter in the BPEL flow retrieves all the address records in the Siebel address table that contain Geocoding In Progress in the Geocode Sync Description field.

5 The BPEL flow reads the batch message and then parses the batch message into individual address messages so that it can retrieve a geocode for each individual address.

   To retrieve a geocode for a single address, the Bulk Geocoding BPEL process calls the real-time geocode BPEL flow for each address. For more information, see "Integration Flow That Retrieves a Geocode for a Single Address in Real-Time" on page 54.

6 If Oracle Real-Time Scheduler returns only a single set of geocodes, then the BPEL process uses the database adapter to add the latitude and longitude values to the address table in the Siebel database.

7 If Oracle Real-Time Scheduler returns multiple geocodes or no geocode, then the database adapter inserts an error message in the error column of the address table in the Siebel database.

WSDL That Retrieves Geocodes for Multiple Addresses

To retrieve geocodes for multiple addresses, Siebel CRM uses the BatchGeoCode WSDL operation. For more information, see Appendix A, "Integration Services."

Web Services That Retrieve Geocodes for a Single Address

This topic describes the Siebel Web services that retrieve geocodes for a single address.

Outbound Web Services That Retrieve Geocodes for a Single Address

Table 41 describes the outbound Web services that retrieve geocodes for a single address.

Table 41. Outbound Web Services That Retrieve Geocodes for a Single Address

Web Service Name	Proxy Business Service Name
BatchGeoService	BatchGeoCode_port
RealTimeService	RealTimeGEO

The Data Type property is Integration Object for all method arguments of the BatchGeoCode Web service that retrieves geocodes for a single address.

Table 42 describes the arguments of the BatchGeoCode Web service method.

Table 42. Arguments of the BatchGeoCode Web Service Method

Name	Integration Object Name	Type
Batch_Geocode_Query:request	SearchSpec	Input

Table 43 describes the arguments of the RealTimeGEOCoding Web service method.

Table 43. Arguments of the RealTimeGEOCoding Web Service Method

Name	Integration Object Name	Type
RTGEOCODING_IN:parameters	SWIAddressIO	Input
RTGEOCODING_OUT:parameters	ListOfSwiaddressio	Output

Integration Objects That Retrieve Geocodes for a Single Address

Table 44 describes the integration objects that retrieve geocodes for a single address.

Table 44. Integration Objects That Retrieve Geocodes for a Single Address

Integration Object Name	Business Object	Type
SWIAddressIO	CUT Address	Siebel Business Object
ListOfSwiaddressio	Not applicable	XML
SearchSpec	Not applicable	XML

Table 45 describes the properties of the SWIAddressIO integration object.

Table 45. Properties of the SWIAddressIO Integration Object

Integration Component	Business Component	Parent Integration Component
CUT Address	CUT Address	Not applicable

Table 46 describes the properties of the ListOfSwiaddressio integration object.

Table 46. Properties of the ListOfSwiaddressio Integration Object

Integration Component	Business Component	Parent Integration Component
ListOfSwiaddressio	ListOfSwiaddressio	Not applicable
CutAddress	CutAddress	ListOfSwiaddressio

Table 47 describes the properties of the SearchSpec integration object.

Table 47. Properties of the SearchSpec Integration Object

Integration Component	Business Component	Parent Integration Component
SearchSpec	SearchSpec	Not applicable

8 How the Connector Performs a Service Region Data Transfer

This chapter describes how the connector performs a service region data transfer. It includes the following topics:

Overview of How the Connector Performs a Service Region Data Transfer

The integration flow that performs a service region data transfer allows you to migrate existing service regions and associated employees and eligible activities to an Oracle Real-Time Scheduler Service Area. You can also use this integration flow to migrate a new service region to Oracle Real-Time Scheduler. The new service region is scheduled in Siebel Scheduler along with employees who are associated with the new service region.

Once the Service Region transfer is complete, the corresponding Service Area created in Oracle Real-Time Scheduler should be manually associated to respective Scheduler Area in Oracle Real-Time Scheduler.

Integration Flow That Performs a Service Region Data Transfer

Figure 13 on page 63 illustrates the integration flow that performs a service region data transfer. For more information, see the following topics:

- Appendix A, “Integration Services”
- “Hook Points That Update Employee Data” on page 87

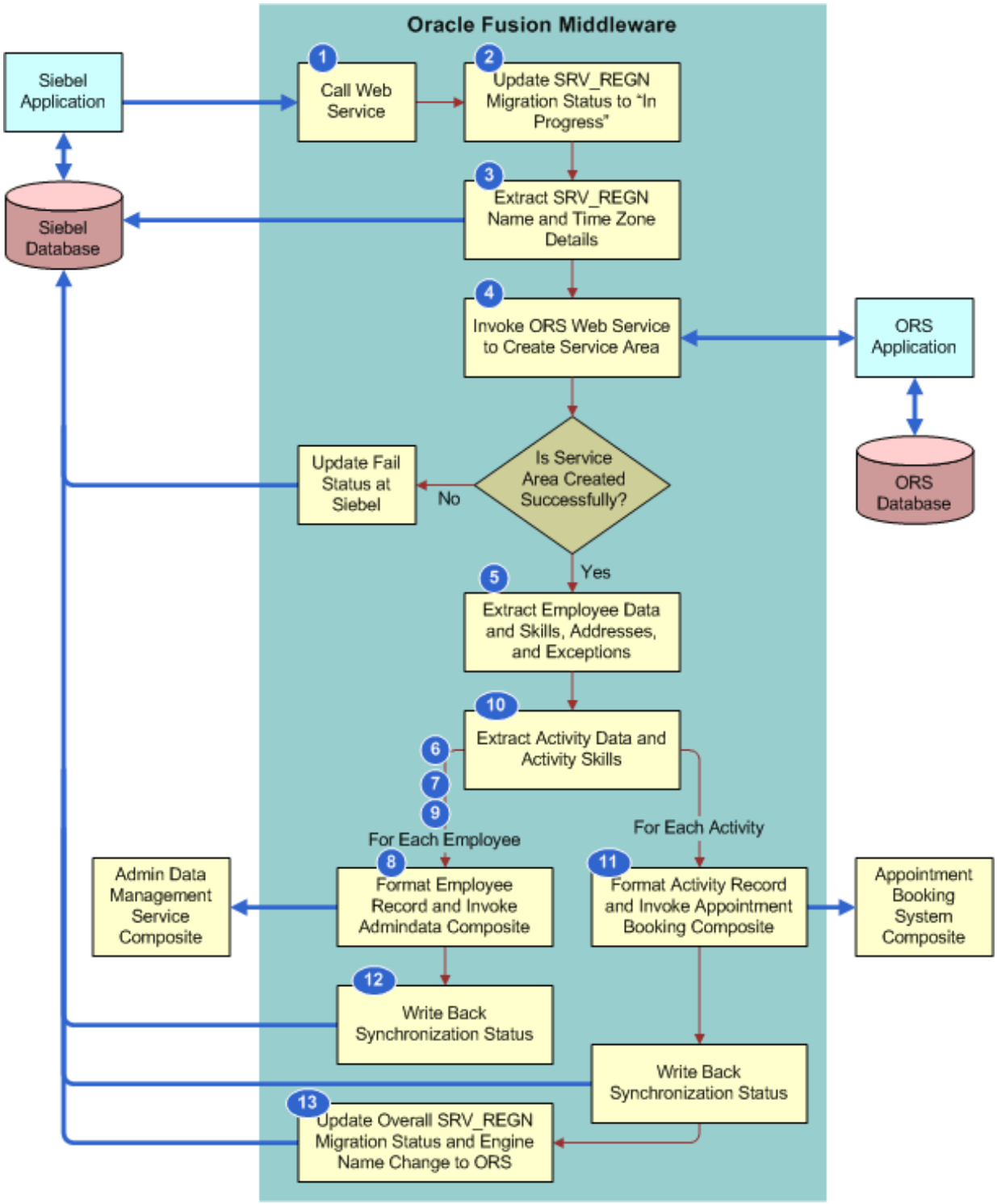

Figure 13. Integration Flow to Perform a Service Region Data Transfer

Explanation of Callouts

The integration flow that performs a service region data transfer includes the following steps:

1. Siebel CRM calls the Web service that starts the process. It supplies to this Web service the Id of the service region that this integration must transfer. This Id is the ROW_ID of the service region in the Siebel database. Siebel CRM uses Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP) to make this call.
2. The Business Process Execution Language (BPEL) flow sets the value of the Synchronization Status field of the service region record to *Migration to ORS in progress*. To perform this work directly on the Siebel database, it uses a database adapter that is connected to the Siebel database.
3. The BPEL flow extracts the name and description of the service region from the Siebel database. To query the Siebel database, it uses a database adapter that is configured in query mode and it uses the service region Id as the query specification.
4. To create an Oracle Real-Time Scheduler Service Area, the BPEL flow makes Web Service calls to Oracle Real-Time Scheduler.
5. Extracts all data that is related to the employee from the Siebel database. To extract all employee data, this flow joins all the relevant Siebel database tables. This data includes details of the start address, end address, associated skills, associated time zone, and so forth. It does not extract the following employee records:
   - Employee records that contain Y in the ORS sync flag
   - Employees records that contain a scheduling availability end date that occurs prior to the next day of the system time that is in effect during the data transfer
6. Reads the first employee record and determines the start address and end address based on the Start Shift From and End Shift At employee fields. If the values for these fields are Home, then the integration will consider the Home address. If the value of these fields is Depot, then the integration will consider the Depot address.

   The integration also validates whether the start and end address have latitude and longitude values.
7. If the address records are not present, then this integration does not synchronize the employee record in the Siebel database with the ORS database. Instead, it saves a synchronization error in the employee record in the Siebel database. This error states that the start address and the end address are missing.
8. This integration will insert the Employee record along with Start and End Address, Employee Skills, and Employee Exception Hours into the ORS database by calling the UPSERT_EMPLOYEE method of the AdminDataManagement_EBF composite.
9. Repeats Step 6 for each subsequent employee record until it finishes processing all employee records.
10. After this flow processes all employee records, the BPEL flow gets the activity records and associated data from the Siebel database, except for activities that meet any of the following conditions:
    - Any activity in the service region whose ORS synchronization status is N and the status is not Done, Cancelled, or Completed.

- Any activity that is expired. An activity is considered not to have expired if any of the following situations are true:
  - The latest start date for the activity will occur at any time in the future.
  - The latest start date is null.
  - The planned end date will occur in the future.
  - The planned end is null.

  This integration considers any time that occurs after the system time that is in effect during the data transfer as a future date.

11 To write data for each activity record to the ORS database, it uses the Upsert_Activity operation of the AppointmentBookingSystemSBLORS_EBF composite. This flow uses a native binding to the AppointmentBookingSystemSBLORS_EBF composite.

12 Saves the result of the synchronous call in the Siebel activity record.

13 Saves the overall status of the data transfer in the Synchronization Status field of the Service Region. It uses the following logic:

- If this flow successfully transfers all employees, addresses, and activities, then it does the following work:
  - Sets the Synchronization Status field to the following value:

    `>>>>>Successfully migrated to ORS`
  - Inserts the following value in the Engine field:

    `>>>>>ORS`
- If this flow cannot successfully transfer at least one employee, address, or activity, then it does the following work:
  - Sets the Synchronization Status field to the following value:

    `>>>>>Partially migrated to ORS`
  - Inserts the following value in the Engine field:

    `>>>>>ORS`
- If this flow encounters an exception, then it does the following work:
  - Sets the Synchronization Status field to the following value:

    `>>>>>Transfer to ORS aborted due to exception`
  - Does not change the value in the Engine field

How Siebel Field Service Integration to Oracle Real-Time Scheduler Avoids Network Overhead

Reading the Siebel database and updating the employee records, address records, or activity records for each individual record causes network overhead. To avoid this situation, this integration uses a domain value map (DVM) parameter that contains a counter. As long as the counter remains below a threshold, the service region data transfer flow appends the input of the calling activity with the synchronization message. If the counter reaches the value specified in the parameter, or if it is the last record, then this flow writes all the updates to the Siebel database in a batch.

DVM That Performs a Service Region Data Transfer

Table 48 describes the DVM that performs a service region data transfer.

Table 48. DVM That Performs a Service Region Data Transfer

DVM	Description
SBL_ORS_Parameter_Definition.dvm	This integration uses the following DVM parameters: ■ Default Batch Size - Employee ■ Default Batch Size - Activity ■ Default Batch Size - Address

Composite This Integration Uses During a Service Region Data Transfer

This integration uses the following project to perform a service region data transfer:

ServiceRegionCutoverToORS_EBF

This project supports multiple daylight savings rules.

Web Services That Perform a Service Region Data Transfer

This topic describes Siebel Web services that perform a service region data transfer.

Outbound Web Service That Performs a Service Region Data Transfer

Table 49 describes the outbound Web service that performs a service region data transfer.

Table 49. Outbound Web Service That Performs a Service Region Data Transfer

Web Service Name	Proxy Business Service Name
Serviceregionmigrationorch+ estrator_client_ep	ServiceRegionMigrationOrchestrator

Table 50 describes the arguments of the process Web service method.

Table 50. Arguments of the Process Web Service Method

Name	Data Type	Integration Object Name	Type
ServiceRegionMigrationOrchestratorReques tMessage:payload	Integration Object	input	Input

Integration Objects That Perform a Service Region Data Transfer

Table 51 describes the integration objects that perform a service region data transfer.

Table 51. Integration Objects That Perform a Service Region Data Transfer

Integration Object Name	Business Object	Type
input	Not applicable	XML

Table 52 describes the properties of the input integration object.

Table 52. Properties of the Input Integration Object

Integration Component	Business Component	Parent Integration Component
input	input	Not applicable

9 How the Connector Performs a Service Region Rollback

This chapter describes how the connector performs a service region rollback. It includes the following topics:

Overview of How the Connector Performs a Service Region Rollback

If an unexpected issue occurs with scheduling a service region in Oracle Real-Time Scheduler, then the service region rollback flow allows you to revert to scheduling the service region in Siebel Scheduler.

Integration Flow That Performs a Service Region Rollback

Figure 14 on page 71 illustrates the integration flow that performs a service region rollback.

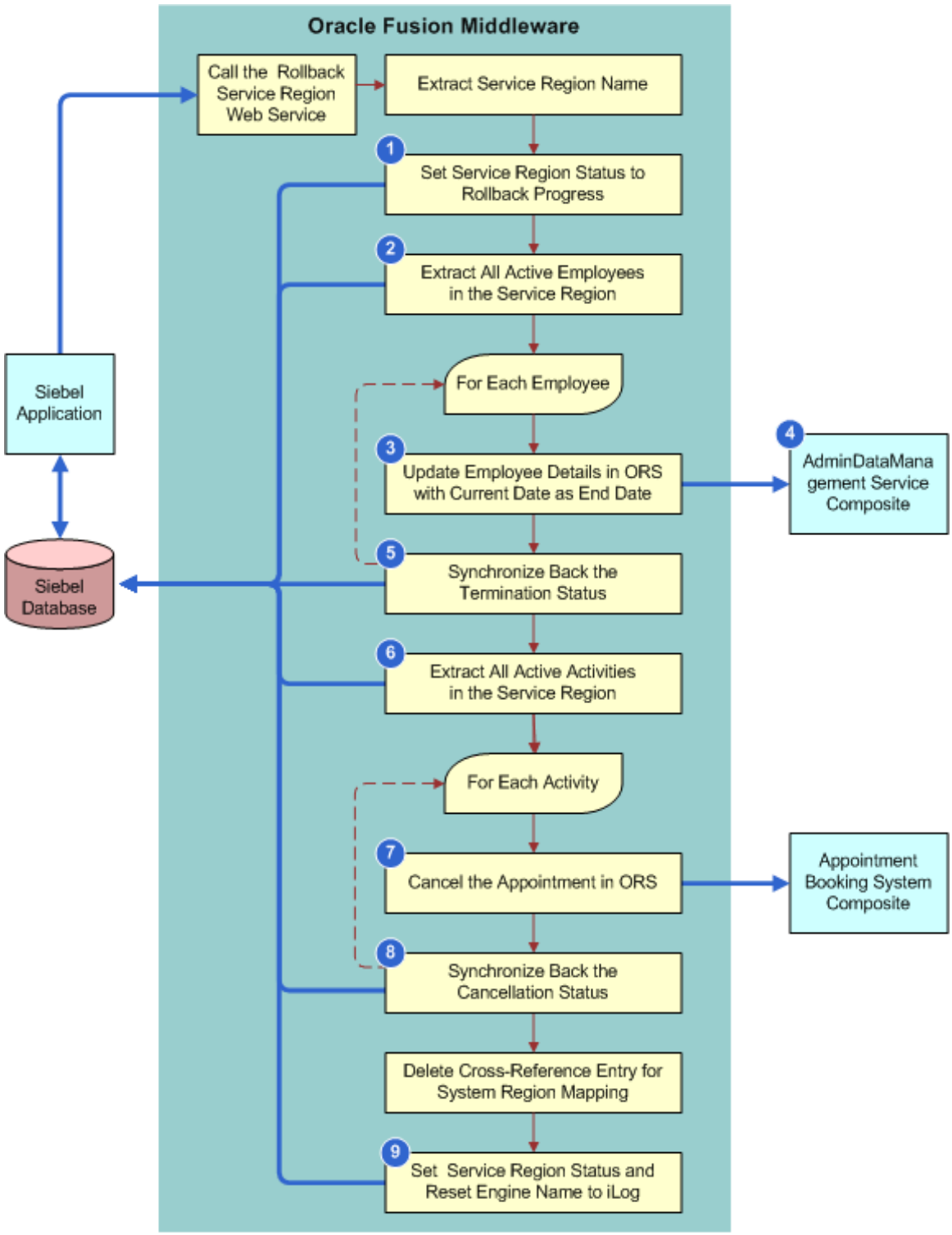

Figure 14. Integration Flow to Perform a Service Region Rollback

Explanation of Callouts

The integration flow that performs a service region rollback includes the following steps:

1. Sets the rollback progress:
   a. Sets the synchronization description field of the service region record in the Siebel database to the following value:

      >>>Rollback from ORS in progress

   b. Gets the name of the service region from the Siebel database.
2. Extracts all employees who belong to the service region whose scheduling availability end date does not occur prior to the system time and date that is in effect at the point of the rollback.
3. For each employee that this flow extracts, calls the Upsert_Employee operation of the AdminDataManagement_EBF composite.
4. The Upsert_Employee operation sets the Employee Enable to DSBL for all the Crews and Resources belong to the Service Area at Oracle Real-Time Scheduler.
5. If the Upsert_Employee operation returns a success, then this flow does the following in the employee record in the Siebel database:
   - Saves the following synchronization message in the employee record:

     >>>Employee Terminated in ORS

   - Sets the ORS Sync Success flag for the employee to N.
6. Extracts all activities in the service region. It uses the same rules that it uses during a data transfer to identify the activities to extract from the database. The activity cannot occur in the past, and it cannot have already been serviced.
7. For each activity extracted, calls the Cancel_Appointment operation of the AppointmentBookingSystemSBLORS_EBF composite.
8. If the cancellation is successful, then this flow sets the following fields of the activity:
   - Synchronization status to Activity Cancelled in ORS
   - Sync Success flag to N
9. Sets the service region status to the following value and sets the engine name to iLog:

   Rollback from ORS complete

This flow uses a direct database update or query for all interactions with the Siebel database.

Web Services That Perform a Service Region Rollback

This topic describes Siebel Web services that perform a service region rollback.

Outbound Web Service That Performs a Service Region Rollback

Table 53 describes the outbound Web service that performs a service region rollback.

Table 53. Outbound Web Service That Performs a Service Region Rollback

Web Service Name	Proxy Business Service Name
ServiceRegionRollback_ep	ServiceRegionRollbackOrchestrator

Table 54 describes the arguments of the Rollback_SrvRegn Web service method.

Table 54. Arguments of the Rollback_SrvRegn Web Service Method

Name	Data Type	Integration Object Name	Type
ServiceRegionRollbackOrchestratorRequest Message:payload	Integration Object	SrvRegnRollbackInput	Input

Integration Object That Performs a Service Region Rollback

Table 55 describes the integration object that performs a service region rollback.

Table 55. Integration Object That Performs a Service Region Rollback

Integration Object Name	Business Object	Type
SrvRegnRollbackInput	Not applicable	XML

Table 56 describes the properties of the SrvRegnRollbackInput integration object.

Table 56. Properties of the SrvRegnRollbackInput Integration Object

Integration Component	Business Component	Parent Integration Component
SrvRegnRollbackInput	SrvRegnRollbackInput	Not applicable

10 Extending the Connector

This chapter describes how to extend the connector. It includes the following topics:

Overview of Extending the Connector

A *hook point* is a service you can use to extend a composite. It uses input that you provide to communicate with a composite and it returns a result from the composite. You can use hook points to extend the following items:

- Schema
- Mediator
- BPEL flow

Note the following:

- This integration uses each hook point as a placeholder in a BPEL flow.
- A hook point does not include any functionality or perform any manipulation other than to access a composite.
- Every hook point uses a synchronous request and reply model.
- You can use hook points to extend a direct integration flow.
- This integration disables every hook point by default. To use a hook point, you must set a DVM (domain value map) parameter for the hook point parameter to true. For more information, see "Parameters You Must Set in the Parameter Definition File" on page 90.
- If you use a hook point, then make sure you avoid a name space conflict in the implementation.

Figure 15 illustrates a generic flow that includes hook points.

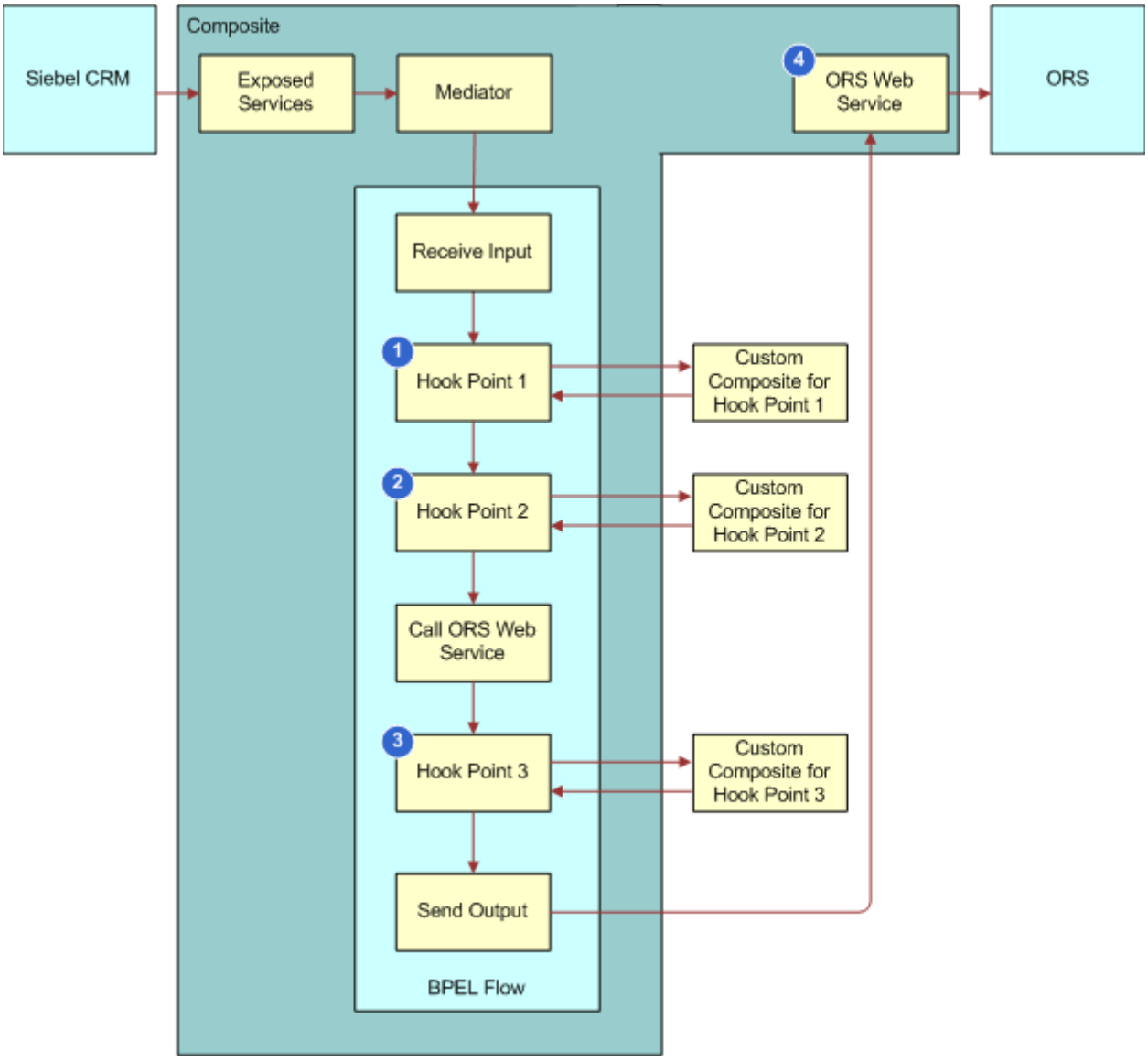

Figure 15. Generic Flow That Includes Hook Points

Explanation of Callouts

This integration performs the following work for most hook points:

1 **Hook Point 1.** The input for hook point 1 is the Siebel request message. For example, the Siebel input for the appointment booking flow is absInput. The output of hook point 1 is the extended Siebel request message.

2 **Hook Point 2.** The input for hook point 2 is the extended Siebel request message to the composite and the Oracle Real-Time Scheduler request message for the operation. The composite that calls the hook point determines this operation. The output for hook point 2 is the Oracle Real-Time Scheduler requested message for the called operation. For more information, see "Hook Point Operations That This Integration Calls" on page 88.

3 **Hook Point 3.** The input for hook point 3 is the Oracle Real-Time Scheduler response message. The composite that calls the hook point determines this output. For example, the Siebel output for the appointment booking flow is absReply. The output for hook point 3 is the Siebel response message.

4 **ORS Web Service.** The external Web service that the composite calls.

Process of Extending the Connector

This topic describes one example of extending the connector. You might use this feature differently, depending on your business model. To extend the connector, perform the following tasks:

1 "Creating Siebel CRM Objects" on page 77

2 "Incorporating Your Extensions in Oracle Fusion Middleware" on page 80

In this example, you add the following new fields to the book appointment flow:

- Minimum Offset
- Maximum Offset
- Primary Activity

Creating Siebel CRM Objects

This task is a step in "Process of Extending the Connector" on page 77.

In this topic, you create Siebel CRM objects.

To create Siebel CRM objects

1 Open Siebel Tools.

2 Display the object types you must modify:

   a Click the View menu, and then the Options menu item.

   b In the Development Tools Options dialog box, click the Object Explorer tab.

   c In the Object Explorer Hierarchy window, make sure the Integration Object tree contains a check mark, and then click OK.

3 Create new table columns:

   a In the Object Explorer, click Table.

   b In the Tables list, query the Name property for S_EVT_ACT.

c Right-click the S_EVT_ACT table, and then choose Lock Object.

d In the Object Explorer, expand the Table tree, and then click Column.

e In the Columns list, add three new columns using values from the following table.

Name	Physical Type	Length	Type
X_MAX_OFFSET	Varchar	30	Extension
X_MIN_OFFSET	Varchar	30	Extension
X_PRIMARY_ACT_ID	Varchar	30	Extension

f In the Tables list, click Apply/DDL.

4 Create new business component fields:

a In the Object Explorer, click Business Component.

b In the Business Components list, query the Name property for Action.

c In the Object Explorer, expand the Business Component tree and then click Field.

d In the Fields list, add three new fields using values from the following table.

Name	Column	Default Value
Max Offset	X_MAX_OFFSET	00:40:00
Min Offset	X_MIN_OFFSET	00:10:00
Primary Activity Id	X_PRIMARY_ACT_ID	Field: "Id"

This configuration uses default values. Siebel CRM includes these values in the fields of each new activity record.

5 Create new integration component fields:

a In the Object Explorer, click Integration Object.

b In the Integration Object list, query the Name property for Abssiebel.

The Abssiebel integration object is predefined with this integration.

c In the Object Explorer, expand the Integration Object tree, and then click Integration Component.

d In the Integration Component list, query the Name property for Action.

e In the Object Explorer, expand the Integration Component tree, and then click Integration Component Field.

f In the Integration Component Fields list, add a new field using values from the following table.

Property	Value
Name	Max Offset
Data Type	DTYPE_TEXT
External Name	Max Offset
External Data Type	DTYPE_TEXT
External Sequence	656
XML Tag	MaxOffset

g In the Integration Component Fields list, add a new field using values from the following table.

Property	Value
Name	Min Offset
Data Type	DTYPE_TEXT
External Name	Min Offset
External Data Type	DTYPE_TEXT
External Sequence	655
XML Tag	MinOffset

h In the Integration Component Fields list, add a new field using values from the following table.

Property	Value
Name	Primary Activity Id
Data Type	DTYPE_TEXT
External Name	Primary Activity Id
External Data Type	DTYPE_TEXT
External Sequence	654
XML Tag	PrimaryActivityId

6 Compile the following projects:

- Activity
- AdvScheduler

The AdvScheduler project is predefined with this integration.

Incorporating Your Extensions in Oracle Fusion Middleware

This task is a step in "Process of Extending the Connector" on page 77.

In this topic, you incorporate your extensions in Oracle Fusion Middleware. To configure this example, you must possess detailed expertise and knowledge of using Oracle JDeveloper.

To incorporate your extensions in Oracle Fusion Middleware

1 Open the Oracle SOA Composer for your integration.

For example, open Internet Explorer, and then navigate to the following address:

`http://computer_address:soa_server_port/soa/composer`

2 At the login screen, enter your user name and password, and then click Login.

3 Enable the hook point you use in this example:

a Click Open, and then choose Open DVM.

b In the Select a DVM dialog box, click SBL_ORS_Parameter_Definition.dvm, and then click Open.

c In the SBL_ORS_Parameter_Definition.dvm screen, enable the CustomExtn_BookAppt_Hook2_Enabled parameter using values from the following table.

Parameter	Value
CustomExtn_BookAppt_Hook2_Enabled	true

For more information, see "Parameters You Must Set in the Parameter Definition File" on page 90.

4 Open Oracle JDeveloper, and then do the following work:

a Add three new fields:

- Use the same name and type that you used in Step 4 on page 78.
- Add these fields in the absSiebel.xsd input XSD. This XSD is located in the following directory:

  `oramds:/apps/SBL_ORS_INTRG/v1/XSD`

- Add these fields in the Action complexType element.

b Add the fields that you added in Step a to the mediator XSLT files:

Mediator_Input_To_BPEL_ApptBookingSys_BookAppt.xsl

This XSLT file is located in the following directory in Oracle MDS:

`oramds:/apps/SBL_ORS_INTRG/v1/XSLT`

This step configures this integration to route the changes you make in Siebel CRM through the mediator to Oracle Fusion Middleware.

c Add the extra fields that you added in Step a to hook point 2 in the following file:

CustomExtn_BookAppt_Hook2_Inpt.xsl

This step configures this integration to send information about the new fields to the hook point.

d To manipulate the transformation from Siebel CRM to Oracle Real-Time Scheduler, map the following fields in the input for hook point 2.

Siebel CRM	Oracle Real-Time Scheduler
PrimaryActivityId	PRIMARY_STOP_ID
MinOffset	MIN_OFFSET
MaxOffset	MAX_OFFSET

To avoid a conflict, make sure all names you use are unique.

5 Deploy the main composite and the hook point:

a To make sure this integration updates the Oracle Meta Data Services with your changes, use the updateMDS ant command.

For more information, see "Ant Commands You Can Use to Compile and Deploy Composites" on page 101.

b Compile and deploy your changes.

For more information, see "Compiling and Deploying Oracle Fusion Middleware Artifacts" on page 100.

6 Test your changes.

To test your extension you run the appointment booking flow. For more information, see *Siebel Field Service Integration to Oracle Real-Time Scheduler Installation and Administration Guide*.

Using Hook Point 1 and Hook Point 3

This topic describes how to use only hook point 2. To make sure the output is correct, hook point 1 and hook point 3 are disabled. If you require hook point 1 or hook point 3, then make sure you do the following work:

- Modify the XSLT transformation files for hook point 1 and hook point 3.
- Set the corresponding DVM parameters to true.

Other Examples of Extending the Connector

This topic describes other examples of extending the connector.

Transforming Fields

You can use this integration to transform fields. For example, assume you must transform the employee Start Depot Address field and End Depot Address field instead of the Employee Start Home field and End Home address field. In this situation, you must configure this integration to send the Employee Logon field and Employee Logoff field to Oracle Real-Time Scheduler.

Table 57 describes the tasks you must perform to transform fields in this example.

Table 57. Tasks You Must Perform to Transform Fields

Siebel CRM	Oracle Fusion Middleware	Oracle Real-Time Scheduler

You can do the following work:
■ Add a new field in the Employee business component. If necessary, extend the S_EMP_PER table.
■ Make sure the corresponding integration object includes the new integration component fields. For example:
  ■ StartDepotAddress
  ■ EndDepotAddress
	You can do the following work:
	■ Manually update the XSD in Oracle MDS.
	■ Update the XSLT file of the transfer mediator.
	■ Modify hook point 1 of the AdminDataManagement_EBF composite to include the extra field in the Web service request to Oracle Real-Time Scheduler.
		No configuration.

Reference Information for Extending the Connector

This topic describes reference information for extending the connector. It includes the following topics:

Predefined Hook Points You Can Use in Integration Flows

This topic describes the predefined hook points you can use in integration flows. It includes the following topics:

Hook Points That Book an Appointment

Table 58 describes predefined hook points in the AppointmentBookingSystemSBLORS_EBF flow. For more information about each flow that this table describes, see "Integration Flows That Book an Appointment" on page 18.

Table 58. Predefined Hook Points That Book an Appointment

Hook Point Name	Placement of Hook Point	How You Can Use the Hook Point
BookAppointment_Hook1	In the integration flow to book an appointment as soon as the input from Siebel CRM reaches Oracle Fusion Middleware. For more information, see Step 1 on page 20.	Manipulate the input from Siebel CRM as soon as it reaches Oracle Fusion Middleware.
BookAppointment_Hook2	In the integration flow to book an appointment before this integration calls the CHS_SLOT method. For more information, see Step 5 on page 20.	Manipulate the input to the CHS_SLOT method.
BookAppointment_Hook3	In the integration flow to book an appointment before this integration sends the output from the CHS_SLOT method to Siebel CRM. For more information, see Step 7 on page 20.	Manipulate the output from the CHS_SLOT method.
ConfirmAppointment_Hook1	In the integration flow to confirm an appointment as soon as the input from Siebel CRM reaches Oracle Fusion Middleware. For more information, see Step 1 on page 20.	Manipulate the input from Siebel CRM as soon as it reaches Oracle Fusion Middleware.

Table 58. Predefined Hook Points That Book an Appointment

Hook Point Name	Placement of Hook Point	How You Can Use the Hook Point
ConfirmAppointment_Hook2	In the integration flow to confirm an appointment before this integration calls the ASSIGN_JOB_COND method. For more information, see Step 5 on page 20.	Manipulate the input to the ASSIGN_JOB_COND method.
ConfirmAppointment_Hook3	In the integration flow to book an appointment before this integration sends the output from the ASSIGN_JOB_COND method to Siebel CRM. For more information, see Step 7 on page 20.	Manipulate the output from the CHS_SLOT method.
UpsertActivity_Hook1	In the integration flow to update and insert an appointment as soon as the input from Siebel CRM reaches Oracle Fusion Middleware. For more information, see Step 1 on page 20.	Manipulate the input from Siebel CRM as soon as it reaches Oracle Fusion Middleware.
UpsertActivity_Hook2	In the integration flow to update and insert an appointment before this integration calls the UPDATE_STOP method. For more information, see Step 5 on page 20.	Manipulate the input to the UPDATE_STOP method.
UpsertActivity_Hook3	In the integration flow to book an appointment before this integration sends the output from the UPDATE_STOP method to Siebel CRM. For more information, see Step 7 on page 20.	Manipulate the output from the UPDATE_STOP method.

Hook Points That Retrieve Activity Details from Oracle Real-Time Scheduler

Table 59 describes predefined hook points in the ActivityLatestStateRetrieval_EBF flow. For more information about each flow described in this table, see "Integration Flow That Retrieves Activity Details fromOracle Real-Time Scheduler" on page 24.

Table 59. Predefined Hook Points That Retrieve Activity Details from Oracle Real-Time Scheduler

Hook Point Name	Placement of Hook Point	How You Can Use the Hook Point
RefreshActivity_Hook1	As soon as the input from Siebel CRM reaches Oracle Fusion Middleware. For more information, see Step 1 on page 20.	Manipulate the input from Siebel CRM as soon as it reaches Oracle Fusion Middleware.
RefreshActivity_Hook2	Before this integration calls the GET_STOP method. For more information, see Step 5 on page 20	Manipulate the input to the GET_STOP method.
Refresh Activity_Hook3	Before this integration sends the output from the GET_STOP method to Siebel CRM. For more information, see Step 7 on page 20.	Manipulate the output from the GET_STOP method.

Hook Points That Synchronize a Shift, Break, or POU Status

Table 60 describes predefined hook points in the SyncBackStopShiftPOUBreakFromORS_EBF flow. The operation for hook point 1 is IntegratedShiftBreakPOUORSWS_Delete. It is an operation of the deleteShift Siebel Web service. For more information about each flow described in this table, see "Integration Flow That Synchronizes a Shift, Break, or POU Status from Siebel CRM to Oracle Real-Time Scheduler" on page 38.

Table 60. Predefined Hook Points That Synchronize a Shift, Break, or POU Status

Hook Point Name	Placement of Hook Point	How You Can Use the Hook Point
SyncBackShift_Hook1	Before this integration calls the GET_SHIFT method of the ORS Web service.	Manipulate the input to the GET_SHIFT method.
SyncBackShift_Hook2	After this integration calls the ORS_WebService Web service.	Manipulate the output from the GET_SHIFT method.
SyncBackDeleteShift_Hook1	Before this integration calls the IntegratedPOUShiftBreakORSWS_Delete method of the deleteShift Web service in the syncback shift flow.	Manipulate the input to the IntegratedPOUShiftBreakORSWS_Delete method.

Table 60. Predefined Hook Points That Synchronize a Shift, Break, or POU Status

Hook Point Name	Placement of Hook Point	How You Can Use the Hook Point
SyncBackPOU_Hook1	Before this integration calls the GET_POU method of the ORS_WebService Web service.	Manipulate the input to the GET_POU method.
SyncBackPOU_Hook2	After this integration calls the GET_POU method of the ORS_WebService Web service.	Manipulate the output to the GET_POU method.
SyncBackPOU_Hook3	Before this integration calls the POUUpsert method of the POUShiftBreakUpdate Web service.	Manipulate the input to the POUUpsert method.
SyncBackShift_Hook3	Before this integration calls the POUUpsert method of the POUShiftBreakUpdate Web service. The SyncBackShift_Hook3 hook point calls the same operation and Web service as the SyncBackPOU_Hook3 hook point.	Manipulate the input to the POUUpsert method.
SyncBackBreak_Hook	Before this integration calls the POUUpsert method of the POUShiftBreakUpdate Web service.	Manipulate the input to the POUUpsert method.
SyncBackActivity_Hook1	Before this integration calls the GET_STOP method of the ORS_WebService Web service.	Manipulate the input to the GET_STOP method.
SyncBackActivity_Hook2	After this integration receives a reply from the GET_STOP method of the ORS Web service.	Manipulate the output from the GET_STOP method.
SyncBackActivity_Hook3	Before this integration calls the ActivitySyncORSWS_Update method of the ORS Web service.	Manipulate the input to the ActivitySyncORSWS_Update method.

Hook Points That Update Employee Data

Table 61 describes predefined hook points in the AdminDataManagement_EBF flow. For more information about each flow described in this table, see "Integration Flows That Update Employee Data in Real Time" on page 44.

Table 61. Predefined Hook Points That Update Employee Data

Hook Point Name	Placement of Hook Point	How You Can Use the Hook Point
EmployeeUpsert_Hook1	In the integration flow to update employee data as soon as the input from Siebel CRM reaches Oracle Fusion Middleware. For more information, see Step 1 on page 46.	Manipulate the input from Siebel CRM as soon as it reaches Oracle Fusion Middleware.
EmployeeUpsert_Hook2	In the integration flow to update employee data before this integration calls the UPDATE_DRIVER method. For more information, see Step 5 on page 46.	Manipulate the input to the UPDATE_DRIVER method.
EmployeeUpsert_Hook3	In the integration flow to update employee data before this integration sends the output from the UPDATE_DRIVER method to Siebel CRM. For more information, see Step 6 on page 47.	Manipulate the output from the UPDATE_DRIVER method.
AddressUpsert_Hook1	In the integration flow to update an address as soon as the input from Siebel CRM reaches Oracle Fusion Middleware.	Manipulate the input from Siebel CRM as soon as it reaches Oracle Fusion Middleware.
AddressUpsert_Hook2	In the integration flow to update an address before this integration calls the UPDATE_COMMON_ADDRESS method.	Manipulate the input to the UPDATE_COMMON_ADDRESS method.
AddressUpsert_Hook3	In the integration flow to update an address before this integration sends the output from the UPDATE_COMMON_ADDRESS method to Siebel CRM.	Manipulate the output from the UPDATE_COMMON_ADDRESS method.

Hook Points That Perform a Service Region Data Transfer

Table 62 describes predefined hook points in the ServiceRegionCutoverToORS_EBF flow. For more information about each flow described in this table, see "Integration Flow That Performs a Service Region Data Transfer" on page 62. These hook points occur in a subflow in the address transfer flow. Figure 13 on page 63 does not display this subflow.

Table 62. Predefined Hook Points That Perform a Service Region Data Transfer

Hook Point Name	Placement of Hook Point	How You Can Use the Hook Point
StartAddressTransfer	Before this integration calls the UPSERT_ADDRESS method in the start address data transfer flow. For more information, see Step 6 on page 64.	Manipulate the input to the UPSERT_ADDRESS method.
EndAddressTransfer	Before this integration calls the UPSERT_ADDRESS method in the end address data transfer flow. For more information, see Step 6 on page 64.	Manipulate the input to the UPSERT_ADDRESS method.
EmployeeTransfer	Before this integration calls the UPSERT_EMPLOYEE method.	Manipulate the input to the UPSERT_EMPLOYEE method.
ActivityTransfer	Before this integration calls the Upsert_Activity method. For more information, see Step 11 on page 65.	Manipulate the input to the Upsert_Activity method.

Hook Point Operations That This Integration Calls

Table 63 lists the hook point operations that this integration call.

Table 63. Hook Point Operations That This Integration Calls

Hook Point Name	Operation That This Integration Calls
BookAppointment_Hook1	custom_extension_book
BookAppointment_Hook2	custom_extension_bookappt_hook2
BookAppointment_Hook3	custom_extension_book_hook3
ConfirmAppointment_Hook1	custom_extension_confirm
ConfirmAppointment_Hook2	custom_extension_confirmappt_hook2
ConfirmAppointment_Hook3	custom_extension_confirm_hook3
UpsertActivity_Hook1	custom_extension_upsertactivity_hook1

Table 63. Hook Point Operations That This Integration Calls

Hook Point Name	Operation That This Integration Calls
UpsertActivity_Hook2	custom_extension_upsertactivity_hook2
UpsertActivity_Hook3	custom_extension_book_hook3
EmployeeUpsert_Hook1	custom_extension_employee_upsert
EmployeeUpsert_Hook2	custom_extension_employeeupsert_hook2
EmployeeUpsert_Hook3	custom_extension_employeeupsert_hook3
AddressUpsert_Hook1	custom_extension_book
AddressUpsert_Hook2	custom_extension_addressupsert_hook2
AddressUpsert_Hook3	custom_extension_book_hook3
RefreshActivity_Hook1	custom_refresh_activitystatus_hook1
RefreshActivity_Hook2	custom_refresh_activitystate_hook2
Refresh Activity_Hook3	custom_refresh_activitystate_hook3
StartAddressTransfer_Hook	custom_serviceregncutover_addresstransfer_hook
EndAddressTransfer_Hook	custom_serviceregncutover_addresstransfer_hook
EmployeeTransfer_Hook	custom_serviceregncutover_employeetransfer_hook
ActivityTransfer_Hook	custom_serviceregncutover_activitytransfer_hook
SyncBackShift_Hook1	custom_extension_syncback_shift_hook1
SyncBackShift_Hook2	custom_extn_syncback_shift_hook2
SyncBackDeleteShift_Hook1	custom_extn_shiftdelete_hook1
SyncBackPOU_Hook1	custom_extn_syncback_pou_hook1
SyncBackPOU_Hook2	custom_extn_syncbackpou_hook2
SyncBackPOU_Hook3	custom_extn_syncbackpou_hook3
SyncBackShift_Hook3	custom_extn_syncbackshift_hook3
SyncBackBreak_Hook	custom_extn_syncback_break_hook1
SyncBackActivity_Hook1	custom_extn_syncback_activity_hook1
SyncBackActivity_Hook2	custom_extn_syncback_activity_hook2
SyncBackActivity_Hook3	custom_extn_syncback_activity_hook3

Parameters You Must Set in the Parameter Definition File

To use a hook point, you must set the enable parameter for the hook point to true. For example, to use the BookAppointment_Hook1 hook point, you must set the following DVM parameter in the SBL_ORS_Parameter_Definition.dvm file to true:

CustomExtn_BookAppt_Hook1_Enabled

Table 64 describes the parameters you must set in the SBL_ORS_Parameter_Definition.dvm file to use a hook point. For an example of setting an enable parameter, see Step 3 on page 80.

Table 64. Parameters You Must Set in the Parameter Definition File

Hook Point Name	Parameter
BookAppointment_Hook1	CustomExtn_BookAppt_Hook1_Enabled
BookAppointment_Hook2	CustomExtn_BookAppt_Hook2_Enabled
BookAppointment_Hook3	CustomExtn_BookAppt_Hook3_Enabled
ConfirmAppointment_Hook1	CustomExtn_ConfirmAppt_Hook1_Enabled
ConfirmAppointment_Hook2	CustomExtn_ConfirmAppt_Hook2_Enabled
ConfirmAppointment_Hook3	CustomExtn_ConfirmAppt_Hook3_Enabled
UpsertActivity_Hook1	CustomExtn_UpsertActivity_Hook1_Enabled
UpsertActivity_Hook2	CustomExtn_UpsertActivity_Hook2_Enabled
UpsertActivity_Hook3	CustomExtn_UpsertActivity_Hook3_Enabled
EmployeeUpsert_Hook1	CustomExtn_EmployeeUpsert_Hook1_Enabled
EmployeeUpsert_Hook2	CustomExtn_EmployeeUpsert_Hook2_Enabled
EmployeeUpsert_Hook3	CustomExtn_EmployeeUpsert_Hook3_Enabled
AddressUpsert_Hook1	CustomExtn_AddressUpsert_Hook1_Enabled
AddressUpsert_Hook2	CustomExtn_AddressUpsert_Hook2_Enabled
AddressUpsert_Hook3	CustomExtn_AddressUpsert_Hook3_Enabled
RefreshActivity_Hook1	CustomExtn_RefreshActivity_Hook1_Enabled
RefreshActivity_Hook2	CustomExtn_RefreshActivity_Hook2_Enabled
Refresh Activity_Hook3	CustomExtn_RefreshActivity_Hook3_Enabled
StartAddressTransfer_Hook	CustomExtn_ServiceRegnCutover_AddressTranfer_Hook_Enabled
EndAddressTransfer_Hook	CustomExtn_ServiceRegnCutover_EndAddressTansfer_Hook_Enabled
EmployeeTransfer_Hook	CustomExtn_ServiceRegnCutover_EmployeeTransfer_Hook_Enabled

Table 64. Parameters You Must Set in the Parameter Definition File

Hook Point Name	Parameter
ActivityTransfer_Hook	CustomExtn_ServiceRegnCutover_ActivityTransfer_Hook_Enabled
SyncBackShift_Hook1	CustomExtn_SyncBackActivity_Hook1_Enabled
SyncBackShift_Hook2	CustomExtn_SyncBackActivity_Hook2_Enabled
SyncBackDeleteShift_Hook1	CustomExtn_SyncShiftDelete_Hook_Enabled
SyncBackPOU_Hook1	CustomExtn_SyncBackPOU_Hook1_Enabled
SyncBackPOU_Hook2	CustomExtn_SyncBackPOU_Hook2_Enabled
SyncBackPOU_Hook3	CustomExtn_SyncBackPOU_Hook3_Enabled
SyncBackShift_Hook3	CustomExtn_SyncBackShift_Hook3_Enabled
SyncBackBreak_Hook	CustomExtn_SyncBackBreak_Hook_Enabled
SyncBackActivity_Hook1	CustomExtn_SyncBackActivity_Hook1_Enabled
SyncBackActivity_Hook2	CustomExtn_SyncBackActivity_Hook2_Enabled
SyncBackActivity_Hook3	CustomExtn_SyncBackActivity_Hook3_Enabled

11 Installing SOA Composites and Configuring This Integration

This chapter describes how to install Service-Oriented Architecture (SOA) composites for Siebel Field Service Integration to Oracle Real-Time Scheduler. It includes the following topics:

Process of Installing SOA Composites for This Integration

To install SOA composites for this integration, do the following:

Preparing to Install SOA Composites

This task is a step in "Process of Installing SOA Composites for This Integration" on page 93.

You must first make sure your environment is prepared to install SOA composites for this integration.

To prepare to install SOA composites

1. Make sure the following products or product suites are installed:
   - Oracle Fusion Middleware, version 11g, including Oracle SOA Suite PS4 and Oracle WebLogic Server.

     For more information, see "Documentation Sources for Configuring Oracle Real-Time Scheduler" on page 15.

- Oracle Database versions that Siebel Business Applications and Oracle Real-Time Scheduler support.

  For more information, see the appropriate database installation documentation on Oracle Technology Network.

- Siebel Business Applications, including Siebel Service. For more information, see the following resources:
  - Books that support your Siebel application. These books reside on the *Siebel Bookshelf* on Oracle Technology Network.
  - Installation documentation for all applicable patch releases on My Oracle Support.
- Oracle Real-Time Scheduler, version 2.1.

  For more information, see "Documentation Sources for Configuring Oracle Real-Time Scheduler" on page 15.

2. Make sure Oracle WebLogic Server is installed and running.
3. Make sure the following items are installed and running:
   - Oracle SOA Server
   - (Optional) Oracle Business Activity Monitoring (BAM) Server

Configuring Data Sources on the Oracle WebLogic Server

You use the Oracle WebLogic Server to create the Java Database Connectivity (JDBC) data source for the Siebel database. To query the Siebel database in some of the integration processes, the DbAdapter server component uses this data source. Example processes include the service region migration to Oracle Real-Time Scheduler and retrieving geocodes for multiple addresses. SOA provides the DbAdapter server component as a way to connect to the Siebel database.

To configure the data sources on the Oracle WebLogic Server Console

1. Log in to the Oracle WebLogic server console.
2. Navigate to Services, JDBC, and then Data Sources.
3. To create a new data source, click New.
4. Provide the following details:
   - Data Source Name
   - JNDI Name
   - Database Type
   - Database Driver
5. Click Next.

6 Make sure the Support Global Transactions option does not contain a check mark.

7 Click Next, and then specify the following details for the Siebel data source:

- Database Name
- Host Name
- Port
- Database Table Owner User Name and Password

8 Click Next, and then click Next again.

9 Choose the relevant target servers to which you are deploying.

For example, the administration server, BAM server, and the SOA server.

10 Click Finish.

11 Navigate to Services, JDBC, and then Data Sources.

12 Choose the data source you created in Step 3.

13 Click the Connection Pool tab.

14 Change the maximum capacity from 15 to 100.

Configuring the Database Adapter on the Oracle WebLogic Server

This task is a step in "Process of Installing SOA Composites for This Integration" on page 93.

To configure the database adapter on the Oracle WebLogic Server

1 In the Oracle WebLogic Server Console, open the Application Server Navigator window.

2 Create an outbound connection pool:

a Click Deployments under Domain Structure, and then click DbAdapter.

b Navigate to the Configuration Tab, and then Outbound Connection Pool.

c Click New.

d Click Connection Factory.

e Click Next.

f Specify the JNDI name as `eis/DB/Siebel_DB`.

g Click Finish.

The Oracle WebLogic server console creates an outbound connection pool for the connection group.

h Save the plan into the Plan.xml file.

It is strongly recommended that you store the Plan.xml file in a separate folder. For example, create a folder named DBPlan in the *MiddlewareHome*/*OracleSOAHome*/soa folder. Using this technique allows multiple files named Plan.xml to exist. For example, one file for JMS, one file for File, and one file for BAM.

3 Configure the database adapter:

a Navigate to Deployments.

b Click DbAdapter.

c Navigate to Configuration, and then Outbound Connection Pool.

d Expand the Connection Factory tree.

e Click the Java Naming and Directory Interface (JNDI) name of the connection pool you created in Step 2.

f Navigate to Properties.

g In the Property Value for the dataSourceName property, enter the JNDI name of the Siebel data source.

h Save your changes and then update DbAdapter.

4 Restart the Oracle WebLogic Server, the BAM server, and the SOA server.

Configuring the BPEL Timeout Setting on the Oracle WebLogic Server

This task is a step in "Process of Installing SOA Composites for This Integration" on page 93.

In this topic, you set a BPEL timeout that supports this integration.

To configure the BPEL timeout setting on the Oracle WebLogic Server

1 In the Oracle WebLogic Server Console, navigate to Deployments.

2 Expand the soa-infra tree, and then expand EJBs.

3 Click BPELEngineBean.

4 Click the Configuration tab.

5 Change the transaction timeout from 30 to 3000 or higher.

6 Click Save.

Adding User Credentials to the Credential Store

This task is a step in "Process of Installing SOA Composites for This Integration" on page 93.

A *credential store* is a repository that includes security data. It can include user name and password combinations or the public key certificates that allow Oracle Fusion Middleware to access an external system, such as Siebel CRM.

Some integration flows call Siebel Inbound Web Services, which are secured Web services. You must configure this integration so that it can authenticate with these services. Before calling them, the integration flow extracts credentials from the credential store and then adds them to the SOAP header in the request.

To add Siebel user credentials to the credential store

1. Log in to Oracle Enterprise Manager.
2. Choose Farm from the drop-down list.
3. Expand the Farm_soa_doman tree, expand the WebLogic Domain tree, and then click soa_domain.
4. Right-click soa_domain, choose Security, and then choose the Credentials menu item.

   Oracle Enterprise Manager displays the Credentials page.
5. In the Credentials page, create a credential map:
   a. In the Credential Store Provider section, click Create Map.
   b. In the Create Map dialog box, enter the following map name:

      `oracle.wsm.security`

   c. Click OK.

      Oracle Enterprise Manager displays the new credential map name. It includes a map icon.
6. Add a key to the credential map that you created in Step 5:
   a. Click Create Key.
   b. In the Create Key dialog box, enter values using information from the following table.

Field	Description
Select Map	Choose oracle.wsm.security.
Key	Enter `SBL_WS_USER`.
Type	Choose Password.
User Name	Enter the name of the Siebel user who is authorized to call a Siebel Web service. For example, `SADMIN`.
Password	Enter the password of the Siebel user who is authorized to call a Siebel Web service.

   c. Click OK.

      Oracle Enterprise Manager displays the new key under the map icon that identifies the map you chose. For more information, see "Administering a Credential Key" on page 99.

7 Add another key to the credential map that you created in Step 5:

a Click Create Key.

b In the Create Key dialog box, enter values using information from the following table.

Field	Description
Select Map	Choose oracle.wsm.security.
Key	Enter mwm.key.
Type	Choose Password.
User Name	Enter the name of the Oracle Real-Time Scheduler user who is authorized to call an Oracle Real-Time Scheduler Web service, for example, SYSUSER.
Password	Enter the password of the Oracle Real-Time Scheduler user who is authorized to call a Siebel Web service.

c Click OK.

Oracle Enterprise Manager displays the new key under the map icon that identifies the map you chose. For more information, see "Administering a Credential Key" on page 99.

8 On the Oracle Fusion Middleware Server, navigate to the Oracle Fusion Middleware configuration directory.

For example:

```
home/eng-sm/unixuser/Oracle/Middleware_11gPS1/user_projects/domains/soa_domain/config/fmwconfig
```

9 Use an XML editor to open the following file:

```
system-jazn-data.xml
```

10 Add the following code to the system-jazn-data.xml file:

```
<grant>
            <grantee>
                <principals>
                    <principal>

<class>oracle.security.jps.internal.core.principals.JpsAnonymousUserImpl</class>
                        <name>anonymous</name>
                    </principal>
                    <principal>

<class>oracle.security.jps.internal.core.principals.JpsAnonymousUserImpl</class>
```

```
                        <name>anonymous-role</name>
                    </principal>
                </principals>
            </grantee>
            <permissions>
                <permission>

<class>oracle.security.jps.service.credstore.CredentialAccessPermission</class>

<name>context=SYSTEM,mapName=oracle.wsm.security,keyName=SBL_WS_USER</name>
                    <actions>read</actions>
                </permission>
            </permissions>
            <permission-set-refs>
            </permission-set-refs>
        </grant>
```

11 Save the system-jazn-data.xml file.

12 Restart the Oracle Fusion Middleware Server.

Administering a Credential Key

You can administer a credential key.

To administer a credential key

1 In Oracle Enterprise Manager, navigate to the Credentials page.

2 Do any of the following:

- To remove a key or map, click Delete. If you delete a credential map, then Oracle Enterprise Manager deletes all keys in the map.
- To view or modify the data in a key or map, click Edit.
- To display credentials for a key, enter the key name in the Credential Key Name window and then click the arrow.
- To refresh the list of credentials after you examine the results of a query, repeat Step 4 on page 97.

Compiling and Deploying Oracle Fusion Middleware Artifacts

This task is a step in "Process of Installing SOA Composites for This Integration" on page 93.

This topic describes how to run scripts that compile and deploy Oracle Fusion Middleware artifacts for this integration.

To compile and deploy Oracle Fusion Middleware artifacts

1 Prepare to run the installation scripts:

   a Install Apache Ant on the computer that you use to run the installation scripts.

   Apache Ant is a tool you can use to deploy composites on the Oracle Web Logic SOA Server. For more information, see http://ant.apache.org.

   b Make sure the PATH environment variable references the binary directory that Ant uses.

   c Unzip the SBL_ORS_INTRGv2.zip file.

   This file is packaged as the Siebel-ORS Integration Connector. Make sure you use the same directory structure that is defined in the zip file.

   d Navigate to the `SBL_ORS_INTRGv2\build` folder.

   e Open the build.properties file, and then update the following parameters using values from the following table.

   The build.properties file contains the properties for the Ant build. Note that all paths in the build.properties file must use the Unix forward slash (/) convention. For example, `D:/home/bin`.

Parameter	Description
weblogic_admin_server_host	Set to the host name for the Oracle WebLogic Server.
weblogic_admin_server_port	Set to the port of the WebLogic Administration Server.
weblogic_soa_mgdserver_name	Set to the SOA Server name.
weblogic_soa_server_hostname	Set to the SOA Server host name.
weblogic_soa_server_portnumber	Set to the SOA Server port number.
weblogic_username	Set to the WebLogic User Name that Oracle WebLogic uses for authentication.
weblogic_password	Set to the WebLogic Password for the authorized user.
siebel.version	Set to 81, which is the Siebel Version.
siebel_server_host	Set to the host address of the server where the Siebel Server is installed.

Parameter	Description
java.passed.home	Set to the location of the JDK you will use.
src.location	Set to the location of the integration code. This location is the base directory where you extract the zip file. For example, C:SBL_ORS_INTEGRATION.
oracle_home	Set to the location of the Oracle Home directory.
mw_home	Set to the middleware installation base directory.

2 Open a command prompt, and then navigate to the SBL_ORS_INTRGv2\build folder.

3 At the command prompt, type deployAll, and then press ENTER.

This command deploys all the Oracle Fusion Middleware composites and Oracle Metadata Service artifacts. To deploy the composites but not the Oracle Metadata Service artifacts, type ant deployAllComposites at the command prompt.

To reverse a deployment, enter ant undeployAll. This command reverses the deployment for all composites and for all Oracle MDS artifacts. For more information, see "Ant Commands You Can Use to Compile and Deploy Composites" on page 101.

Ant Commands You Can Use to Compile and Deploy Composites

Table 65 lists the ant commands that you can use to compile and deploy composites.

Table 65. Ant Commands You Can Use to Compile and Deploy Composites

Command	Descriptions
ant all	Compiles and deploys all composites including Oracle Meta Data Services.

ant deployAll	Deploys all composites including Oracle Meta Data Services.
	
	Deploys custom composites according to the following value in the deployCustomExtensionsAlso parameter in the properties file:
	
	■ **Y.** Deploys the following composites:
	  ■ All composites that are listed in the customExtensions parameter
	  ■ Integration composites
	■ **N.** Deploys only predefined integration composites. Does not deploy custom composites.

ant deployAllComposites	Deploys the same items that the deployAll command compiles and deploys except does not compile or deploy Oracle Meta Data Services.

Table 65. Ant Commands You Can Use to Compile and Deploy Composites

Command	Descriptions
`ant updateMDS`	Deploys only the Oracle Meta Data Services artifacts. Does not deploy any composites.
`ant deployComposite`	Deploys a specific composite according to the following information that you provide: ■ Composite name ■ Location of where the composite resides ■ Full path of the deployment configuration plan
`ant deployCompositeWithoutConfigPlan`	Same as the deployComposite command but you do not provide the path to the deployment configuration plan.
`ant compileComposite`	Compiles a specific composite according to the following information that you provide: ■ Composite name ■ Location of where the composite resides
`ant compileAndDeployComposite`	Compiles and deploys a specific composite according to the following information that you provide: ■ Composite name ■ Location of where the composite resides ■ Full path to the deployment configuration plan
`ant compileAllCustomComposites`	Compiles all custom composites according to the composites listed in the properties file.
`ant deployAllCustomComposites`	Deploys all custom composites according to the composites listed in the properties file.
`ant compileAndDeployAllCustomComposites`	Compiles and deploys all custom composites according to the composites listed in the properties file.
`ant undeployAll`	Undeploys all composites including Oracle Meta Data Services.
`ant undeployAllComposites`	Undeploys all composites. Does not undeploy Oracle Meta Data Services.
`ant undeployMDS`	Undeploys only Oracle Meta Data Services.
`ant undeployComposite`	Undeploys a specific composite according to the composite name that you provide.

Oracle Fusion Middleware Projects

Table 66 lists the Oracle Fusion Middleware projects this integration uses.

Table 66. Oracle Fusion Middleware Projects

Projects	Descriptions
ActivityLatestStateRetrieval_EBF	Retrieves the latest status of the activity from Oracle Real-Time Scheduler to Siebel CRM.
AdminDataManagment_EBF	Provides real-time synchronization of administrative data, including the transfer of employee data records, employee leaves, employee skills, address records, holidays, and so forth.
AppointmentBookingSystemSBLORS_EBF	Used by the Appointment Booking System. It includes the following functionality: ■ Book Appointment ■ Confirm Appointment ■ Cancel Appointment ■ Contract Scheduling ■ Rescheduling ■ Booking Beyond Scheduling Horizon ■ Insert Activity ■ Activity Status Updates
BatchGeoCode_EBF	Uses the Geocode Server of the Oracle Real-Time Scheduler to retrieve geocodes for multiple addresses. It does the following: ■ If Oracle Real-Time Scheduler returns a geocode, and if Oracle Real-Time Scheduler does not return multiple geocodes, then the BatchGeoCode_EBF project writes the geocode directly back to the Siebel database. ■ If Oracle Real-Time Scheduler does not return a geocode, and if Oracle Real-Time Scheduler returns multiple geocodes, then the BatchGeoCode_EBF project writes an error in the Siebel database.

Table 66. Oracle Fusion Middleware Projects

Projects	Descriptions
RealTimeSingleAddressGeoCoding	Allows the Geocode Server of the Oracle Real-Time Scheduler to retrieve a geocode for a single address from Siebel CRM. If Oracle Real-Time Scheduler returns multiple geocodes, then the project displays these geocodes. The RealTimeSingleAddressGeoCoding flow does not directly update the Siebel database.
ServiceRegionRollbackFromORS_EBF	Reverts back scheduling to Siebel Scheduler for a service region that is scheduled in Oracle Real-Time Scheduler.
ServiceRegionCutoverToORS_EBF	Transfers a Service Region along with Employees and Activities to be scheduled inOracle Real-Time Scheduler.
ShiftWeeklyTemplateMaintenanceSBLORS	Provides real-time synchronization of Schedule data including Schedule Hours.
StatusUpdateSBLtoORS_EBF	Synchronizes the activity status from Siebel CRM to Oracle Real-Time Scheduler.
SyncbackDispatcherUpdates_EBF	Provides synchronization of Activity Status updates from Oracle Real-Time Scheduler to Siebel CRM.
SyncbackShiftBreakPOUDispatcherUpdates_EBF	Provides synchronization of Shift, POU, and Break updates from Oracle Real-Time Scheduler to Siebel CRM.
SyncbackInactiveDispatcherUpdates_EBF	Provides synchronization of Cancelled or Inactive Activity Status updates from Oracle Real-Time Scheduler to Siebel CRM.
WorkCalendarMaintenanceSBLORS_EBF:	Provides real-time synchronization of Schedule Exception data including Schedule Exception Hours.

Projects That the Installation Scripts Deploy

The installation scripts deploy the following projects:

- ActivityLatestStateRetrieval_EBF
- AdminDataManagment_EBF
- AppointmentBookingSystemSBLORS_EBF
- BatchGeoCode_EBF
- RealtimeSingleAddressGeocoding

- ServiceRegionRollbackFromORS_EBF
- StatusUpdateSBLtoORS_EBF
- ServiceRegionCutoverToORS_EBF
- ShiftWeeklyTemplateMaintenanceSBLORS_EBF
- SyncBackDispatcherUpdates_EBF
- SyncBackInactivesDispatcherUpdates_EBF
- SyncbackShiftBreakPOUDispatcherUpdates_EBF
- WorkCalendarMaintenanceSBLORS_EBF

Configuring Domain Value Maps

This task is a step in "Process of Installing SOA Composites for This Integration" on page 93.

This topic includes the following topics:

A domain value map operates on data values. These values transit through the infrastructure at run time. A DVM allows you to map data values that reside in a vocabulary in a given domain to a vocabulary that resides in a different domain.

For example, assume a domain represents a city with a long name, such as Los Angeles, while another domain represents a city with the short name for that same city, such as LA. You can use a domain value map to map Los Angeles to LA. A *point-to-point mapping* is a direct mapping of values between two or more domains. Mapping Los Angeles to LA is an example of a point-to-point mapping.

Mapping an Activity Status From Siebel CRM to Oracle Real-Time Scheduler

This topic describes how to edit the SBL_to_ORS_ActivityStatusMapping DVM. You can create and configure many different status values for activities in Siebel CRM. However, Oracle Real-Time Scheduler includes a fixed set of predefined status values, each of which Oracle Real-Time Scheduler designs for a particular stage in routing. You must map statuses from the Siebel environment to the corresponding statuses from Oracle Real-Time Scheduler in the middle tier. You can use DVMs to modify and commit changes without having to stop Oracle Fusion Middleware. You use an activity status mapping with status update flows. A field service technician can use Siebel Wireless application or Siebel Mobile connected application to change the status of an activity. During an update, this integration uses the appropriate status mapping in the DVM to map the change that the field service technician makes, and then calls Oracle Real-Time Scheduler.

Columns You Use to Map an Activity Status

To map an activity status, you use the following columns:

- Column 1: Siebel
- Column 2: ORS

Requirements for Mapping an Activity Status

If you use the SBL_to_ORS_ActivityStatusMapping DVM, then you must make sure your mapping meets the following requirements:

- **The data in the Siebel column must be unique.** Multiple Siebel statuses can map to the same status in Oracle Real-Time Scheduler. However, there must not be more than one entry in the map with the same Siebel status.
- **The status is case-sensitive.** If the Siebel status is ABC, then a DVM row must exist that includes a Siebel column value of ABC. This integration treats other values as different statuses. Example values include ABC, abc, or any other combination. The same rule applies for Oracle Real-Time Scheduler. Because all valid Oracle Real-Time Scheduler statuses are in upper case, the ORS column must use uppercase and valid status values. This integration treats a case mismatch between a domain value map (DVM) column and the data that the column corresponds to as a status mismatch. This situation can result in a rejected message. For lists of valid activity statuses, see the “Documentation Sources for Configuring Oracle Real-Time Scheduler” on page 15.

DVM Entries That Map an Activity Status

Table 67 lists the entries that are predefined in the SBL_to_ORS_ActivityStatusMapping DVM. This DVM must include all Siebel status values that a field service technician can send.

Table 67. Entries That Are Predefined in the SBL_to_ORS_ActivityStatusMapping DVM

Siebel Column	ORS Column
Schedule	M1SH
Dispatched	M1DS
Cancelled	M1CA
Declined	M1DC
Acknowledge	DISPATCHED
Enroute	M1EN
In Progress	M1ON
Done	M1CO

Mapping an Activity Status From Oracle Real-Time Scheduler to Siebel CRM

This Oracle Real-Time Scheduler integration uses the ORS_to_Siebel_ActivityStatusMapping DVM while it synchronizes an activity status for an activity from Oracle Real-Time Scheduler to Siebel CRM. This integration maps an activity status from Oracle Real-Time Scheduler to Siebel CRM in the following situations:

- If the call center agent clicks Refresh Activity to get the latest appointment details
- If the activity status is updated to Despatched or Inactive in Oracle Real-Time Scheduler

Columns You Use to Map an Activity Status

To map a stop status, you use the following columns:

- Column 1: ORS
- Column 2: Siebel

Requirements for Mapping an Activity Status

If you use the ORS_to_Siebel_ActivityStatusMapping DVM, then you must make sure your mapping meets the following requirements:

- The data in the ORS column must be unique. Multiple Oracle Real-Time Scheduler statuses can map to the same status in Siebel CRM. However, there must not be more than one entry in the map that contains the same status value in the ORS column.

- The requirements to map a stop status are the same as the requirements to map an activity status.

DVM Entries That Map an Activity Status

Table 68 lists the entries that are predefined in the ORS_to_Siebel_ActivityStatusMapping DVM. All possible activity status values must be present. For the list of valid statuses for an activity status, see "Documentation Sources for Configuring Oracle Real-Time Scheduler" on page 15.

Table 68. Entries That Are Predefined in the ORS_to_Siebel_ActivityStatusMapping DVM

ORS Column	Siebel Column
M1SH	Schedule
M1DS	Dispatched
M1CA	Cancelled
M1DC	Declined
M1EN	Enroute
M1CO	Done
M1ON	In Progress

Mapping a Shift Status From Siebel CRM to Oracle Real-Time Scheduler

This Oracle Real-Time Scheduler integration uses the Siebel_to_ORS_ShiftStatusMapping DVM while it synchronizes a Shift status from Siebel CRM to Oracle Real-Time Scheduler. This Oracle Real-Time Scheduler integration maps a Shift status from Siebel CRM to Oracle Real-Time Scheduler if the Siebel Field Technician updates the status of the Shift from Siebel Wireless or Siebel Mobile connected application.

Requirements for Mapping a Shift Status

If you use the Siebel_to_ORS_ShiftStatusMapping DVM, then you must make sure your mapping meets the following requirements:

- The data in the Siebel column must be unique. Multiple Oracle Real-Time Scheduler statuses can map to the same status in Siebel CRM. However, there must not be more than one entry in the map that contains the same status value in the Siebel column.

DVM Entries That Map a Shift Status

Table 69 lists the entries that are predefined in the ORS_to_Siebel_ShiftStatusMapping DVM. All possible stop status values must be present. For the list of valid statuses for an activity status, see "Documentation Sources for Configuring Oracle Real-Time Scheduler" on page 15.

Table 69. Entries That Are Predefined in the ORS_to_Siebel_ShiftStatusMapping DVM

Siebel Column	ORS Column
Start	M1ST
Complete	M1CO

Mapping a Break Status from Siebel CRM to Oracle Real-Time Scheduler

This integration uses the Siebel_to_ORS_BreakStatusMapping DVM while it synchronizes a Break status from Siebel CRM to Oracle Real-Time Scheduler. This integration maps a Break status from Siebel CRM to Oracle Real-Time Scheduler if Siebel Field Technician updates the status of the Break from Siebel Wireless or Siebel Mobile connected application.

Requirements for Mapping an Activity Status

If you use the Siebel_to_ORS_BreakStatusMapping DVM, then you must make sure your mapping meets the following requirements:

- The data in the Siebel column must be unique. Multiple Oracle Real-Time Scheduler statuses can map to the same status in Siebel CRM. However, there must not be more than one entry in the map that contains the same status value in the Siebel column.

DVM Entries That Map a Break Status

Table 70 lists the entries that are predefined in the Siebel_to_ORS_BreakStatusMapping DVM. All possible stop status values must be present. For the list of valid statuses for a break status, see "Documentation Sources for Configuring Oracle Real-Time Scheduler" on page 15.

Table 70. Entries That Are Predefined in the Siebel_to_ORS_BreakStatusMapping DVM

Siebel Column	ORS Column
Start	M1ST
Complete	M1CO

Mapping a Time Zone from Siebel CRM to Oracle Real-Time Scheduler

This integration uses the SBL_ORS_TimeZoneMapping DVM to map the Siebel's Time Zone abbreviation as well as Time Zone Name to Oracle Real-Time Scheduler's Time Zone code. This mapping is being used while transferring Service Region and Employee from Siebel CRM and also during Appointment Booking in the Appointment Booking System.

Requirements for Mapping a Time Zone

If you use the SBL_ORS_TimeZoneMapping DVM, then you must make sure your mapping meets the following requirements:

- The Standard Abbreviation of the Time Zone defined at Siebel CRM should be mapped to corresponding Time Zone Code at Oracle Real-Time Scheduler.
- The Name of the Time Zone defined at Siebel CRM should be mapped to corresponding Time Zone Code at Oracle Real-Time Scheduler.

DVM Entries That Map a Time Zone

Table 71 lists the entries that are predefined in the SBL_ORS_TimeZoneMapping DVM. All possible Time Zone values must be present. For the list of Timezone codes in Oracle Real-Time Scheduler, see "Documentation Sources for Configuring Oracle Real-Time Scheduler" on page 15.

Table 71. Entries That Are Predefined in the SBL_ORS_TimeZoneMapping DVM

Siebel Column	ORS Column
Pacific Standard Time	PST001
(GMT-08:00) Pacific Standard Time (US & Canada)	PST001

Mapping a Time Zone from Oracle Real-Time Scheduler To Siebel CRM

This integration uses the ORS_SBL_TimeZoneMapping DVM to map the Oracle Real-Time Scheduler's Time Zone code to Siebel's Time Zone abbreviation. This mapping is being used while refreshing the activity.

Requirements for Mapping a Time Zone

If you use the ORS_SBL_TimeZoneMapping DVM, then you must make sure your mapping meets the following requirements:

- The Standard Abbreviation of the Time Zone defined at Siebel CRM should be mapped to corresponding Time Zone Code in Oracle Real-Time Scheduler.

DVM Entries That Map a Time Zone

Table 72 lists the entries that are predefined in the ORS_SBL_TimeZoneMapping DVM. All possible stop status values must be present. For the list of Timezone codes in Oracle Real-Time Scheduler, see "Documentation Sources for Configuring Oracle Real-Time Scheduler" on page 15.

Table 72. Entries That Are Predefined in the ORS_SBL_TimeZoneMapping DVM

ORS Column	Siebel Column
PST001	Pacific Standard Time

Mapping Schedule Weekdays from Siebel CRM to Oracle Real-Time Scheduler

This integration uses the SBL_ORS_ScheduleWeekdays DVM to map the Siebel's Schedule Weekdays to Oracle Real-Time Scheduler's Weekday code. This mapping is being used while transferring the Work Schedule from Siebel CRM.

Requirements for Mapping Schedule Weekdays

If you use the SBL_ORS_ScheduleWeekdays DVM, then you must make sure your mapping meets the following requirements:

- The Language Independent Code of Schedule Weekday List of value defined at Siebel CRM should be mapped to corresponding Weekday Code in Oracle Real-Time Scheduler.

DVM Entries That Map a Weekday Schedule

Table 73 lists the entries that are predefined in the SBL_ORS_ScheduleWeekdays DVM. All possible stop status values must be present. For the list of valid Code for Weekdays in Oracle Real-Time Scheduler, see "Documentation Sources for Configuring Oracle Real-Time Scheduler" on page 15.

Table 73. Entries That Are Predefined in the SBL_ORS_ScheduleWeekdays DVM

Siebel Column	ORS Column
Monday	M101
Tuesday	M102
Wednesday	M103
Thursday	M104
Friday	M105
Saturday	M106
Sunday	M107

Mapping Language Code of Siebel CRM to Oracle Real-Time Scheduler

This integration uses the SBL_ORS_LanguageCodeMapping DVM to map the Siebel's Language Code to Oracle Real-Time Scheduler's Language Code. This mapping is being used to support Multilingual deployment in Siebel CRM and Oracle Real-Time Scheduler.

Requirements for Mapping a Language Code

If you use the SBL_ORS_LanguageCodeMapping DVM, then you must make sure your mapping meets the following requirements:

- The Language Code defined at Siebel CRM should be mapped to corresponding Language Code at Oracle Real-Time Scheduler.

Examples of Mapping a Language Code

Table 74 includes a correct example of mapping a Language Code.

Table 74. Correct Example of Mapping a Language Code

Siebel Column	ORS Column
ENU	ENG
ESN	ESA

Mapping End Point URLs for all Oracle Real-Time Scheduler and Siebel Web Services

This integration uses the SBL_ORS_Endpoint_Definition DVM to map the end point URL for each Inbound Web Service exposed by Oracle Real-Time Scheduler and Siebel CRM.

Columns You Use to Map an Activity Status

To map a Activity status, you use the following columns:

- Column 1: Service Name
- Column 2: Endpoint Url

Requirements for Mapping an End Point URL

If you use the SBL_ORS_Endpoint_Definition DVM, then you must make sure your mapping meets the following requirements:

- Update Endpoint URL for each of the Service Name accordingly based on physical machine name.

Examples of Mapping an End Point URL

Table 75 includes a correct example of mapping an End Point URL.

Table 75. Correct Example of Mapping an End Point URL

Service Name	End Point URL
Employee Maintenance	http://<ORSserver name>:<port>/ouaf/XAIApp/xaiserver/M1-EmployeeMaintenance
ShiftExceptionORSWebService	http://<Siebelserver name>:<port>/eai_<lang>/start.swe?SWEExtSource=WebService&SWEExtCmd=Execute&WSSOAP=1

Mapping Country Code of Siebel CRM to Oracle Real-Time Scheduler

This integration uses the SBL_ORS_CountryCodes DVM to map the Siebel's Country Code to Oracle Real-Time Scheduler's Country Code. This integration Country code on Siebel to the corresponding Country code defined at Oracle Real-Time Scheduler. The mapping is being used while sending Geocoding request for an Address from Siebel CRM to Oracle Real-Time Scheduler.

Requirements for Mapping a Country Code

If you use the SBL_ORS_LanguageCodeMapping DVM, then you must make sure your mapping meets the following requirements:

- The Language Independent Code for Country List of value defined in Siebel CRM should be mapped to corresponding Country Code in Oracle Real-Time Scheduler.

Examples of Mapping a Country Code

Table 76 includes a correct example of mapping a Country Code.

Table 76. Correct Example of Mapping a Country Code

Siebel Column	ORS Column
USA	USA
France	FRA

Mapping Oracle Real-Time Scheduler Expertise to Siebel CRM Skills

The SBL_Expertise_Code_Definition DVM stores the proficiency values that a skill supports. You must map any value that you define in the SBL_Expertise_Code_Definition DVM to the EXPERTISE_CD list of values in Siebel CRM.

Columns You Use to Map Skills

To map a skill, you use the following columns:

- Column 1: Expertise Code
- Column 2: Relative Value

Relative Values Determine Expertise Ranking

The relative value determines expertise ranking. For example, if an Expert must possess a higher proficiency than a Novice, then Expert must include a higher integer value than Novice.

DVM Entries That Map a Skill

Table 77 lists entries that are predefined in the SBL_Expertise_Code_Definition DVM. One record must hold the Expertise Code as Default_Expertise. The Relative Value for this should be either NOVICE, INTERMEDIATE, or EXPERT. Another record must hold the Expertise Level as Default_Expertise_Level.The Relative Value for this should be any expertise level other than 0. Value 0 will be considered as no skill level and should not be used.

Table 77. Entries That Are Predefined in the SBL_Expertise_Code_Definition DVM

Expertise Code	Relative Value
Default_Expertise	Novice
Default_Expertise_Level	1
Novice	1
Intermediate	2
Expert	3

Examples of Mapping a Skill

Table 78 includes a correct example of mapping a skill.

Table 78. Correct Example of Mapping a Skill

Expertise Code	Relative Value
Default_Expertise	Novice
Default_Expertise_Level	1
Novice	1
Intermediate	2
Expert	3

Incorrect Example 1

Table 79 includes an incorrect example of mapping a skill. The required record for Default_Expertise is missing. Table 77 includes this record.

Table 79. Incorrect Example of Mapping a Skill

Expertise Code	Relative Value
Novice	1
Intermediate	2
Expert	3

Mapping Run-Time Parameters

This topic describes how to edit the SBL_ORS_Parameter_Definition DVM. This integration uses this DVM as the configuration file to extract parameters in the run-time environment. This DVM provides a lookup source for default values, endpoint configuration, and so forth. This integration uses all flows that map run-time parameters.

Columns You Use to Map Run-Time Parameters

To map run-time parameters, you use the following columns:

- Column 1: Parameter
- Column 2: Value

Entries You Must Include to Map Parameters

Table 80 lists the entries you must include in the SBL_Expertise_Code_Definition DVM. You must use the case and spaces for each parameter as they appear in Table 80. You cannot change the case, and you must include the spaces.

Table 80. Entries You Must Include in the SBL_ORS_Parameter_Definition DVM

Parameter	Description
ORS-Request Slot MaxCost	Used to book an appointment. If this integration requests a slot for an activity from Siebel CRM, then it sends the value of this parameter as the maximum allowed cost for that activity. The value can be any decimal number greater than 0.1. 999999 is an example value of the ORS-Request Slot MaxCost parameter.
ORS-No of Slots	Defines the number of slots to display on a slot request from Siebel CRM. The value can be any integer greater than 1. 15 is an example value of the ORS-No of Slots parameter.
ORS-Confirm Slot MaxCost	Used to confirm an appointment. If the cost to confirm an activity for a particular shift is greater than the cost specified in ORS-Confirm Slot MaxCost, then Oracle Real-Time Scheduler does not allow the activity. The value can be any decimal number greater than 0.1. 999999 is an example value of the ORS-Confirm Slot MaxCost parameter.
ORS-FindShift	If the ORS-Confirm Slot MaxCost parameter value is too high, then the scheduler rejects it and then examines the ORS-FindShift parameter to determine if the scheduler must locate another shift that meets the maximum cost that the ORS-Confirm Slot MaxCost parameter specifies. The value for the ORS-FindShift parameter is YES or NO. If the value is YES, then the scheduler attempts to find a suitable shift for that activity that costs less than the value in the ORS-Confirm Slot MaxCost parameter. NO is an example value of the ORS-FindShift parameter.

Table 80. Entries You Must Include in the SBL_ORS_Parameter_Definition DVM

Parameter	Description
ORS Default Service Area with Srv Regn	Determines if this integration forces Oracle Real-Time Scheduler to associate with a Service Area, and only that Service Area, the same name as the service region. If the ORS-Default Service Area with Srv Regn parameter is: ■ YES, then when this integration synchronizes the employee to Oracle Real-Time Scheduler from Siebel Service, it associates the Service Area with the request. ■ NO, then this integration does not send the Service Area mapping from Oracle Fusion Middleware to Oracle Real-Time Scheduler. You must use the Oracle Real-Time Scheduler user interface to map the Service Areas. This integration retains any mapping you do in the Oracle Real-Time Scheduler user interface even when it performs a resynchronization from Siebel Service. YES is an example value of the ORS-Default Service Area with Srv Regn parameter.
ORS-Relative Rank of DEFAULT Expertise	Specifies the relative rank of the default expertise level. If this integration creates a new attribute in Oracle Real-Time Scheduler from Siebel Service, even if no expertise level is specified for the attribute, then it creates a proficiency named default for the attribute. The value in the parameter specifies the relative value of the proficiency. If this integration sends the attribute with proficiency at some point in the future, then it compares the rank of this default expertise to other expertise ranks in order to determine whether the default proficiency is lower in precedence or higher in precedence. For more information, see "Mapping Oracle Real-Time Scheduler Expertise to Siebel CRM Skills" on page 114. The value 900 is an example value of the ORS-Relative Rank of DEFAULT Expertise parameter.
Siebel_ID_of_MWM_Base_TimeZone	Specifies the ROW_ID of the Timezone in Siebel CRM corresponding to the Base Timezone at Oracle Real-Time Scheduler. It gets the ROW_ID of the Timezone in Siebel CRM which corresponds to the Base Timezone defined in Oracle Real-Time Scheduler and place that value here. For example, if the Base timezone in Oracle Real-Time Scheduler is PST, then gets the ROW_ID of the Pacific Time Zone record from Siebel CRM.

Optional Entries You Can Include to Map Parameters

Table 81 describes the optional entries that you can include to map parameters.

Table 81. Optional Parameters for SBL_ORS_Parameter_Definition DVM

Parameter	Description
ORS-Request Slot MaxCost-*ServiceRegionName* ORS-No of Slots-*ServiceRegionName* ORS-Confirm Slot MaxCost-*ServiceRegionName* ORS-FindShift-*ServiceRegionName* ORS-Default Service Area with Srv Regn-*ServiceRegionName*	Defines parameters for the service region. For example: ORS-FindShift-*ServiceRegionName* where: ■ *ServiceRegionName* is the name of the service region. For example: ORS-FindShift-New York City. If you do not configure these service region parameters on the DVM, then this integration uses the default parameters that are described in "Entries You Must Include to Map Parameters" on page 116. Any service region parameter that you define on the DVM modifies the default parameter. Note the following examples: ■ 9990 is an example value for the ORS-Request Slot MaxCost-*ServiceRegionName* parameter ■ 20 is an example value for the ORS-No of Slots-*ServiceRegionName* parameter. ■ 7777 is an example value for the ORS-Confirm Slot MaxCost-*ServiceRegionName* parameter. ■ YES is an example value for the ORS-FindShift-*ServiceRegionName* parameter. ■ NO is an example value of the ORS-Default Service Area with Srv Regn-*ServiceRegionName* parameter.
Default Batch Size - Employee	Sets the number of records that this integration groups before it saves them to the Siebel database. For more information, see "About the Default Batch Size - Employee Parameter" on page 119. 100 is an example value of the Default Batch Size - Employee parameter.
Default Batch Size - Activity	Determines the synchronization batch size for activities. This usage is similar to the Default Batch Size - Employee parameter. For more information, see "About the Default Batch Size - Employee Parameter" on page 119. The value 100 is an example value of the Default Batch Size - Activity parameter.

Table 81. Optional Parameters for SBL_ORS_Parameter_Definition DVM

Parameter	Description
Default Batch Size - Address	Determines the synchronization batch size for addresses. This usage is similar to the Default Batch Size - Employee parameter. For more information, see "About the Default Batch Size - Employee Parameter" on page 119. The value 100 is an example value of the Default Batch Size - Address parameter.
Default Batch Size - Geocoding	Determines the batch size to retrieve geocodes for multiple addresses. The batch geocoding flow waits until the number of geocodes that this integration holds in memory reaches the value in the Default Batch Size - Geocoding parameter, and then saves them in the Siebel database. This usage is similar to the Default Batch Size - Employee parameter. For more information, see "About the Default Batch Size - Employee Parameter" on page 119. The value 100 is an example value of the Default Batch Size - Geocoding parameter.

About the Default Batch Size - Employee Parameter

During a service region data transfer, this integration extracts employees from the Siebel database and attempts to synchronize them with Oracle Real-Time Scheduler, one record at a time. It stores the status of the synchronization in the Siebel database, along with the timestamp of the synchronization operation. This status describes whether the synchronization is successful, not successful, and the reasons for any failed synchronization.

To set the number of records that this integration groups before it saves them to the Siebel database, you can use the Default Batch Size - Employee parameter. For example, if Default Batch Size - Employee is 100, then the service region data transfer flow waits until the synchronization status in memory is 100 records, and then writes these records to the Siebel Database.

If the overall total number of records is less than 100, then this integration saves the records to the Siebel database when it reaches the total.

You can use any integer value greater than 0.

Editing Domain Value Maps

This task is a step in "Process of Installing SOA Composites for This Integration" on page 93.

To edit a DVM, you use the Oracle Business Process Composer.

To edit domain value maps

1 Enter the following address in the URL window of a Web browser:

`http://MachineName:SOAServerPort/soa/composer`

This URL accesses the Oracle Business Process Composer.

2 Enter the administrator user name and password, and then click Login.

3 In the Business Process Composer, click Open, and then choose Open DVM.

4 Choose one of the DVMs you must edit.

You must edit each DVM that this integration requires. For more information, see "DVMs You Can Edit" on page 120.

You can use the Show drop-down menu to choose all DVMs, including DVMs in the Oracle MDS.

After you open a DVM, the Oracle Business Process Composer displays all existing mappings. The header specifies the columns in the DVM. All entries are read-only.

If a DVM is in edit mode, then the Oracle Business Process Composer displays the Add button. If you choose a particular record within a DVM, then the Oracle Business Process Composer displays the edit and delete modes.

5 Do any of the following:

- To edit the existing DVM, click Edit in the top area.
- To add a new entry, click Add, enter the new values, and then click OK.
- To edit a record, click the record, and then click the Edit button that resides next to the Add button.
- To delete an existing mapping, click the record and then click Delete.

  A confirmation screen appears.

6 Click Commit at the top of the page.

You can optionally save changes on the user session and make more changes later before you save them. However, to save these changes to the run time environment, you must click Commit.

DVMs You Can Edit

You can edit the following DVMs:

- SBL_to_ORS_ActivityStatusMapping. For more information, see "Mapping an Activity Status From Oracle Real-Time Scheduler to Siebel CRM" on page 107.
- ORS_to_Siebel_StopStatusMapping DVM. For more information, see "Mapping an Activity Status From Oracle Real-Time Scheduler to Siebel CRM" on page 107.
- SBL_Expertise_Code_Definition. For more information, see "Mapping Oracle Real-Time Scheduler Expertise to Siebel CRM Skills" on page 114.

- SBL_ORS_Parameter_Definition. For more information, see "Mapping an Activity Status From Oracle Real-Time Scheduler to Siebel CRM" on page 107.

A Integration Services

This appendix describes integration services that Siebel Field Service Integration to Oracle Real-Time Scheduler uses. This appendix includes the following topic:

WSDL Files That Are Used for the Integration Services

All the Web Service Definitions (WSDL Files) and the corresponding XML Schema Definitions (XSD Files) being referred in the WSDL files are available in the SOA Metadata Services under the following path:

WSDL Files: oramds:/apps/SBL_ORS_INTRG/v2/WSDL

XSD Files: oramds:/apps/SBL_ORS_INTRG/v2/XSD

Below is the list of WSDL files being used for the integration services.

- AdminDataMgmtServiceWS.wsdl
- AppointmentBookingSystem.wsdl
- RealTimeGEOCoding.wsdl
- RetrieveLatestActivityState.wsdl
- ShiftWeeklyTemplateMaintenanceSBLORS_EBF.wsdl
- StatusUpdates_WS.wsdl
- SyncBackDispatcherActivityUpdates.wsdl
- SyncBackShiftPOUBreakDispatcherUpdates.wsdl
- WorkCalendarMaintenanceSBLORS_EBF.wsdl

Below is the list of XSD files being used for the integration services

- addressing.xml
- absSchema.xsd
- absSiebel.xsd
- ABS_Orchestrator.xsd
- ActivityStateRetrievalOrchestrator.xsd
- AdminDataManagementService.xsd
- adminDataMgmtEmployeeSiebel.xsd

- BreakTemp.xsd
- despatchesPoll_table.xsd
- GeocodingAddress.xsd
- InactivesPoll_table.xsd
- IntegratedShiftBreakPOU.xsd
- ListOfShiftexceptionORSSvc.xsd
- ListOfShiftORS.xsd
- M1-ActivityDispatched.xsd
- M1-ActivityStatusInquiry.xsd
- M1-ConfirmAct.xsd
- M1-CrewShiftUpdateToHost.xsd
- M1-EmployeeMaintenance.xsd
- M1-EmployeeMaintenance_old.xsd
- M1-LocationMaintenance.xsd
- M1-RouteAppointmentRequestToScheduler.xsd
- MWMCustomHeaderVariables.xsd
- MWM_FaultSchema.xsd
- ORSActivityStatus.xsd
- ORS_WSDL_Fields.xsd
- ORS_WSDL_Packets.xsd
- ORS_WSDL_Webservices.xsd
- ShiftBreakPOU_SBL77_WSDL_IntegratedPOU.xsd
- ShiftBreakPOU_SBL77_WSDL_POUUpsert.xsd
- ShiftBreakPOU_WSDL_IntegratedPOU.xsd
- ShiftBreakPOU_WSDL_POUUpsert.xsd
- SiebelWSHeader.xsd
- statusUpdateResponse.xsd
- testadminDataMgmtEmployeeSiebel.xsd
- Update_Status_Activity.xsd
- Update_Status_Break.xsd
- Update_Status_POU.xsd
- Update_Status_Shift.xsd

B Mapping Definitions

This appendix describes mapping definitions that Siebel Field Service Integration to Oracle Real-Time Scheduler uses. It includes the following topics:

Mapping Definitions That Manage an Appointment

This topic describes mapping definitions to manage an appointment.

Mapping Definitions That Book, Confirm, Insert, or Reschedule an Appointment

Table 82 describes mapping definitions to book an appointment. Note the following:

- The integration component in Siebel CRM is a child object type of the integration object.
- The parent element is in the Oracle Real-Time Scheduler Web service schema.

Table 82. Mapping Definitions That Book, Confirm, Insert, or Reschedule an Appointment

Siebel CRM		Oracle Real-Time Scheduler		Description
Integration Component	Field Name	Parent Element	Field Name	
Action	Id	Task	hostExternalId	
	NoSoonerThanDate/ Planned		startDateTime	
	Due/ Plannedcompletion		endDateTime	

Table 82. Mapping Definitions That Book, Confirm, Insert, or Reschedule an Appointment

Siebel CRM		Oracle Real-Time Scheduler		Description
Integration Component	Field Name	Parent Element	Field Name	
	ServiceRegionId		serviceArea	
	ServiceRegionTimeZone		timeZone	
	LateCost		lateCost	
	WorkTimeMin(*60)		estimatedDuration	
	ServiceCity		city	
	ServiceCountry		country	
	ServiceLatitude		geocodeLatitude	
	ServiceLongitude		geocodeLongitude	
	ServiceState		state	
	ServiceStreetAddress		address2	
	ServiceZipCode		postal	
	Error Desc		ServerMessage/TEXT	
	Request_status		No Mapping Available	Value is set to SUCCESS or FAIL based on the message returned from MWM.
	Language code		No Mapping Available	Value is set in FMW after looking up the value in the DVM.
FS Activity Skill	ItemTypeName+LowChar1+ExpertiseCode		capabilityType	
	No Mapping Available		action	Value is hardcoded as M1UP in FMW.

Table 82. Mapping Definitions That Book, Confirm, Insert, or Reschedule an Appointment

Siebel CRM		Oracle Real-Time Scheduler		Description
Integration Component	Field Name	Parent Element	Field Name	
			taskType	Value is hardcoded as SIEBEL_GENERIC_ACTIVITY in FMW.
			overrideCapabilities	Value is hardcoded as M1YS in FMW.
			crewSize	Value is hardcoded as 1 in FMW.
			count	Value is hardcoded as 1 in FMW.
			appointment	Value is hardcoded as N in FMW.
			timeWindowUsage	Value is hardcoded as M1EF in FMW.
			conditionalBooking	Value is hardcoded as Y for confirm appointment in FMW.

Mapping Definitions That Synchronize a Status

This topic describes mapping definitions that synchronize a status update.

Mapping Definitions That Synchronize an Activity

Table 83 describes mapping definitions that synchronize an activity. The Siebel CRM fields in Table 83 reside in the Action business component. The Oracle Real-Time Scheduler fields reside in the Taskcomponent.

Table 83. Mapping Definitions That Synchronize a Activity

Siebel CRM	Oracle Real-Time Scheduler		Description
Field Name	Parent Element	Field Name	
Id		hostExternalId	
Status		StatusFromHost	
StatusUpdateTimeStamp		StatusUpdateDate	

Table 83. Mapping Definitions That Synchronize a Activity

Siebel CRM Field Name	Oracle Real-Time Scheduler Parent Element	Oracle Real-Time Scheduler Field Name	Description
RESULT			Value is set as FAIL or SUCCESS in FMW.
languageCode			Value is set in FMW after looking up in the DVM

Mapping Definitions That Synchronize a Shift, Break, or POU

Table 84 describes mapping definitions that synchronize a shift, break, or POU. The Siebel CRM fields in Table 84 reside in the Integrated Shift Break POU business component. The Oracle Real-Time Scheduler fields reside in the Shift, Break, or POU component.

Table 84. Mapping Definitions That Synchronize a Shift, Break, or POU

Siebel CRM Field Name	Oracle Real-Time Scheduler Parent Element	Oracle Real-Time Scheduler Field Name	Description
EntityReferenceID		ShiftID	
Status		boStatus	
Shift		ShiftID	
crewName		TechnicianID	
DueDate		startDateTime	
EndTime		completionDateTime	
EntityReferenceID		taskId	
Type			Value is hardcoded as SHIFT_BREAK in FMW.
languageCode			Value is set in FMW after looking up in the DVM.

Mapping Definitions That Update Data in Real Time

This topic describes mapping definitions that update data in real time.

Mapping Definitions That Update Employee Data in Real Time

Table 85 describes mapping definitions that update exception data in real time. Except where noted in the table, the Siebel CRM fields in Table 85 reside in the Employee business component.

Table 85. Mapping Definitions That Update Employee Data in Real Time

Siebel CRM		Oracle Real-Time Scheduler		Description
Integration Component	Field Name	Parent Element	Field Name	
Employee	Id	Crew	externalUserId	
	FirstName		firstName	
	LastName		lastName	
	RowId		externalUserId	
	SchedulingAvailabilityEndDate		subscriptionEndDate or endDateTime	
	SchedulingAvailabilityStartDate		subscriptionStartDate or startDateTime	
	ServiceRegionId		serviceArea	
	TimeZoneNameStd		timeZone	
	languageCode		Value is set in FMW after looking up in the DVM.	
Employee Skill / Employee Skill Item	ItemTypeName+LowChar1+ExpertiseCode		skill	
	ItemTypeName+LowChar1+ExpertiseCode		description	

Mapping Definitions That Update Schedule Data in Real Time

Table 86 describes mapping definitions that update schedule data in real time. Except where noted in the table, the Siebel CRM fields in Table 86 reside in the Shift business component.

Table 86. Mapping Definitions That Update Employee Data in Real Time

Siebel CRM		Oracle Real-Time Scheduler		Description
Integration Component	Field Name	Parent Element	Field Name	
Shift	Id	Shift Weekly Template	externalId	
	ExceptionId		externalExceptionId	
	Description		description	
	SyncSuccessFlag			Value is set as Y or N.
	languageCode			Value is set in FMW after looking up in the DVM.
			bo	Value is hardcoded as M1-CrewShiftTemplate in FMW.
			crewShiftType	Value is hardcoded as SIEBEL_GENERIC_SHIFT_TYPE in FMW.
			shiftCostProfile	Value is hardcoded as SIEBEL_STD_PROFILE in FMW.
			dripMode	Value is hardcoded as M1AL in FMW.
			shiftWeeklyTemplateId	Value is hardcoded as SIEBEL_GENERIC in FMW.
			standbyShift	Value is hardcoded as M1NO in FMW.

This integration maps Shift Hours from Oracle's Siebel CRM to the Shift Weekly Template in Oracle Real-Time Scheduler by applying following logic:

For all the Shift Hours with same Start Day:

- The Start Time of the record with the Shift Start flag set to Y is mapped to Start Time for the Shift in Oracle Real-Time Scheduler.

- The End Time of the last record is mapped to the End Time for Shift in Oracle Real-Time Scheduler.
- The Gaps between these records are being mapped to Breaks in Oracle Real-Time Scheduler.

Mapping Definitions That Update Exception Data in Real Time

Table 87 describes mapping definitions that update exception data in real time. Except where noted in the table, the Siebel CRM fields in Table 87 reside in the Employee business component. The Oracle Real-Time Scheduler fields reside in the Crew component.

Table 87. Mapping Definitions That Update Exception Data in Real Time

Siebel CRM		Oracle Real-Time Scheduler		Description
Integration Component	Field Name	Parent Element	Field Name	
Employee	Id	Crew	externalUserId	
	FirstName		firstName	
	LastName		lastName	
	RowId		externalUserId	
	SchedulingAvailabilityEndDate		subscriptionEndDate or endDateTime	
	SchedulingAvailabilityStartDate		subscriptionStartDate or startDateTime	

Mapping Definitions That Retrieve Geocodes for Single Addresses and Multiple Addresses

Table 88 describes mapping definitions that retrieve geocodes for single addresses and multiple addresses. The Siebel CRM fields in Table 88 reside in the CUT Address business integration component. The Oracle Real-Time Scheduler fields reside in the Address component.

Table 88. Mapping Definitions That Retrieve Geocodes for Single Addresses and Multiple Addresses

Field Name in Siebel CRM	Field Name in Oracle Real-Time Scheduler	Description
StatusDesc	geocdFailescVal	Any failure during the process of getting geocodes will be mapped here.
LanguageCode		Value is set in FMW after looking up the information in the DVM.
StreetAddress	ADDRESS1	No more information available.
StreetAddress2	ADDRESS2	
CrossStreet	CrossStreet	FMW has the required logic to map the Province, County, or Metro Area from Siebel CRM to stateOrProvince if State information is not available.
Postal Code	Postal	No more information available.
Country	COUNTRY	
GeocodeValidFlag	GeocodeValidFlag	Value is set to Y or N in FMW.
GeocodeFailDesc	GeocodeFailDesc	Value is set to the error message returned from Oracle Real-Time Scheduler.

Index

S

T

W