SCM Cloud

Integrating Service Logistics with Field Service

24B

SCM Cloud
Integrating Service Logistics with Field Service

24B

F92875-01

Author: Stacey Tucker-Blosch

Contents

ORACLE

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Use help icons ⑦ to access help in the application. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons.

Get Support

You can get support at *My Oracle Support*. For accessible support, visit *Oracle Accessibility Learning and Support*.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest *ideas* for product enhancements, and watch events.

Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program*. Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to *oracle_fusion_applications_help_ww_grp@oracle.com*.

Thanks for helping us improve our user assistance!

ORACLE

1 About This Guide

Audience and Scope

This guide outlines the implementation and configuration steps required to integrate Oracle Fusion Service Logistics and Oracle Field Service Cloud (OFSC) to create a value-added business process and user experience. The administrator must enter the documented configurations and install the documented files to create the integration.

Each implementation of Oracle Field Service Cloud and Oracle Fusion Service Logistics is unique, and leads to the implementation of application customizations that support unique business requirements. While the steps in this document describe how to connect a non-customized Oracle Field Service Cloud instance to a non-customized Oracle Fusion Service Logistics instance, they can be combined with customizations that have already been applied to each instance.

> **Note:** With release 20A (11.13.20.01.0), "Oracle Engagement Cloud" is now known as Oracle CX Sales and Oracle Fusion Service. Existing Oracle Engagement Cloud users will retain access to Oracle CX Sales and Fusion Service features under their preexisting licensing agreements. Any new users created within your current Oracle Engagement Cloud license count will also retain the same access to Oracle CX Sales and Oracle Fusion Service. To obtain additional features or manage your subscription, refer to your Oracle Cloud Applications Console. This document describes features available to users under Oracle CX Sales, Oracle Fusion Service, and Oracle Engagement Cloud licensing agreements.

Related Guides

To understand more about the information covered in this guide, refer to the following table for a list of related guides.

Title	Description
Getting Started with Service Logistics Implementation	Lists the steps required to configure Service Logistics
Using Service Logistics	Describes the functionality and user tasks for Service Logistics

ORACLE

2 Introduction

Integration Component Architecture Between Service Logistics and Field Service

Service Logistics lets customers optimize parts logistics, source and order service parts, record costs, and invoice customers. Field Service offers time-based, self-learning, and predictive technology for dispatching field service technicians to resolve customer issues.

The Service Logistics to Field Service integration offers the following:

Field Service Technicians Download: Field Service Technicians are set up as Person Parties in Oracle Fusion Trading Community Model. A usage of Field Service Technician can be associated with a Person Party from several Oracle Cloud User Interfaces (UIs) including the Service Logistics Manage Field Service Technicians setup UI. An Integration Cloud Service is then performed to create the technician in Field Service if the technician doesn't exist or update the technician if they already exist.

Stocking Locations and Inventory Balances Download: Subinventories defined in Oracle Fusion Inventory Management can be set up as Stocking Locations in the Service Logistics Manage Stocking Locations setup UI. Once defined as Stocking Locations, these subinventories are assigned a type of Technician. Stocking Locations can then be assigned to technicians in the Service Logistics Manage Field Service Technicians setup UI. Stocking locations hold parts inventory that technicians use when fixing customer issues. An Integration Cloud Service is performed on a periodic basis that syncs stocking locations and their inventory balances from Inventory Management to Field Service.

Stocking Locations and Inventory Balances Incremental Download: Instead of loading all inventory balances, which can take a long time to complete, only those items transacted on the same day as the integration are loaded. This is a smaller subset of all items and runs much faster. You should set up this integration to run multiple times a day and every day of the week.

Technician Inventory Balances Download: As an alternative to the Stocking Locations and Inventory Balances download, this integration stores the inventory balances directly on the technician resource instead of the truck resource. This is a more common approach among Oracle Field Service Cloud customers. Only inventory balances in the technician's default usable stocking location, as defined in Service Logistics, are synced. This integration should only be used if the technician is assigned to only one stocking location.

Technician Inventory Balances Incremental Download: Instead of loading all inventory balances, which can take a long time to complete, the only items loaded are those that have been transacted on the same day that you run the integration. This is a smaller subset of all items and runs much faster. You should set up the integration to run multiple times a day and every day of the week. This integration should be used if the technician is assigned to only one stocking location.

Part Item Number Download: The Supply Chain Cloud parts catalog is downloaded to Oracle Field Service Cloud so that field service technicians know which parts can be ordered.

Parts Orders Integration: From Oracle Field Service Cloud, the technicians can order parts that are required for an activity or they can order parts to replenish their trunk stock. Service Logistics sources the parts and creates the transfer orders. Parts orders for an activity is also visible through the B2B Service work order.

ORACLE

Technician Receive Parts: This integration is a follow-up for the parts orders integration. It allows field service technicians to receive parts in Oracle Field Service Cloud, which were either ordered for an activity or used to replenish their trunk stock.

Preventive Maintenance Work Orders: For customer owned asset's preventive maintenance execution, you must convert the maintenance forecast results into service requests and work orders. You can schedule this integration to perform these tasks. Customers can configure different business rules to:

1. Retrieve the work order area from the Oracle Field Service Cloud setup.
2. Pass it as an input.
3. Generate the preventive maintenance work orders.

Field Service Debrief Integration: Field service activities are created when a service work order is created in the Oracle Fusion Service UIs. Field Service Technicians use the Oracle Field Service Cloud Debrief UI on their mobile devices to report on:

- Labor hours recorded.

- Parts used and recovered.

- Expenses incurred on the activities assigned to them.

When the activity is completed, these debrief transactions are automatically created in Service Logistics and are visible in the Manage Charges and Estimates page. Field Service Administrators can then:

1. Review these debrief transactions.
2. Make any changes or corrections.
3. Post the debrief transaction.

This generates a customer invoice for billing, adjusts inventory balances, updates customer's asset configuration, and captures cost of service.

The following figure displays how Oracle Fusion Service Logistics components and Oracle Field Service Cloud components are integrated using Oracle Integration Cloud Service.
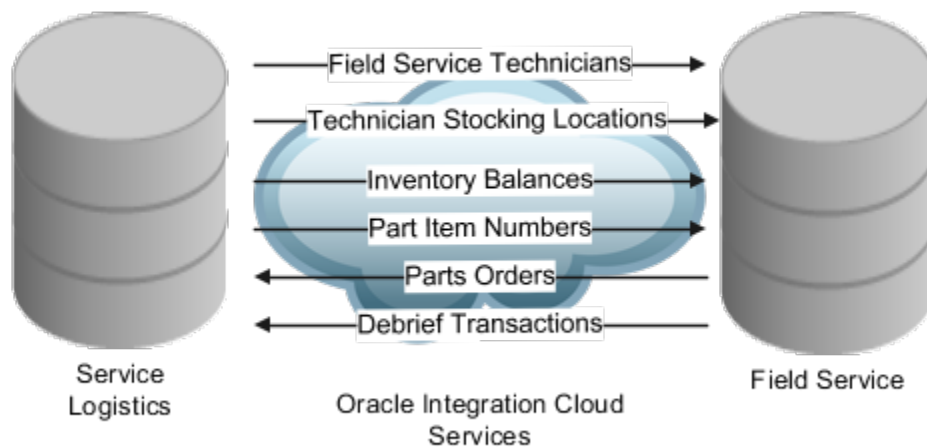
Oracle Integration Cloud (OIC)

Use the **Oracle Service Logistics — Oracle Field Service | Integrate Supply-Chain Data** recipe as a sample for syncing technicians, parts, and inventory between Oracle Fusion and Field Service clouds, and for generating preventive maintenance orders and configuring custom charge processing rules.

ORACLE

> **Note:** The **Oracle Service Logistics — Oracle Field Service | Integrate Supply-Chain Data** recipe is available in the Integration Store. It's only intended for use as a sample and it isn't confirmed to be error-free. Oracle doesn't provide support for this recipe.

Connections

- Connection: Service Logistics Technician
- Connection: Service Logistics OFSC
- Connection: Service Logistics Locations
- Connection: Service Logistics ERP Cloud
- Connection: Service Logistics OFSC REST
- Connection: Service Logistics Fusion REST

Integrations

- Integration: Oracle Service Logistics OFS Technician
- Integration: Oracle Service Logistics OFS Inventory
- Integration: Oracle Service Logistics OFS Inventory Incremental
- Integration: Oracle Service Logistics OFS Technician Inventory
- Integration: Oracle Service Logistics OFS Technician Inventory Incremental
- Integration: Oracle Service Logistics OFS Order Parts
- Integration: Oracle Service Logistics OFS Replenish Parts
- Integration: Oracle Service Logistics OFS Debrief
- Integration: Oracle Service Logistics OFS Receive Parts
- Integration: Oracle Service Logistics OFS Preventive Maintenance
- Integration: Oracle Service Logistics OFS Parts Catalog
- Integration: Oracle Service Logistics OFS Auto Debrief
- Integration: Oracle Service Logistics OFS Custom Rule Processor

> **Note:** Starting with the 24B release, all the integrations are recipes in the Projects feature of OIC Gen 3, which is available if you're moving from OIC Gen 2 to OIC Gen 3.

To get the **Oracle Service Logistics — Oracle Field Service | Integrate Supply-Chain Data** recipe:

- Gen 2: Go to the Oracle Integration Store and search the Gen 2 recipe. Enter **service logistics** in the search bar and search.
- Gen 3: On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.
  a. On the Oracle Integration home page, in the **Use a recipe/accelerator** section, click **View all**.
  b. Enter **service logistics** in the search bar and search.
  c. Click **Configure** on the recipe to configure its resources. The project workspace opens, displaying all the recipe resources.
  d. Configure the resources before you activate and run the recipe.

For more information, see the Integration Component Architecture Between Oracle Fusion Service and Oracle Field Service topic in the Integrating Fusion Service with Field Service guide.

Related Topics

- Integration Component Architecture Between Oracle Fusion Service and Oracle Field Service

Requirements and Licensing

Subscriptions to the following cloud services are required for implementing the Service Logistics and Field Service integration using Oracle Integration Cloud Service:

- Oracle Fusion Service Logistics: The integration is designed to work with Oracle Fusion Service Logistics Release 21A or later. Note that you can't get a direct subscription to Oracle Fusion Service Logistics. You need to subscribe to other Supply Chain cloud products to opt-in for the Service Logistics functional area.

- Oracle Field Service Cloud: The integration is designed to work with Oracle Field Service Cloud Release 21A or later.

- Oracle Integration Cloud Service: The integration is designed to work with Oracle Integration Cloud or Integration Cloud Service(ICS) version 20.3.3.0.0 or later.

ORACLE

3 Service Logistics Configuration

Configure Service Logistics

To configure Service Logistics for this integration, you must do the following:

1. Create an integration user that uses the relevant privileges associated with the Field Service Administrator job role.

   > **Note:** Although Service Logistics offers predefined job roles, you should create your own unique job roles and enable only the functionality that you plan to use.

2. Complete the following Service Logistics Setup:

   o Set profile option **Default Parent Resource Name** to match the top node in the resource hierarchy in Oracle Field Service. Note that the **Default Parent Resource Name** is a text field that must exactly match the external name of the resource in Oracle Field Service. The Default Parent Resource Name must be set up in the Work Area lookup before it can be added to this profile.

ORACLE

4 Field Service Configuration

Configure Field Service

To configure Field Service for the integration with Service Logistics, do the following:

1. Set up API Access in the Configuration - Application user interface.
2. Set up the unit of measures for the items that have inventory balances in the technician stocking locations. To do this:
   a. In Field Service, navigate to Configuration > Properties.
   b. Search for the property called **UOM**. Enter values for **Unit of Measure** and **UOM Code**.
3. Set up service activity codes for labor debrief.
   a. Navigate to Configuration > Properties.
   b. Search for property label **labor_service_activity**.
   c. Enter values for **Service Activity** and **Service Activity Code**. The service activity code must match the service activity code defined in Service Logistics.
4. Set up service activity codes for labor expense.
   a. Navigate to Configuration > Properties.
   b. Search for property label **expense_service_activity**.
   c. Enter values for **Service Activity** and **Service Activity Code**. The service activity code must match the service activity code defined in Service Logistics.
5. Set up service activity codes for parts used debrief.
   a. Navigate to Configuration > Properties.
   b. Search for property label **part_service_activity_used**.
   c. Enter values for **Service Activity** and **Service Activity Code**. The service activity code must match the service activity code defined in Service Logistics.
6. Set up service activity codes for parts returned debrief.
   a. Navigate to Configuration > Properties.
   b. Search for property label **part_service_activity_returned**.
   c. Enter values for **Service Activity** and **Service Activity Code**. The service activity code must match the service activity code defined in Service Logistics.
7. Set up labor items.
   a. Navigate to Configuration > Properties.
   b. Search for property label **labor_item_number**.
   c. Enter values for labor item. The labor item must match the item number defined in Product Information Management.
8. Set up expense items.
   a. Navigate to Configuration > Properties.
   b. Search for property label **expense_item_number**.
   c. Enter values for expense item. The expense item must match the item number defined in Product Information Management.

5 Oracle Integration Cloud Service Configuration

Configure Oracle Integration Cloud Services

Oracle Integration Cloud Services is used to sync Service Logistics field service technicians, stocking locations, inventory balances, and product catalog with Field Service. The following tasks must be performed to set up the secure integration between customer-specific instances:

1. Create the Oracle Integration Cloud User.
2. Import the Oracle Integration Cloud integration flows:

    a. Sign in to Oracle Integration Cloud.
    b. Navigate to **Integrations**, then **Packages**.
    c. Click the **Import** button.
    d. Select the file downloaded from the Marketplace.
3. Specify connection url and credentials for all Service Logistics connections.
4. After all connections are defined, activate the Service Logistics integrations.

See the section Oracle Integration Cloud in Chapter 2 for a list of these connections and integrations.

Verify Integration Synchronization

Follow the steps listed below to verify the synchronization.

In Service Logistics:

1. Open the Manage Field Service Technicians user interface.
2. Add a new person as a field service technician and assign a stocking location to the user.
3. In Oracle Integration Cloud, search for integration **Service Logistics Technicians** and select **Submit Now**. If integration completes successfully, sign in to Oracle Field Service and verify that the technician is available.

Set Up Password Policies for Integration Users

You can set up password policies if you'd like to limit the number of days that a user's password will work. Expiration days can be set for up to 9999 days.

To set up password policies for integration users:

1. Sign in as a system administrator.
2. Open the Security Console page.
3. Create a new user category.
4. Edit the password policy for the new user category.

ORACLE

5. Add the integration user to the new user category.

ORACLE

6 Integration Process Flow

Understand How the Integration Works

To understand how the Service Logistics to Field Service integration works, see the following sections:

Technician Download (Integration Name: Service Logistics Technician): Service Logistics field service technicians are downloaded to Field Service according to the following steps:

1. The Oracle Integration Cloud flow is a scheduled integration that you can run on demand or on a schedule.
2. A SOAP Service (`PersonService.findPerson`) is called to get a list of all field service technicians.
3. An OFSC Adapter (`resources.Update Resource`) is called to update the technician resource if it already exists.
4. If resource doesn't exist:

   - A common REST Service (`profileValues`) is called to get the parent node for the resource from profile **Default Parent Resource Name**.
   - OFSC Adapter (`Resource.Create Resource`) is called to create the resource. The field service technician resource being created will be assigned a parent resource as defined in the profile.
5. The technician details that are downloaded to OFSC include:

   - Parent Resource
   - Person Party ID
   - Full Name
   - Email
   - Mobile Phone Number
   - Status(active/inactive)

> **Note:** You can run:
> - Inventory balances integrations to store inventory on a truck resource.
>
> - Technician inventory balances integrations to store inventory on a technician resource directly.

Inventory Balances Download (Integration Name: Service Logistics Inventory): Inventory balances for technician stocking locations are downloaded to Field Service using the following steps:

1. The Oracle Integration Cloud flow is a scheduled integration that you can run on demand or on a schedule.
2. A Service Logistics REST Service (`stockingLocations` REST API) is called to get all technician stocking locations.
3. OFSC Adapter (`Resource.Get Resource`) is called to check if the stocking location already exists.
4. If the stocking location doesn't exist:

   - OFSC Adapter (`Resource.Create Resource`) is called to create the stocking location as a truck resource. The truck resource is tied to the parent resource (from profile **Default Parent Resource Name**).
5. Service Logistics REST Service (`trunkStocks `) is called to get inventory balances for the stocking location.
6. OFSC REST Service (`resources/custom-actions/bulkUpdateInventories`) is called to replace inventory balances in Field Service.
7. The stocking location details that are downloaded to OFSC include:

ORACLE

- ○ Resource ID (Truck ID)
- ○ Stocking Location Name (Organization Code + Subinventory Name)
- ○ Item Number
- ○ Item Description
- ○ Item Revision
- ○ Serial Number
- ○ On-hand Quantity
- ○ Primary Unit of Measure

Inventory Balances Incremental Download (Integration Name: Service Logistics Inventory Incremental): Newly updated inventory balances for technician stocking locations are downloaded to Field Service using the following steps:

1. The Oracle Integration Cloud flow is a scheduled integration that you should run multiple times a day and every day of the week.
2. A Service Logistics REST Service (`stockingLocations` REST API) is called to get all technician stocking locations.
3. OFSC Adapter (`Resource.Get Resource`) is called to check if the stocking location already exists.
4. If the stocking location doesn't exist:
    - ○ OFSC Adapter (`Resource.Create Resource`) is called to create the stocking location as a truck resource. The truck resource is tied to the parent resource (from profile **Default Parent Resource Name**).
5. The Inventory REST Service `(inventoryCompletedTransactions)` is called to find all items that have been transacted on the current day.
6. The Service Logistics REST Service (`trunkStocks`) is called to get inventory balances for the item.
7. The OFSC REST Service (`resources/custom-actions/bulkUpdateInventories`) is called to create or update inventory balances in Field Service.
8. The stocking location details that are downloaded to OFSC include:
    - ○ Resource ID (Truck ID)
    - ○ Stocking Location Name (Organization Code + Subinventory Name)
    - ○ Item Number
    - ○ Item Description
    - ○ Item Revision
    - ○ Serial Number
    - ○ On-hand Quantity
    - ○ Primary Unit of Measure

Technician Inventory Balances Download (Integration Name: Service Logistics Technician Inventory): Inventory balances for technician's default usable subinventory are downloaded to Field Service using the following steps:

1. The Oracle Integration Cloud is a scheduled integration that you can run on demand or on a schedule.
2. The Service Logistics REST Service (`stockingLocations` REST API) is called to get all the technician default usable stocking locations.
3. The Service Logistics REST Service (`trunkStocks`) is called to get inventory balances for the stocking location.
4. The OFSC REST Service (`resources/custom-actions/bulkUpdateInventories`) is called to replace inventory balances in Field Service.
5. The stocking location details that are downloaded to OFSC include:
    - ○ Resource ID (Technician Party ID)

ORACLE

- ○ Stocking Location Name (Organization Code + Subinventory Name)
- ○ Item Number
- ○ Item Description
- ○ Item Revision
- ○ Serial Number
- ○ On-hand Quantity
- ○ Primary Unit of Measure

Technician Inventory Balances Incremental Download (Integration Name: Service Logistics Technician Inventory Incremental): Newly updated inventory balances for technician's default usable stocking locations are downloaded to Field Service using the following steps:

1. The Oracle Integration Cloud is a scheduled integration that you should run multiple times in a day and every day of the week.
2. The Service Logistics REST Service (`stockingLocations` REST API) is called to get all the technician default usable stocking locations.
3. The Inventory REST Service `(inventoryCompletedTransactions)` is called to find all items that have been transacted on the current day.
4. The Service Logistics REST Service (`trunkStocks`) is called to get inventory balances for the item.
5. The OFSC REST Service (`resources/custom-actions/bulkUpdateInventories`) is called to update inventory balances in Field Service.
6. The stocking location details that are downloaded to OFSC include:

   - ○ Resource ID (Technician Party ID)
   - ○ Stocking Location Name (Organization Code + Subinventory Name)
   - ○ Item Number
   - ○ Item Description
   - ○ Item Revision
   - ○ Serial Number
   - ○ On-hand Quantity
   - ○ Primary Unit of Measure

Part Item Number Download (Integration Name: Service Logistics Parts Catalog): Field Service Technicians need part item numbers to order replacement parts. Part item numbers are downloaded using the following process:

1. A batch program loads items from the Oracle Product Information Cloud to Field Service using Oracle Integration Cloud. The batch program is an OIC integration program that can be run on demand or scheduled to run from OIC.

ORACLE

2. This integration downloads all items for the inventory organization defined in profile Default Inventory Organization. Only items with Service Logistics Billing Type tied to Billing Category = Material are included. The item details downloaded include:
   - Item Number
   - Item Description
   - Item Revision
   - Primary Unit of Measure

   > **Note:** You must set up the labor and expense items in OFSC to match the labor and expense items in the item master in Product Information Management. See Chapter 4 for details.

Order Parts for an Activity (Integration Name: Service Logistics Order Parts): Parts orders integration between Service Logistics and Field Service occurs as follows:

1. Field service technicians click the **Order** button to order parts from OFSC using Service Logistics.
2. This creates an order activity in OFSC.
3. Step 1 raises an event that triggers the integration.
4. An OFSC Adapter (`Activity Inventory`) is called to get all the parts ordered by the technician.
5. The data elements passed from OFSC to Service Logistics are:
   - Item Number
   - Quantity
   - Unit of Measure
   - Ship To Address Type (Technician or Customer)
   - Work Order ID (Fusion Service Work Order ID)
6. A Service Logistics REST service (`partRequirementLines`) is called to create part requirements and find a supply source (`partRequirementLines/{partsReqLineId}/action/getPreferredSource`).
7. The supply orchestration REST service (`supplyRequests`) is used to create the transfer order to ship the parts to the technician or to the customer site.
8. The parts ordered by the field service technician for the work order appear in the Oracle Field Service work order, along with other parts that ordered by the agent using the Fusion Service pages.
9. If the part isn't found, a backorder is created for the replenishment source.
10. The following is downloaded to OFSC:
    - Transfer order number
    - Order activity status
    - Transfer order header ID

Note that field service technicians can order more than one part number and more than one quantity of the part.

Order Parts to Replenish Trunk Stock (Integration Name: Service Logistics Replenish Parts): Parts orders integration between Service Logistics and Field Service occurs as follows:

1. Field service technicians click the **Order** button to order parts from OFSC to replenish their trunk stock.
2. This creates an order activity in OFSC.
3. Step 1 raises an event that triggers the integration.
4. An OFSC Adapter (`Activity Inventory`) is called to get all the parts ordered by the technician.
5. The data elements passed from OFSC to Service Logistics are:
   - Item Number

- ○ Quantity
- ○ Unit of Measure
- ○ Ship To Address Type (Technician or Customer)
- ○ Work Order ID (Fusion Service Work Order ID)

6. A Service Logistics REST service (`partRequirementLines`) is called to create part requirements and find a supply source (`partRequirementLines/{partsReqLineId}/action/getPreferredSource`).
7. The supply orchestration REST service (`supplyRequests`) is used to create the transfer order to ship the parts to the technician or to the customer site.
8. If the part isn't found, a backorder is created for the replenishment source.
9. The following is downloaded to OFSC:

- ○ Transfer order number
- ○ Order activity status
- ○ Transfer order header ID

Note that field service technicians can order more than one part number and more than one quantity of the part.

Debrief Integration (Integration Name: Service Logistics Debrief): Debrief integration between Field Service and Service Logistics happens as follows:

1. When the technician completes the activity in Field Service, an event is raised that triggers this integration.
2. A Field Service REST service (`activities/{activityId}/installedInventories`) is called to fetch all the labor, parts, and expense debrief lines.
3. A Field Service REST service `(activities/{activityId}/deinstalledInventories`) is called to fetch all returned parts.
4. A Service Logistics REST service (`debriefs/{debriefHeaderId}/child/lines`) is called to create the:

- ○ Debrief transactions
- ○ Charges
- ○ Reservations for the parts used (the reservation is released when charges are posted).

5. The Auto Process Debrief Charges job is launched and tries to post the charges automatically, based on a set of rules that you define.
6. The field service administrator must review charges that don't post automatically and post them manually.
7. The debrief information that uploads to Service Logistics includes:

   a. Labor Debrief

      - Service Activity
      - Labor Item
      - Start Time
      - End Time

   b. Material Debrief

      - Service Activity
      - Item Number
      - Quantity
      - Unit of Measure

   c. Expense Debrief

      - Service Activity

ORACLE

- Expense Item
- Amount
- Currency Code

Receive Parts (Integration Name: Service Logistics Receive Parts): This integration completes the receiving parts process by a field service technician. The integration between Service Logistics and Field Service occurs as follows:

1. The field service technician provides the received item quantity and clicks the **Receive** button, which starts the receiving process.

   The application:
   - Creates an activity in Oracle Field Service.
   - Raises an event that triggers the Service Logistics Receive Parts integration.

2. The integration process retrieves the activity details from Oracle Field Service Cloud using the Oracle Field Service Adapter (Activity).
3. The application calls an Inventory Management REST service (linesToReceive) to retrieve the expected shipment lines that can be received using the activity detail information (part item number and transfer order header ID).
4. If the application finds a valid shipment line, it calls another Inventory Management REST service (receivingTransaction) to create a receive transfer order. The application passes the following parameters while creating this transfer order transaction:
   - Quantity and UOM Code from Oracle Field Service Activity details.
   - Item number, Shipment Header Id, Shipment Line Id.
   - Destination Type code defaults to INVENTORY, Source Document Code to TRANSFER ORDER, and Transaction Type to RECEIPT.

Preventive Maintenance (Integration Name: Service Logistics Preventive Maintenance): This is a scheduled integration for generating preventive maintenance work orders.

1. Schedule Oracle Integration Service Logistics Preventive Maintenance to run on a daily or weekly basis.
2. This integration accepts the following five parameters:
   - MaintenanceOrganizationCode: Short name of the maintenance organization that runs the preventive work.
   - Category: Category for the service request.
   - WorkOrderIntegrationCd: Use ORA_WO_INT_OFSC for Oracle Field Service Cloud (OFSC) work order creation or ORA_WO_INT_SVC for generic work order creation.
   - WorkOrderStatusCode: Initial status code used for creating the work order.
   - WorkOrderTypeCd: Work order type code used for creating the work order.
3. This integration starts with setting a default time zone, which you can change based on the customer's time zone.
4. Based on the organization code and work order type code passed as input, the application retrieves the corresponding identifier to pass for the process.
5. The application calls Maintenance REST service (MaintenanceWorkOrders) to retrieve all released customer assets-based work orders with future planned start dates.
6. For each maintenance work order returned by the previous process, the application does the following:
   - Calls the Maintenance REST service (DocumentReference) to check for document reference of type ORA_SERVICE_REQUEST against the current maintenance work order. If there's reference data, then it skips processing.

ORACLE

- o If there's no reference data, the application retrieves current asset details information by calling an SCM REST service (installBaseAssets).
- o If the asset detail is missing the asset location, then it skips the processing. Otherwise, it calls the HCM SOAP API (LocationService) for the asset's current location ID to retrieve location details.
- o The application calls the SCM REST service (WorkDefinition) to retrieve work definition details based on the maintenance work order work definition ID. The application then calls CRM REST service (Contact) for the asset's contact details.
- o For an OFSC work order creation (based on the parameter that you set up before), the application retrieves the work area by calling the CRM REST service (svcWOAreas). By default, the country and postal code are passed to this rest service, but you can provide more parameters based on your setup.
- o For an OFSC work order creation: If there's an error during the work area retrieval processing, then the processing stops for the current maintenance work order.
- o Based on the service request category, the application calls the CRM REST service (Categories) to retrieve the category ID and then calls another CRM REST service using the customer number and customer site number for the asset to retrieve the asset address.
- o Create a service request by calling the CRM REST service (ServiceRequests) with the following parameters:

    - Title: Work definition name.
    - Problem Description: Work definition name description.
    - Account Party Id: Asset customer ID.
    - Primary Contact Party Id: Party ID from the contact details based on the asset contact.
    - Inventory Item Id: Asset item.
    - IB Asset Id: Asset ID.
    - Category Id: ID retrieved based on the service category name passed as the schedule parameter.

- o Create a work order by calling the CRM REST service (CustomerWorkOrders) with following parameters:

    - Title: Work definition name.
    - SId: Service request ID.
    - Account Party Id: Asset customer ID.
    - Contact Party Id: Party ID from contact details based on the asset contact.
    - IB Asset Id: Asset ID.
    - Wo Integration Cd, Work Order Status Code: From the corresponding input parameter.
    - Wo Type Id: Retrieved at the beginning of the processing based on the input provided as work order typeCd.
    - Wo Area: For OFSC work order, it's retrieved as mentioned.
    - Resolution Due Date: Planned completion date from maintenance work order.
    - Contact Name, Contact Email, Contact Phone: Contact details retrieved based on the asset's contact id as mentioned.
    - Address Component: Address details retrieved by calling SOAP API as mentioned.
    - Time Zone Code: Value of the default time zone as set in the integration.

- o For the current maintenance work order the application calls another SCM REST service (maintenanceWorkOrder) to find the maintenance operations that the work order requires. For each of these operations do the following:

    - For the current maintenance work order call the SCM REST service (maintenanceWorkOrders/ {WorkOrderId}/child/WorkOrderOperation/{WoOperationId}/child/WorkOrderOperationMaterial) to find materials that the work order requires to perform a maintenance operation.

ORACLE

- For each material create a part requirement line based on the following attribute mapping:

    - Destination Organization Id: Organization ID of the operation.
    - Need-by Date: Planned start date of operation.
    - Parent Entity Id: Work Order ID created during this integration processing.
    - Inventory Item Id, Quantity, UOM Code: As defined in operation materials.
    - Ship-to Address Type: Defaults as CUSTOMER.
    - Ship-to Party Id: Service request account party ID.

- The application calls the Maintenance REST service (maintenanceWorkOrders/{WorkOrderId}/child/documentReference) to record the reference data based on the following parameters:

    - Source System Type Code: Defaults to ORA_INTERNAL.
    - Work Order Id: Maintenance work order ID that the integration is currently processing.
    - Document Type: Defaults to ORA_SERVICE_REQUEST.
    - Document Header Id: Service request ID that's created during processing.

7. Error Handling: This scheduled integration creates service requests, work orders, and part requirement lines. It also validates various attributes. If there's a failure during processing, then the application:

    - Stops the processing for that maintenance work order.

    - Logs the message.

    - Deletes any data that it created during the processing, such as service requests, work orders, or part requirement lines.

Auto Parts Debrief (Integration Name: Oracle Service Logistics OFS Auto Debrief): This integration populates the debrief with parts information if the information is available for the work order. The integration flow between Service Logistics and Field service is as follows:

1. The field service technician starts the debrief process on Oracle Field Service Cloud (OFSC) by clicking the **Start** button, which starts an activity.
2. This activity raises an event that starts the Oracle Service Logistics OFS Auto Debrief integration.
3. The integration process receives the event details, which includes an activity ID and work order details.
4. The integration process calls an OFSC Rest API to find any parts already added for that debrief task.
5. If there's a part that has already been added, the processing stops.
6. If no part is found for the activity, the integration process calls the CustomerWorkOrders REST API to find the work order that's attached to the activity.
7. The integration process calls another partRequirementLines Rest API for the work order to find the parts already attached to the work order.
8. If there are parts already added with the work order, it tries to add these into the debrief as follows:
    a. Verifies the available parts in the resource inventory by calling the Oracle Field Service REST API (resources/{resourceId}/inventories).
    b. If there's a part, it's added to install inventory by calling the Oracle Field Service REST API (resources/{resourceId}/inventories/{inventoryId}/custom-actions/install)

Custom Rule Processor (Integration Name: Oracle Service Logistics OFS Custom Rule Processor): This is an example of an integration that shows how to use a custom rule for posting charge lines. Processing for this integration is as follows:

1. The integration starts when you post charges, if the Enable Custom Rule Processing when Posting Charges profile is set to Yes.
2. Based on the data, the integration process calls the REST API (debriefs/(debriefHeaderId)) to retrieve the debrief header information.

ORACLE

3. In this example, the customer account information from Step 2 is checked, and only for the specific customer posting the charges. This is done by calling the Auto Process Debrief Charges ESS job and for cases where the charge line is marked as Review.

> **Note:** This is a specific scenario. You can add more complex rules to process based on need.