SCM Cloud

Integrating Service Logistics with Field Service

26A

SCM Cloud
Integrating Service Logistics with Field Service

26A

G46500-01

Author: Stacey Tucker-Blosch

Contents

ORACLE

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

Some application pages have help icons ⑦ to give you access to contextual help. 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. If the page has contextual help, help icons will appear.

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.

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

ORACLE

1 About This Guide

Audience and Scope

This guide outlines the implementation and configuration steps required to integrate Oracle 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 more features or manage your subscription, see your Oracle Cloud Applications Console. This document describes features available for Oracle CX Sales, Oracle Fusion Service, and Oracle Engagement Cloud licensing agreements.

Related Guides

To understand more about the information covered in this guide, see 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.
Integrating Fusion Service with Field Service	Describes component architecture between Fusion Service and Field Service.
Using Integrations in Oracle Integration 3	Describes components, provides product tours, tutorials, and reviewing tips.
Using Service Logistics	Describes the functionality and user tasks for Service Logistics.

ORACLE

2 Introduction

## Integration Component Architecture Between Service Logistics and Field Service

Information flows between Service Logistics and Field Service using integrations.

Overview

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

Oracle Integration (OIC) offers a rich set of business and technical integrations called accelerators, and also sample templates called recipes. See the Requirements and Licensing topic to learn about the required subscriptions details.

This diagram shows how information flows between Service Logistics and Field Service using the integrations in the **Oracle Service Logistics — Oracle Field Service | Integrate Supply-Chain Data** recipe package:
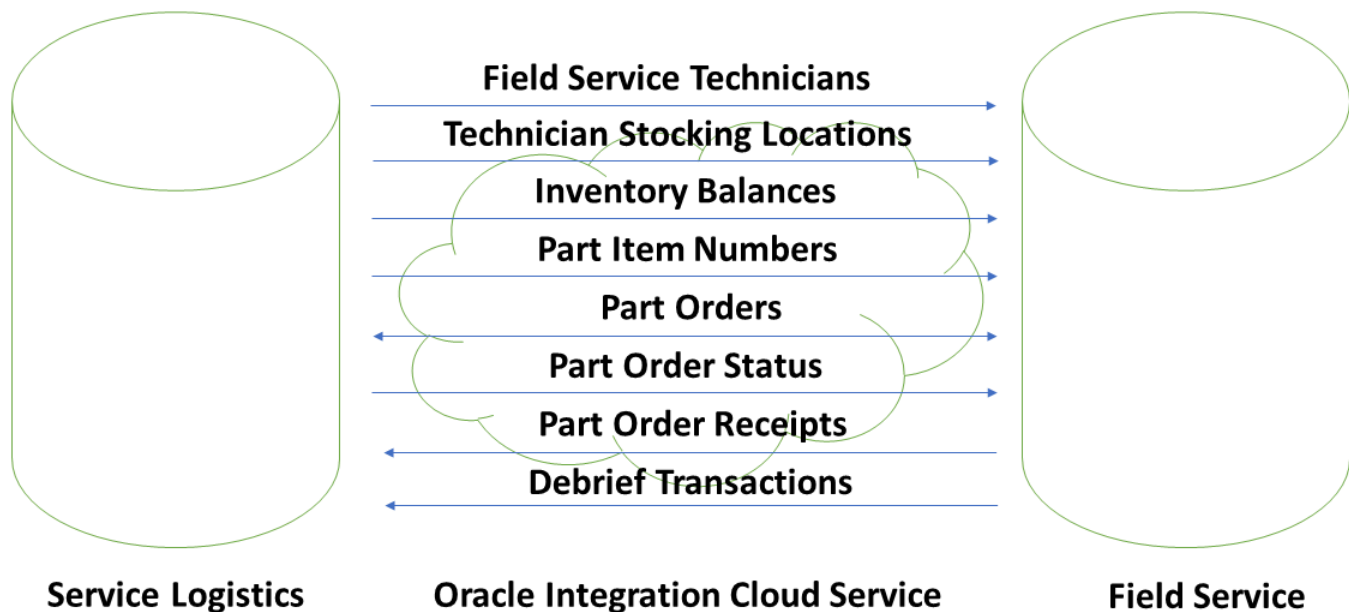
The following summarizes the Service Logistics to Field Service integrations:

Field Service Technicians Download: Field service technicians are set up as person parties in Oracle Fusion Trading Community model. A field service technician usage 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 then creates the technician in Field Service if the technician doesn't exist, or updates 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 technician type. 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 then periodically 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 takes a long time to complete, only 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: Another option 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 (OFSC) 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 OFSC, so that field service technicians know which parts they can order. Only items with a material service billing type and an inventory organization that matches the user's Default Inventory Organization profile option (INV_DEFAULT_ORG_ID) are downloaded to OFSC.

Parts Order and Receiving: A field service agent can start a part order directly from Oracle Field Service Cloud (OFSC). This action generates a part requirement line within the associated work order. Based on the categorization of the part, the system either creates a transfer order (for inventory-managed parts) or a purchase requirement (for buy-only parts).

The parts processing integration between Oracle Service Logistics and OFSC is achieved through four key integrations, all are available as part of the Oracle Field Service | Integrate Supply-Chain Data recipe package in the Oracle Integration Cloud (OIC) Marketplace.

Preventive Maintenance Work Orders: In addition to the ESS job that generates Preventive Maintenance service requests and work orders, there's a recipe that does the same thing. However, the OIC recipe is highly configurable, while the ESS job isn't.

See the Set Up Field Service Preventive Maintenance topic in the Getting Started with Service Logistics Implementation Guide for details on how this OIC recipe gets the work order area from OFSC using the Fusion Service API, which is needed to create the Preventive Maintenance work order.

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 OFSC debrief UI on their mobile devices to report:

- Labor hours recorded.
- Parts used and recovered.
- Expenses incurred on the activities assigned to them.

ORACLE

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 the customer's asset configuration, and captures the cost of service.

Auto Parts Debrief: Part installation debrief lines are created in OFSC when the field service technician starts an OFSC activity. These debrief lines are created for parts ordered in Fusion Service and Service Logistics UIs, and for those created in OFSC.

Auto Order Creation:

Allows orders created in Service Logistics for a field service work order to appear in OFSC, so technicians can receive them. This also prevents technicians from ordering duplicate parts.

This integration occurs using two recipes. When a transfer order or purchase requisition is created in Service Logistics for parts ordering, Service Logistics calls the Oracle Service Logistics OFS Order Events recipe, which then calls the Oracle Service Logistics OFS Auto Create Order recipe.

Custom Rule Processor: Provides an option to enable a business rule while posting charges. Customers can implement a business specific rule that's configured in this integration, which acts as a post charge processing rules engine.

See the Extend Charges Posting Rules topic in the Getting Started with Service Logistics Implementation guide for details on how you can define custom processing rules to post charges in Service Logistics.

Purchase Order and Transfer Order Processing

As part of order processing, the OFSC part order status is set to Ready to Receive when a corresponding transfer order ships from inventory. For a purchase requisition order, the status is set to Ready to Receive after the purchase order is approved. Also, if an ASN is received, the purchase order can be automatically received for the technician, and the corresponding OFSC parts order status is updated.

> **Note:** This integration is for non-serialized items only. If you create an advanced shipment notification in the Supplier Dashboard, you can't enter serial numbers. It also only works for purchase orders created for part requirements that are created in OFSC.

Oracle Integration (OIC)

See the Oracle Integration (OIC) Recipe chapter for information about the **Oracle Service Logistics — Oracle Field Service | Integrate Supply-Chain Data** recipe package.

Related Topics
- About the Field Service Integration Recipe

Requirements and Licensing

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

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

ORACLE

3 Oracle Integration (OIC) Recipe

About the Field Service Integration Recipe

Use the **Oracle Service Logistics — Oracle Field Service | Integrate Supply-Chain Data** recipe package 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.

> **Note:** This recipe isn't confirmed to be error-free. Oracle doesn't provide support for this recipe package.

The recipe package includes these recipe integrations and connection points:

Recipe Integrations

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

Connection Points

- Connection: Service Logistics ERP Cloud REST
- Connection: Service Logistics Fusion REST
- Connection: Service Logistics Locations SOAP
- Connection: Service Logistics OFSC
- Connection: Service Logistics OFSC REST
- Connection: Service Logistics Technicians SOAP

ORACLE

Related Topics

Before You Install the Recipe

You must perform the following configuration tasks on your Fusion Service instance to successfully connect and sync to external systems using Oracle Integration.

Configure Service Logistics

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

1. Create an integration user that uses the 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:

   Set the Default Parent Resource Name profile option to match the top node in the resource hierarchy in Oracle Field Service.

   Note: 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.

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 UI.
2. Set up the unit of measures for the items that have inventory balances in the technician stocking locations.
   a. In Field Service, go to **Configuration**, then **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. Go to **Configuration**, then **Properties**.
   b. Search for the **labor_service_activity** property label.
   c. Enter values for the **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. Go to **Configuration**, then **Properties**.
   b. Search for the **expense_service_activity** property label.
   c. Enter values for the **Service Activity** and **Service Activity Code**. The service activity code must match the service activity code defined in Service Logistics.

ORACLE

5. Set up service activity codes for parts used in debrief.
   a. Go to **Configuration**, then **Properties**.
   b. Search for the **part_service_activity_used** property label.
   c. Enter values for the **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 in debrief.
   a. Go to **Configuration**, then **Properties**.
   b. Search for the **part_service_activity_returned** property label.
   c. Enter values for the **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. Go to **Configuration**, then **Properties**.
   b. Search for the **labor_item_number** property label.
   c. Enter values for the labor item. The labor item must match the item number defined in Product Information Management.
8. Set up expense items.
   a. Go to **Configuration**, then **Properties**.
   b. Search for the **expense_item_number** property label.
   c. Enter values for the expense item. The expense item must match the item number defined in Product Information Management.

Related Topics
- About the Field Service Integration Recipe
- Install and Configure the Recipe
- Activate and Run the Recipe

Install and Configure the Recipe

Install the recipe on your Oracle integration instance to deploy and configure the integration and associated resources.

To install and configure the recipe:
1. From the Oracle Integration home page, in the **Use a recipe/accelerator** section, click **View all**.
2. Find this recipe by searching for Service Logistics, then click **Get**.
3. Verify the confirmation message that says the recipe was successfully installed, and that the recipe card shows **In use**.
4. Click **Configure** on the recipe to configure its resources.
5. The project workspace opens, displaying all the recipe resources.

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.

Configure the Connections
Configure the following resources before you activate and run the recipe.

ORACLE

Configure the Service Logistics ERP Cloud REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Properties	Enter your Fusion instance URL for ERP Cloud Host for example, https://myinstance.fusionapps.com/.
Enable two way SSL for outbound connections (Optional)	Select **No**.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave **Basic Authentication** selected.
Username	Enter the FA username.
Password	Enter the password for the user.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click the **Go back** icon.

Configure the Service Logistics Fusion REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave **REST API Base URL** selected.
Connection URL	Enter your Fusion instance URL, for example, https://myinstance.fusionapps.com/.
Enable two way SSL for outbound connections (Optional)	Select **No**.

ORACLE

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave **Basic Authentication** selected.
Username	Enter the FA username.
Password	Enter the password for the user.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click the **Go back** icon.

Configure the Service Logistics Locations SOAP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Properties	Enter the FA WSDL URL for example, https://fusionapps:443/crmService/FoundationPartiesLocationService?WSDL.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave **Basic Authentication** selected.
Username	Enter the FA username.
Password	Enter the password for the user.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click the **Go back** icon.

Configure the Service Logistics OFSC Connection

1. In the Connections section, click the connection name.

ORACLE

2. In the Properties section, enter the following details:

Field	Information to Enter
Properties	Enter Field Service Cloud API URL.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave **Basic Authentication** selected.
Username	Enter the username that you defined as client id on OFSC for application connection.
Password	Enter the client secret from OFSC.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click the **Go back** icon.

Configure the Service Logistics OFSC REST Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Connection Type	Leave **REST API Base URL** selected.
Connection URL	Enter your Field Service instance rest URL, for example, https://myinstance.fusionapps.com/rest/ofsccode.
Enable two way SSL for outbound connections (Optional)	Select **No**.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave **Basic Authentication** selected.
Username	Enter the username that you defined as client id on OFSC for application connection.
Password	Enter the client secret from OFSC.

ORACLE

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click the **Go back** icon.

Configure the Service Logistics Technicians SOAP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the following details:

Field	Information to Enter
Properties	Enter the FA WSDL URL for example https://fusionapps:443/crmService/ FoundationPartiesPersonService?WSDL.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Leave **Basic Authentication** selected.
Username	Enter the FA username.
Password	Enter the password for the user.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.
6. To return to the project workspace, click the **Go back** icon.

Configure the Service Logistics Order Parts Lookup

When creating a transfer order or purchase requisition, the preparer and deliver-to requester information is required. These attributes are mapped in the integration within the CreateSupplyRequestLine node and the parts are stored in a lookup.

To configure this functionality you must update the SL_UTILITY_LKP lookup for the Preparer Person Id code.

The values correspond to the person_id of the user who's allowed to create transfer orders in the Fusion application. This is typically retrieved from the PER_PERSON_NAMES_F table.

Here's an example query:

- `SELECT person_id`

- `FROM fusion.per_person_names_f`

- `WHERE first_name = 'Marsha' AND last_name = 'Brent';`

In the lookup, you'll need to provide the person_id that you've retrieved from the query.

ORACLE

Set Up the Secure Integration Between Customer-specific Instances

Oracle Integration syncs Service Logistics field service technicians, stocking locations, inventory balances, and the 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 User.
2. Import the Oracle Integration flows:
   a. Navigate to Integration Store homepage.
   b. Click **Browse Store**.
   c. Enter **Service Logistics** in the **Search** field.
   d. Click **Get** to start installing the recipe.
3. Specify connection URL and credentials for all Service Logistics connections.
4. After all connections are defined, activate the Service Logistics integrations.

Verify Integration Synchronization

Follow the steps listed below to verify the synchronization.

In Service Logistics:

1. Open the Manage Field Service Technicians UI.
2. Add a new person as a field service technician and assign a stocking location to the user.
3. In Oracle Integration, search for the **Service Logistics Technicians** integration and select **Submit Now**. If the 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.
5. Add the integration user to the new user category.

Related Topics
- About the Field Service Integration Recipe
- Before You Install the Recipe
- Activate and Run the Recipe

Activate and Run the Recipe

After you've configured the connections and completed the necessary tasks mentioned before, you can activate and run the recipe.

ORACLE

How the Field Service Integration Recipe Works

To understand how the Field Service integration recipe works, review the descriptions in this section.

> **Note:** To confirm that the integration has been activated:
>
> 1. In the project workspace, click **Activate**.
>
> 2. In the Activate project panel, with the default project deployment selected, choose an appropriate tracing option, then click **Activate**.
>
> 3. A message confirms that the integration has been activated. Refresh the page to view the updated status of the integration.

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.
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.

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.

ORACLE

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
        - Expense Item
        - Amount
        - Currency Code

Capture Order Creation Event (Integration Name: Oracle Service Logistics OFS Order Events): This is an event-based integration that's called when a transfer order is created or a purchase requisition is approved in Service Logistics for a field service work order. The integration then calls the Oracle Service Logistics OFS Auto Create Order scheduled integration to create an order in OFSC.

Auto Create Order (Integration Name: Oracle Service Logistics OFS Auto Create Order): A scheduled integration that's called by the Oracle Service Logistics OFS Order Events recipe, and does the following:

1. Finds the part requirement and work order details.
2. Retrieves the field service activity against the field service work order for the related transfer order or purchase requisition.
3. Creates an order activity in OFSC with the order information from Service Logistics.
4. Creates customer inventory activity in OFSC.

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 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:

ORACLE

- ○ 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:

    - ○ 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 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

Order Parts for an Activity (Integration Name: Service Logistics Order Parts):

ORACLE

This integration is triggered whenever a part is ordered in OFSC for a work order. It performs the following actions:

- Creates a part requirement line in the associated work order.
- Depending on the part type, it creates either:
    - A transfer order (for inventory-managed parts).
    - A purchase requisition (for buy-only 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 back order is created for the replenishment source.
10. The following is downloaded to OFSC:

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

Note: Field service technicians can order more than one part number and more than one quantity of the part.

Receive Parts (Integration Name: Service Logistics Receive Parts):

This integration occurs when a field service technician receives a part order or replenishment in Oracle Field Service.

Note: You must complete the post-order integration steps processing before launching the Receive Parts integration in OFSC. See the Post-order Integration Steps section.

- It marks the corresponding transfer order or purchase order as received.
- It also updates Fusion Inventory.

ORACLE

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.

Order Parts to Replenish Trunk Stock (Integration Name: Service Logistics Replenish Parts):

This integration supports scenarios where field service technicians order parts to replenish their trunk stock.

As with the Order Parts integration:

- A transfer order is created in Fusion.
- The integration includes the same configuration requirements for `preparerid` and `deliverToRequesterId`, but across three mapping nodes:
  - `CreateSupplyRequest1`
  - `CreateSupplyRequest2`
  - `CreateSupplyRequest3`

These nodes must be updated with the `person_id` of the Fusion user responsible for processing the transactions. Transfer orders created with this integration are always marked as Ready To Receive, and after the post-order integrations steps are completed, the field service agent can receive them in OFSC. See the Post-order Integration Steps section.

Replenish parts orders integration between Service Logistics and Field Service occurs as follows:

1. Field service technicians choose the **Van Replenishment** option 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

ORACLE

- Quantity
- Unit of Measure
- Ship To Address Type (Technician or Customer)

6. 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.
7. The following is downloaded to OFSC:

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

> **Note:** Field service technicians can order more than one part number and more than one quantity of the part.

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. The batch program is an OIC integration program that can be run on demand or scheduled to run from OIC.
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.

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.

ORACLE

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.

   ○ If there's no reference data, the application retrieves current asset details information by calling an SCM REST service (installBaseAssets).

   ○ 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.

   ○ 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.

   ○ 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.

   ○ 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.

   ○ 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.

   ○ 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.

   ○ 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.
      - Work Order Integration Code, Work Order Status Code: From the corresponding input parameter.
      - Work Order Type Id: Retrieved at the beginning of the processing based on the input provided as work order typeCd.
      - Work Order 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.

   ○ 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:

ORACLE

- 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.
  - o 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.
  - o 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:

   - o Stops the processing for that maintenance work order.

   - o Logs the message.

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

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 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:

   - o A common REST Service (`profileValues`) is called to get the parent node for the resource from profile **Default Parent Resource Name**.
   - o 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:

   - o Parent Resource

   - o Person Party ID

   - o Full Name

   - o Email

   - o Mobile Phone Number

   - o 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.

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 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)
   - 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 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

ORACLE

    ○ Primary Unit of Measure

Purchase Order and Transfer Order Processing (Integration Name: Oracle Service Logistics OFS Order Status) :

This is a scheduled integration used to update the status of transfer orders and purchase requisitions to Ready to Receive for field service technicians. This scheduled integration is only needed for the order parts and not the replenish parts flow.

- It assumes that:
  - ○ Transfer orders have been shipped from inventory.
  - ○ Purchase requisitions have been approved.

Also, if the parameter `Auto_receive_Purchase_Order` is set to **Y**, the integration can automatically receive purchase orders that are already associated with an ASN (Advanced Shipping Notice).

A field service technician orders parts, which are converted into a transfer order or purchase order based on the item type setup. The steps used to track the purchase order and transfer order from the Fusion applications are performed by this integration, which makes them ready to receive:

1. This is a scheduled integration, which should be run on demand or on a schedule depending on your needs. It uses the Auto Receive Purchase Order parameter. This parameter is set to No by default. If the parameter is set to Yes, it receives the items for purchase orders that have advanced shipment notifications (ASN) information available.
2. It finds all open activities and their associated order parts for each field service technician, and starts processing them one at a time (Calls the Field Service REST Activities).
3. For each of the activities, it tries to find if a transfer order exists and then finds the corresponding order details to gather information on the order type and current order status (Calls the SCM REST transferOrders).
4. If a transfer order is created against the activity and if the order is already fulfilled from inventory and shipped, it updates the order in Field Service to Ready to Receive (Calls the Field Service REST plug-in UpdateInventory).
5. If a transfer order isn't created against the activity it checks for a corresponding purchase order (Calls the REST API PurchaseRequsitions.
6. If a purchase order is created (returned by the previous API), it then checks for the associated ASN information attached to the order. If there's an ASN, based on the value set in the parameter (See Step 1), either autoreceive is performed, or the item is marked as Ready to Receive (Similar to Step 4).
7. If the setup is done to allow autoreceive, it performs receiving (calls REST API receivingReceiptRequests) and then updates the status on Field Service as Received. For this scenario, the field service technician doesn't have to perform the receive operation.
8. If the Auto Receive Purchase Order parameter is set to No, and if a purchase order is created with or without an ASN, it's processed in the same way. The order status is updated to Ready to Receive just like a transfer order is handled. The field service technician will receive it as a final step.

Post-order Integration Steps

You must process the post-order integration steps before launching the Receive Parts integration in OFSC.

- Transfer orders must be shipped from Inventory to be eligible for receiving.
- Purchase requisitions must go through the approval process. The steps must be completed for the order status integration to update the status and allow technicians to receive the items in OFSC.

ORACLE

Related Topics

- About the Field Service Integration Recipe
- Before You Install the Recipe
- Install and Configure the Recipe

ORACLE