Oracle
Sales Cloud
Integrating with Oracle Service Cloud

Release 12
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>i</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>About This Guide</td>
<td>1</td>
</tr>
<tr>
<td>Audience and Scope</td>
<td>1</td>
</tr>
<tr>
<td>Related Guides</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>Overview of Oracle Sales Cloud and Oracle Service Cloud Integration</td>
<td>3</td>
</tr>
<tr>
<td>Integration Component Architecture</td>
<td>3</td>
</tr>
<tr>
<td>Ensuring Matching Geographies</td>
<td>5</td>
</tr>
<tr>
<td>About Integration Process Flows</td>
<td>5</td>
</tr>
<tr>
<td>Overview of Echo Suppression and Bi-Directional Synchronization</td>
<td>7</td>
</tr>
<tr>
<td>Required Files for the Integration</td>
<td>7</td>
</tr>
<tr>
<td>Software Requirements for Integrating Oracle Sales Cloud with Oracle Service Cloud</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Oracle Sales Cloud Configuration</td>
<td>9</td>
</tr>
<tr>
<td>Prerequisites for Integrating Oracle Sales Cloud with Oracle Service Cloud</td>
<td>9</td>
</tr>
<tr>
<td>Enabling Security</td>
<td>11</td>
</tr>
<tr>
<td>Creating an Integration User Account</td>
<td>12</td>
</tr>
<tr>
<td>Assigning Integration Roles</td>
<td>13</td>
</tr>
<tr>
<td>Requesting a Credential Store Framework (CSF) Key for the R12 Oracle Sales Cloud Release</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Oracle Service Cloud Configuration</td>
<td>15</td>
</tr>
<tr>
<td>Prerequisites for Integrating Oracle Service Cloud with Oracle Sales Cloud</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Oracle Integration Cloud Service Configuration</td>
<td>19</td>
</tr>
<tr>
<td>Importing the ICS Integration Flows</td>
<td>19</td>
</tr>
<tr>
<td>Configuring Integration Connections</td>
<td>19</td>
</tr>
<tr>
<td>Activating the Integration Package</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Field Mapping</td>
<td>23</td>
</tr>
<tr>
<td>About Field Mapping</td>
<td>23</td>
</tr>
</tbody>
</table>
# 7 Customizing the Integration

- Customizing the CountryWithProvince ICS Lookup 27
- Adding Validation Rules to a Field 27

# 8 Appendix: Required Files

- Overview of Required Files 31

# 9 Appendix: Performing Bulk Import

- Overview of the Bulk Import Process 35
- Overview of Bulk Import of Account Records from Oracle Sales Cloud to Oracle Service Cloud 35
- Overview of Bulk Import of Contact Records from Oracle Sales Cloud to Oracle Service Cloud 35
- About the Account and Contact Bulk Import Process 36
- Exporting Country and State or Province Mapping Files from Oracle Service Cloud 36
- Performing Bulk Import of Account Records from Oracle Sales Cloud to Oracle Service Cloud 37
- Performing Bulk Import of Contacts Records from Oracle Sales Cloud 42
- Adding Columns to the PersonProfile.csv File 48
- Creating an Import Template for Oracle Service Cloud 50
- Process of Importing Merged Files into Oracle Service Cloud 51
- Performing Bulk Export then Import of Organizations from Oracle Service Cloud to Oracle Sales Cloud 52
- Creating and Running a New Custom Report in Oracle Service Cloud 53
- Performing Post-Processing of the Data File 54
- Importing Post-Processed Organization Data into Oracle Sales Cloud 56
- Importing Account PartyID Values into Oracle Service Cloud 57
- Overview of Performing Bulk Export then Import of Contacts from Oracle Service Cloud to Oracle Sales Cloud 58
- Creating a New Custom Report in Oracle Sales Cloud 59
- Performing Post-Processing of the Contacts Data File 60
- Importing Post-Processed Contacts Data into Oracle Sales Cloud 62
- Importing Contact PartyID Values into Oracle Service Cloud 64
Preface

This preface introduces information sources that can help you use the application.

Oracle Applications Help

Use the help icon to access Oracle Applications Help in the application. If you don’t see any help icons on your page, click the Show Help icon in the global header. Not all pages have help icons. You can also access Oracle Applications Help at https://fusionhelp.oracle.com.

Using Applications Help

Watch: This video tutorial shows you how to find help and use help features.

Additional Resources

- **Community:** Use Oracle Applications Customer Connect to get information from experts at Oracle, the partner community, and other users.
- **Guides and Videos:** Go to the Oracle Help Center to find guides and videos.
- **Training:** Take courses on Oracle Cloud from Oracle University.

Documentation Accessibility

For information about Oracle’s commitment to accessibility, see the Oracle Accessibility Program.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send e-mail to: oracle_fusion_applications_help_ww_grp@oracle.com.
1 About This Guide

Audience and Scope

This guide is intended for anyone who is involved in integrating Oracle Service Cloud with Oracle Sales Cloud. You must perform the integration steps in this guide to integrate Oracle Service Cloud with Oracle Sales Cloud.

If you want to set up and work with the additional features of Oracle Sales Cloud, see Oracle Sales Cloud documentation on Oracle Help Center at https://docs.oracle.com.

Related Guides

You can refer to the following related guides to understand more about the integration tasks covered in this guide.

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Sales Cloud Getting Started with Your Implementation</td>
<td>Describes how to set up a sales automation solution in Oracle Sales Cloud using a case study to describe concepts and procedures.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Implementing Sales</td>
<td>Describes how to configure and set up Sales.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Getting Started with Oracle Sales Cloud Customizations</td>
<td>Introduces you to simple, common customizations of Oracle Sales Cloud using tools, such as Application Composer and Page Composer.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Customizing Sales</td>
<td>Describes how to use tools to customize and extend Oracle Sales Cloud.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Understanding File-Based Data Import and Export</td>
<td>Describes how to import legacy and other data into Oracle Sales Cloud using File-Based Data Import.</td>
</tr>
</tbody>
</table>

Related Topics

- Oracle Sales Cloud Documentation
- Oracle Service Cloud Documentation
2 Introduction

Overview of Oracle Sales Cloud and Oracle Service Cloud Integration

This document outlines the implementation and configuration steps required to integrate create and update processes on accounts and contacts and association processes between accounts and contacts in Oracle Sales Cloud with create and update processes of organizations, and association processes between contacts and organizations in Oracle Service Cloud.

The integration is designed to support customers who want to take advantage of the latest capabilities of the Oracle Sales Cloud application, and use their existing investment in Oracle Service Cloud (RightNow). In the integration, accounts and contacts from Oracle Sales Cloud can be synchronized with organizations and contacts in Oracle Service Cloud. This synchronization is executed using Oracle Integration Cloud Service as the backbone for mapping and information exchange.

This integration document is designed to be used as a template. This document is a starting point that shows how Oracle Sales Cloud and Oracle Service Cloud can be connected to create a value-added business process and user experience. You must enter the documented configurations and install the documented patches to create the integration.

However, it is not a turnkey solution. Each implementation of Oracle Sales Cloud and Oracle Service Cloud is unique, and each customer has different needs that have led them to implement application customizations that support their unique business requirements. While the steps in this document describe how to connect a noncustomized Oracle Sales Cloud instance to a noncustomized Oracle Service Cloud instance, they can be combined with customizations that have already been applied to each instance.

This Oracle Integration Cloud Service (ICS)-based solution enables subscribers to integrate Oracle Sales Cloud and Oracle Service Cloud cloud services.

For more information about Oracle Integration Cloud Service, see: http://docs.oracle.com/cloud/latest/intcs_gs/index.html

Integration Component Architecture

This document outlines the bidirectional integration between Oracle Sales Cloud and Oracle Service Cloud Integration. Account and Contact managements are the primary use cases handled in the Oracle Sales Cloud and Oracle Service Cloud Bi-Directional integration. Account Management consists of account creation and updates. Contact Management consists of contact creation and updates. This integration does not cover account and contact delete operations.

Bi-Directional integration forces various integration architectures such as guaranteed delivery. Oracle Sales Cloud and Oracle Service Cloud Bi-Directional integration introduced Integration Cloud Service (ICS) in the integration component. ICS is a complete, secure, but lightweight integration solution that enables you to connect your applications in the cloud. It simplifies connectivity between your applications, and can connect both your applications that exist in the cloud and your applications that are still maintained on-premise.

This integration manages error handling and guaranteed delivery by introducing concrete fault handling and prevention measures in the integration layer that are realized through ICS. The integration domain covers typical elements and integration functionality, such as adapters for connectivity to back-end systems, routing, transformation, and filtering.
The following figure shows the components of the Oracle Sales Cloud and Oracle Service Cloud integration.

Oracle Sales Cloud Integration Services

The following Oracle Sales Cloud web services are used in the integration:

- **Account Service.** Use this web service to manage accounts in Oracle Sales Cloud.
- **Contact Service.** Use this web service to manage contacts in Oracle Sales Cloud.
Oracle Service Cloud Integration Services

Oracle Service Cloud uses one polymorphic web service which handles all objects including those objects the integration uses: the Organization and the Contact objects.

Ensuring Matching Geographies

Geographies refer to countries, states and provinces, and the way in which each system stores these values. For the integration to function successfully, both Oracle Sales Cloud and Oracle Service Cloud must have matching geographies. In other words, countries, states, and provinces defined in one application must match those of the other application. For example, if one application uses two letter codes to define the state or province, the other application must use the same format rather than using full names or a different code.

For more information about managing geographies, see the Setting Up Geographies topic in Implementing Sales on Oracle Help Center: http://docs.oracle.com/cloud/latest/salescs_gs/OASAL/toc.htm

About Integration Process Flows

The integration supports the following process flows:

Synchronizing Accounts Between Oracle Service Cloud and Oracle Sales Cloud

When synchronizing accounts from Oracle Service Cloud to Oracle Sales Cloud you must be aware of the following differences between the two data models: In Oracle Sales Cloud, accounts can be designated as Prospects or Customers, with Prospect being the default type. In Oracle Service Cloud Prospect does not exist, only Customer. So, when an organization is created in Oracle Service Cloud (with or without an address) the data, when synchronized, is sent to Oracle Sales Cloud as is. The default type that the organization record maps to in Oracle Sales Cloud is Prospect. You can, however, customize the integration flow to display the Customer type by default, if required.

The prebuilt integration package works in the following way:

- When a new organization is created in Oracle Service Cloud, the account is synchronized to Oracle Sales Cloud, and the type is set to Prospect.
- If an Oracle Service Cloud organization does not contain an address and is synchronized to Oracle Sales Cloud, it sets to type Prospect.

Synchronizing Primary Account Details with a Contact Integration Flow

The Oracle Sales Cloud to Oracle Service Cloud integration supports synchronizing account and contact relationships. However, you must consider certain constraints when performing setup. In Oracle Sales Cloud a contact can be linked to multiple accounts while in Oracle Service Cloud a contact can only be linked to one account (Organization). Due to this difference only the contact’s primary account relationship is synchronized between Oracle Sales Cloud and Oracle Service Cloud.
This means that the primary account on the Oracle Sales Cloud Contact Details page will, following synchronization, match
the organization on the Oracle Service Cloud Contact Details page. Additionally, a contact only appears under an Oracle
Service Cloud organization’s list of contacts if that organization is set as the primary account for the contact in Oracle Sales
Cloud.

Though both applications support having Accounts (Organizations) linked to multiple contacts, due to the previously
described restriction, only those contacts which have the account set as primary are synchronized to Oracle Service Cloud.

**Examples of Synchronizing the Primary Account for a Contact.**

Three contact records under the names John Smith, Jane Smith, and Mary Jones are listed under two companies in Oracle
Sales Cloud.

- John Smith has Smith Inc. listed as the primary account in Oracle Sales Cloud. Following synchronization to Oracle
Service Cloud Smith Inc. shows up as John Smith’s Organization. Jane Smith has Smith Inc. listed as the primary
account in Oracle Sales Cloud. Following synchronization to Oracle Service Cloud Smith Inc. appears as Jane
Smith’s Organization. Smith Inc. shows both John and Jane as contacts in both Oracle Sales Cloud and Oracle
Service Cloud.

  Jane Smith has Smith Inc. listed as the primary account in Oracle Sales Cloud. Following synchronization to Oracle
Service Cloud Smith Inc. appears as Jane Smith’s Organization.

  Smith Inc. shows both John and Jane as contacts in both Oracle Sales Cloud and Oracle Service Cloud.

- Mary Jones has Jones Inc. listed as the primary account in Oracle Sales Cloud. In Oracle Service Cloud, Jones Inc.
appears as Mary Jones’ organization.

  In Oracle Sales Cloud, Smith Inc can have Mary Jones set as primary Contact (thus creating a relationship). However
Mary Jones can have another account, Jones Inc. set as the primary account (which is another relationship). In this
case, only the Mary Jones to Jones Inc. relationship is synchronized. So, following synchronization, Oracle Service
Cloud does not show Mary Jones as a contact for Smith Inc. Mary Jones appears as a contact for Jones Inc.

**Viewing, Creating and Updating Incidents in Oracle Sales Cloud**

This process enables a user (generally a salesperson) to create a new service incident on behalf of a customer. Incidents are
routed to a support agent for assessment, and support agents, using Oracle Service Cloud, manage incidents. To create
a new incident in Oracle Service Cloud, a user requires contact information, so contact information is also sent over from
Oracle Sales cloud at the time the incident is logged. The service agent then uses the contact information to first create a new
contact (if required) and then create a service incident.

> **Note:** Account information is not required to create an incident in Oracle Service Cloud.

The following is true when creating service incidents:

- Service incidents can be created from stand-alone contacts, they need not be associated with an account.
- Users cannot customize the service incident form.
- Identity propagation must be handled in the Oracle Service Cloud web service invocations.

When a user clicks the Service Incidents subtab a list of incidents is displayed. If the user clicks the Create Incident button,
the Create Incident form is displayed. Of the required values, Account is prepopulated with the account that is associated
with the incident. After the user enters the required information and clicks the Save and Close button, the incident is created
and committed to the database. If the user clicks the Save and Continue button, the incident is created and saved to Oracle
Service Cloud, but the Edit Incident form is then displayed enabling the user to enter further information.
When the user clicks the Save button the incident is committed to Oracle Service Cloud, and the Edit Incident form remains open. Clicking the Save and Close button closes the Edit Incident form, and returns the user to the List Incident view. The Edit Incident form also displays all existing messages. When a user clicks the Messages subtab all messages related to a chosen incident are displayed. The messages are displayed by Response, Internal Note and Customer Entry, and are sorted in descending order from the most recent. Messages are read-only and cannot be edited.

Overview of Echo Suppression and Bi-Directional Synchronization

During bidirectional account and contact synchronization echoes are generated. This means that when an event is triggered in Oracle Sales Cloud it is synchronized through ICS to Oracle Service Cloud which then fires an event in Oracle Service Cloud and then back to Oracle Sales Cloud, on and on. The ICS-based integration uses an echo suppression mechanism which stops unwanted update or create events (the echoes) from going back to the source application.

Required Files for the Integration

See the Required Files appendix for a detailed list of files that are required to perform the integration. To access the required files, see Integrating Oracle Sales Cloud with Oracle Service Cloud (Doc ID 2059792.1) on My Oracle Support. Under Attachments, select the For Release 12 implementations attachment.

Save the OSC-SVC.zip file. In this folder are the following sub folders:

Oracle Sales Cloud to Oracle Service Cloud Integration Flow Package. This is the integration package which you are deploying in ICS. It contains all integration logic for each bidirectional flow (account/contact create/update).

• ICS/OSC_SVC.par

Customization Scripts for the Integration. Use these scripts to ensure that addresses coming from Oracle Service Cloud are properly stored, in the same format, in Oracle Sales Cloud.

• (Linux) Scripts/ICS_Customization/Linux/getCountriesWithProvince.sh
• (Windows) Scripts/ICS_Customization/Windows/getCountriesWithProvince.ps1

Reference Implementation of Bulk Loading Post Processing Script. These scripts serve as reference implementation which you use to process the exported data from Oracle Sales Cloud during bulk import. These scripts ensure that the combined data is compatible for import into Oracle Service Cloud. Similarly, there are reference scripts to process the data exported from Oracle Service Cloud and transform it to a format compatible for import into Oracle Sales Cloud.

Bulk Loading from Oracle Sales Cloud to Oracle Service Cloud:

• Bulk Loading of Organization data: Scripts/Bulk_Loading/Account/SVC_OSC/StateProvinceMapping.sh
• Bulk loading of Contact data:
  o Scripts/Bulk_Loading/Contact/SVC_OSC/ContactSSR.sh
  o Scripts/Bulk_Loading/Contact/SVC_OSC/StateProvinceMapping.sh

Bulk Loading from Oracle Service Cloud to Oracle Sales Cloud:

• Bulk Loading of Account data: Scripts/Bulk_Loading/Account/SVC_OSC/Account.sh
Bulk Loading of Contact data: Scripts/Bulk_Loading/Contact/SVC_OSC/Contact.sh

Reports. These XML files, located in the Scripts/Bulk_Loading/Reports folder, are used to create reports in Oracle Service Cloud to extract data as required for the bulk loading process.

Software Requirements for Integrating Oracle Sales Cloud with Oracle Service Cloud

This topic lists the software requirements for implementing the Oracle Sales Cloud and Oracle Service Cloud integration using Oracle Integration Cloud Service.

The software requirements are as follows:

- **Software Requirements for Oracle Sales Cloud**: The integration is designed to work with Oracle Sales Cloud (OSC) Release 12. For the list of supported web browsers for Oracle Sales Cloud, see: http://www.oracle.com/us/products/system-requirements/overview/index.html

- **Software Requirements for Oracle Service Cloud**: The integration is designed to work with Oracle Service Cloud (SVC) Release 15.8. For all documentation related to the latest Oracle Service Cloud release, see:https://cx.rightnow.com/app/answers/detail/a_id/7969. Within this page, see the Oracle Service Cloud User Guide: https://documentation.custhelp.com/euf/assets/docs/august2015/olh/wwhelp/wwhimpl/js/html/wwhelp.htm

- **Software Requirements for Oracle Integration Cloud Service**: The integration is designed to work with Oracle Integration Cloud Service (ICS) version 15.3.5 or later.
3 Oracle Sales Cloud Configuration

Prerequisites for Integrating Oracle Sales Cloud with Oracle Service Cloud

The integration stores external reference records in Oracle Service Cloud for accounts and contacts which have been synchronized to Oracle Service Cloud. To enable this functionality, you must first perform the following prerequisite tasks.

Note: To use the Service Incident UI you must first log a service request with help desk to configure a -Dhttps setting for your proxy host, and a J SSE SSL connection to enable you to connect to the Oracle Service Cloud instance.

Configuring the Oracle Sales Cloud Instance for Service Incidents

You use Oracle Functional Setup Manager to enter user name and password information for users of the integration.

1. Sign in to Oracle Functional Setup Manager with administrator privileges.
2. Locate and open the following task: Manage Sales Cloud to Service Cloud Integration and enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Enter the URL of the Oracle Service Cloud instance: https://host_name/ cgi-bin/ interface/cfg</td>
</tr>
<tr>
<td>Security Policy</td>
<td>From the drop-down list, select oracle/wss_username_token_over_ssl_client_policy.</td>
</tr>
<tr>
<td>User Name</td>
<td>Enter the user name of the Oracle Service Cloud configuration setting: EVENT_NOTIFICATION_MAPI_USERNAME.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password value of the Oracle Service Cloud configuration setting: EVENT_NOTIFICATION_MAPI_PASSWD.</td>
</tr>
</tbody>
</table>

Enabling the Service Incident Subtab

The Service Incident subtab is not enabled by default, so you must perform the following steps to enable it.

1. Sign in to Oracle Sales Cloud as an administrator.
2. Under Settings and Actions, go to Manage Sandbox, and create a new sandbox.
3. Sign out of Oracle Sales Cloud, then sign back in to Oracle Sales Cloud.
4. Click Navigator, then under Tools, select Application Composer.
5. Select Common from the Application drop-down list.
6. Expand the Contact node in the Standard Objects list under the Object menu.
7. Click Pages.
8. Click the Simplified Pages tab.
10. Under Subtabs Region, click the Hide, Show or Reorder Subtabs icon.
11. Publish the sandbox.
12. Move Service Incidents from the Available Subtabs list to the Selected Subtabs list, and then click Save and Close.
13. Click Navigator, and then under Sales, click the Contacts link, and do the following:
   a. Click an existing contact, or create a new contact record.
   b. Verify that the new Service Incidents subtab appears at the bottom of the subtabs list.

Overview of Service Incident Message Types
The following table displays available Service Incident messages that customers can view.

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Display As</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note</td>
<td>Private Note</td>
</tr>
<tr>
<td>Staff Account</td>
<td>Response</td>
</tr>
<tr>
<td>Customer</td>
<td>Customer Entry</td>
</tr>
<tr>
<td>Customer Proxy</td>
<td>Customer Entry</td>
</tr>
<tr>
<td>Chat</td>
<td>Chat</td>
</tr>
<tr>
<td>Rule Response</td>
<td>Response</td>
</tr>
<tr>
<td>Rule Response Template</td>
<td>Response</td>
</tr>
<tr>
<td>Voice</td>
<td>Voice</td>
</tr>
</tbody>
</table>

Verifying the Source System Record
You first must define a source system for Oracle Service Cloud as Oracle Sales Cloud supports multiple external reference records for each account or contact. As a prerequisite to setting up Oracle Sales Cloud and Oracle Service Cloud for ICS-based integration, you must verify the source system record in Oracle Sales Cloud.

1. Sign in to Oracle Sales Cloud with administrator privileges.
2. Click Navigator, select Setup and Maintenance, and then search for the following task: Manage Trading Community Source Systems, and click the task link in the results list.
3. In the Manage Trading Community Source Systems screen, select Starts with from the drop-down list beside Code, and enter RNOW in the text field, and then click Search.
4. In the Search Results field, verify that RNOW has appeared and that the Enable for Trading Community Members check box is selected.
5. If the Enable for Trading Community Members check box is not selected, do the following:
   a. Select the row with the code value of RNOW and click the Edit icon.
b. In the Edit Source System: RightNow Service Cloud screen, check the box corresponding to Enable for Trading Community Members and click **Save and Close**.

If Necessary: Creating the Source System Record in Oracle Sales Cloud

If RNOW is not listed in the Trading Community Source System list, you must create it using the following task before proceeding any further.

1. Sign in to Oracle Sales Cloud with administrator privileges.
2. Click Navigator, then select Setup and Maintenance, and in the Search view, enter **Manage Trading Community Source System**, and then click the task link in the results list.
3. On the Manage Trading Community Source Systems page, select **Starts with** from the drop-down list beside Code, and enter **RNOW** in the text field, and then click **Search**.
4. Click the **Create** icon under Search Results.
5. On the Create Source System page, fill in the values from the following table, and when you are finished, select the box beside **Enable Trading Community Members** and then click **Save and Close**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>RNOW.</td>
</tr>
<tr>
<td>Name</td>
<td>RightNow Cloud Service.</td>
</tr>
<tr>
<td>Description</td>
<td>Maintains cross reference between the Oracle Fusion applications database and records imported using comma-separated files.</td>
</tr>
</tbody>
</table>

Verifying Source System Entities

Use the following task to verify source system entities.

1. Sign in to Oracle Sales Cloud with administrator privileges.
2. Click Navigator, and then select Setup and Maintenance.
3. In the Search view, enter **Manage Source System Entities**, and then click the task link in the results list.
4. In the Manage Source System Entities page, select **RightNow Service** on the Source Systems for Trading Community Members list and ensure the following check boxes are selected: **Address, Contact Points, Parties**.

Enabling the Trading Community Events Profile Option

Now you must enable the Trading Community Events profile option using the following task.

1. Sign in to Oracle Sales Cloud with administrator privileges.
2. Click Navigator, and then select Setup and Maintenance.
3. In the Search view, enter **Manage Trading Community Common Profile Options**, then click the task link in the results list.
4. In the Manage Trading Community Common Profile Options Overview page, select the following profile option: **HZ_ENABLE_EVENT_TRACKING**.
5. In the **HZ_ENABLE_EVENT_TRACKING: Profile Values** page, set the Site level value to **Yes**, and then click **Save and Close**.
6. In the Manage Trading Community Common Profile Options Overview page, select the following profile option: **HZ_INVOKE_OBJ_WF_ON_TRACKING**.
7. In the **HZ_INVOKE_OBJ_WF_ON_TRACKING: Profile Values** page, set the Site level value to **Yes**, then click **Save and Close**.
Enabling Security

You can configure your ICS instance to use the **Username Password Token** security policy which allows secure access to the Oracle Sales Cloud instance. The Username Password Token policy requires login credentials to enable access to resources on Oracle Sales Cloud.

Your Oracle Sales Cloud instance exposes the Oracle Service Cloud service catalog and event catalog to ICS and those resources are secured in Oracle Sales Cloud. You must create an Integration User ID called FUSION_APPS_ICS_APPID and assign the required roles and privileges to the user. The integration user FUSION_APPS_ICS_APPID must have the following roles and privileges to access the protected resources:

- ALL_INTEGRATION_POINTS_ALL_DATA
- FND_MANAGE_CATALOG_SERVICE_PRIV
- SOA Operator

Creating an Integration User Account

To initiate the Oracle Sales Cloud Service Catalog or Event Catalog web services from ICS, Oracle recommends that you create a unique user called the Integration User Account user. Use the following procedure to create the new user.

1. Sign in to Oracle Sales Cloud using system administrator privileges.
2. Click Navigator, and then select Setup and Maintenance.
3. In the Search view, enter **Manage Users**, and then click the task link in the results list.
4. Complete the following fields:
   - **Last Name**: FUSION_APPS_ICS_APPID
   - **Email**: Enter a valid e-mail address
   - **Hire Date**: Enter the current date
   - **User Name**: FUSION_APPS_ICS_APPID
   - **Send user name and password**: Select the check box
   - **Person Type**: Employee
   - **Legal Employer**: Select a valid legal organization from the drop-down list.
   - **Business Unit**: Select a valid business unit from the drop-down list.
5. Click **Save and Close**.
   An e-mail is sent to the e-mail address after the user has been created.
6. Sign out of Oracle Sales Cloud, then sign back in to the Oracle Sales Cloud instance using the FUSION_APPS_ICS_APPID user name and the temporary password provided in the notification e-mail.
7. Change the password when prompted at the first login.
   The Oracle Sales Cloud Welcome page appears.
8. Sign out from Oracle Sales Cloud.
Assigning Integration Roles

Now you must assign users with the following roles and privileges using Oracle Security Console.

⚠️ Note: You must have privileges sufficient to create new roles, such as IT Security Manager.

1. In Oracle Sales Cloud, click Tools, then select Security Console.
2. Click the Create Role button in the top right corner of the screen.
3. In the Create Role: Basic Information page, create a new record using the information from the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Name</td>
<td>ICS Integration Role</td>
</tr>
<tr>
<td>Role Code</td>
<td>INT_ICS_Integration_Role</td>
</tr>
<tr>
<td>Role Category</td>
<td>CRM - Job Roles</td>
</tr>
<tr>
<td>Description</td>
<td>Custom Role for Accessing OSC Services Catalog</td>
</tr>
</tbody>
</table>

4. Click Next.
7. Click Next.
9. On the Create Role: Role Hierarchy page, do the following:
   a. Click Add role.
   b. In the Search field, enter Sales Admin.
   c. Select Sales Administrator, and then click Add Role Membership.
   d. Close the Add Role Membership window.
10. In the Create Role: Role Hierarchy page, do the following:
    a. Click Add role.
    b. In the Search field, enter SOA operator.
    c. Select the SOA operator role, and then click Add Role Membership.
    d. Close the Add Role Membership window.
11. Click Next.
12. In the Create Role: Users page, click Add user and then do the following:
    a. In the Search field, enter FUSION_APPSICS_APPID.
    b. Choose the necessary user, then select Add user to Role.
    c. Close the Add user window.
13. Click Next, then on the Summary and Impact page, review the details, and click Save and Close.
Requesting a Credential Store Framework (CSF) Key for the R12 Oracle Sales Cloud Release

To use the ICS-based integration of Oracle Sales Cloud and Oracle Service Cloud you must first log a service request with Oracle to create a CSF key which stores credentials which enable Oracle Sales Cloud to access ICS.

When creating the service request, include in the following in the subject line: Create the CSF key for the ICS-based OSC-SVC integration. Additionally, provide your Identity domain information (such as: icssvc.identity.domain= idm2152) for both your ICS and Oracle Sales Cloud subscriptions.

After you submit the service request, Oracle will contact you to obtain the user name and password used for accessing your ICS instance.
4 Oracle Service Cloud Configuration

Prerequisites for Integrating Oracle Service Cloud with Oracle Sales Cloud

The integration between Oracle Service Cloud and Oracle Sales Cloud requires that the following prerequisite tasks be completed.

Verifying Functionality to Publish Business Events

You first verify that you can publish business events.

1. Sign in to the Oracle Service Cloud application as a user with administrator privileges.
2. On the Navigation pane, click Configuration, Site Configuration, and then Configuration Settings.
3. In the Configuration Base field select the Site check box.
4. In the Key field enter a wildcard string such as "EVENT%." Note that this searches for and return all strings beginning with EVENT.
5. Click Search.

If several configuration parameters beginning with EVENT_ are displayed, such as, EVENT_NOTIFICATION_ENABLED, then the functionality to publish events is available in your version of Oracle Service Cloud.

Displaying All Data Synchronization Parameters and Enabling Data Synchronization Functionality

Use this task to enable data synchronization functionality.

1. In the search results page displayed in the previous task, verify that the following six event configuration parameters are present:
   - EVENT_NOTIFICATION_ENABLED
   - EVENT_NOTIFICATION_MAPI_SEC_IP_RANGE
   - EVENT_NOTIFICATION_MAPI_ICSUSER
   - EVENT_NOTIFICATION_MAPI_PASSWD
   - EVENT_NOTIFICATION_SUBSCRIBER_USERNAME
   - EVENT_NOTIFICATION_SUBSCRIBER_PASSWD
2. Select EVENT_NOTIFICATION_ENABLED.

   This is a Boolean parameter. It is the global switch that controls whether or not business events from Oracle Service Cloud are published to external applications. The default value is No, which means that events are not published to external applications.
3. Set the value to Yes if it is not already set.
4. Click Save & Close.
Optional: Setting the IP Range for Incoming Messages

By default, the configuration parameter EVENT_NOTIFICATION_MAPI_SEC_IP_RANGE is null (empty). This parameter enables incoming messages to be honored only if originating from a specific IP addresses. This is an optional parameter. If left unpopulated, no IP restrictions are enforced and requests from any IP address is honored.

If the parameter has even one value, in other words, if it is not null, only requests from the specified IP address are accepted.

If it is necessary to limit the address or addresses from which requests should be accepted, enter the addresses as a comma-separated list.

1. From the list of configuration parameters, click **EVENT_NOTIFICATION_MAPI_SEC_IP_RANGE**.
2. Enter the list of IP addresses as comma-separated values.
   For example, 121.110.54.12, 135.87.76.45 and so on.
3. Click **Save & Close**.

Setting Credentials for Incoming Requests

Two configuration parameters, EVENT_NOTIFICATION_MAPI_USERNAME and EVENT_NOTIFICATION_MAPI_PASSWD store the credentials used by the external applications when invoking RightNow web services for either subscription requests or transactional requests. As neither parameter has a default value, you must specify a value for each.

The EVENT_NOTIFICATION_MAPI_USERNAME configuration parameter stores the UserID that is specified on the header of incoming request.

The EVENT_NOTIFICATION_MAPI_PASWD parameter stores the password associated with the UserID specified in incoming request.

>Note: The password is stored in an encrypted format for security purposes.

1. From the list of configuration parameters, click **EVENT_NOTIFICATION_MAPI_USERNAME**.
2. Specify the user name and save the changes.
3. From the list of configuration parameters, click **EVENT_NOTIFICATION_MAPI_PASSWD**.
4. Specify the password and save the changes.

Setting Credentials for Outgoing Requests

Two configuration parameters, EVENT_NOTIFICATION_SUBSCRIBER_USERNAME and EVENT_NOTIFICATION_SUBSCRIBER_PASSWD store the credentials to be used by Oracle Service Cloud when sending event notifications to the external subscriber. As neither parameter has a default value, you must specify a value for each.

The EVENT_NOTIFICATION_SUBSCRIBER_USERNAME configuration parameter stores the UserID that should be used on the event notification message sent to the external application that has subscribed to the event.

The EVENT_NOTIFICATION_SUBSCRIBER_PASWD stores the password associated with the UserID specified in the config verb above.

>Note: The password is stored in an encrypted format for security purposes.

1. From the list of configuration parameters, click **EVENT_NOTIFICATION_SUBSCRIBER_USERNAME**.
2. Specify the user name and save the changes.
Note: Make sure the user name you chose is also defined in ICS.

3. From the list of configuration parameters, click `EVENT_NOTIFICATION_SUBSCRIBER_PASSWD`.
4. Specify the password and save the changes.
5 Oracle Integration Cloud Service Configuration

Importing the ICS Integration Flows

The first step when setting up for ICS-based integration is to import the ICS integration flows. To import the required Oracle Sales Cloud and Oracle Service Cloud integration flows you must download the ICS integration flow package to your local computer.

The required ICS integration flow package file is: OSC_SVC.par.

1. Sign in to the ICS instance.
2. On the Welcome page, click the Packages icon.
3. On Package page, click the Import Package button.
4. On Import Package File dialog box, click Browse and then select OSC_SVC.par, then click Import Package.

This creates all the integration flows contained within the package as well as the connections to Oracle Sales Cloud and Oracle Service Cloud.

Configuring Integration Connections

The integration between Oracle Service Cloud and Oracle Sales Cloud requires that the following prerequisite tasks be completed.

Configuring a Connection to the Oracle Sales Cloud Instance

First you configure the connection to the Oracle Sales Cloud instance.

1. On the ICS homepage, click the Connections icon.
2. On the Connections page, make sure that Oracle Sales Cloud appears.
3. Click the Sales Cloud entry to view the Oracle Sales Cloud connection detail page.
4. Click the Configure Connectivity button and in the Connection Properties window, enter the following values:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSC Service Catalog WSDL URL</td>
<td>Enter the service catalog URL for your Oracle Sales Cloud instance.</td>
</tr>
<tr>
<td></td>
<td>For example: https://&lt;common domain host&gt;/fndAppCoreService/ServiceCatalogServiceWSDL</td>
</tr>
<tr>
<td>OSC Event Catalog URL</td>
<td>Enter the event catalog URL for your Oracle Sales Cloud instance. This is an optional field.</td>
</tr>
<tr>
<td></td>
<td>For example: <a href="https://CRM">https://CRM</a> domain host&gt;/soa-infra</td>
</tr>
</tbody>
</table>

5. Click OK.
6. Click the **Configure Credentials** button, and in the Credentials window, enter the following values:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Policy</td>
<td>Username Password Token.</td>
</tr>
<tr>
<td>Username</td>
<td>Enter FUSION_APPS_ICS_APPID.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password.</td>
</tr>
<tr>
<td>Confirm Password</td>
<td>Reenter the password.</td>
</tr>
</tbody>
</table>

7. Click **OK**.
8. Click the **Test** icon at the top right corner of the ICS Connection Configuration page.
9. When the status meter shows 100% Complete, click **Save**.
10. Click the **Exit** button.

### Configuring the Connection to the Oracle Service Cloud Instance

After configuring the connection to the Oracle Sales Cloud instance, you then configure the connection to the Oracle Service Cloud instance.

1. Sign in to the ICS instance.
2. On the Home page, click the **Connections** icon.
3. On the Connections page, make sure that Oracle Service Cloud appears.
4. Click the Service Cloud entry to view the Oracle Service Cloud connection detail page.
5. Click the **Configure Connectivity** button and in the Connection Properties window, enter the following values:
   - **Property Name**: OSC Service Catalog WSDL URL
   - **Property Value**: Enter the Service Catalog URL from your Oracle Service Cloud instance.
     
     For example: https://datasync-162-rel12.qb.lan/cgi-bin/datasync_162_rel12.cfg/services/soap?wsdl=typed_v1.3

6. Click **OK**.
7. Click the **Configure Credentials** button, and in the Credentials window, enter the following values:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Policy</td>
<td>Username Password Token.</td>
</tr>
<tr>
<td>Username</td>
<td>Enter the user name.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password.</td>
</tr>
<tr>
<td>Confirm Password</td>
<td>Confirm the password.</td>
</tr>
</tbody>
</table>
8. Click OK.
9. Click the Test icon at the top right corner of the ICS Connection Configuration page.
10. When the status meter shows 100% Complete, click Save.

Activating the Integration Package

To activate the integration flows you must have successfully imported the OSC_SVC integration package and configured Oracle Sales Cloud and Oracle Service Cloud connections.

1. Sign in to Oracle Integration Cloud Service (ICS) instance.
2. On Home page click Integrations icon.
3. Click the Active button under OSC_SVC_CONTACT_CREATE.
4. Check the Enable detail tracing check box and click the Active button on the confirmation dialog window.
5. Ensure the flow was activated successfully.
6. Repeat the active steps for the following integration flows:
   - OSC_SVC_ACCOUNT_CREATED
   - OSC_SVC_ACCOUNT_UPDATED
   - OSC_SVC_CONTACT_UPDATED

After activation, verify that the integration synchronization is functional. Do this, for example, by creating or updating an account or contact record in the Oracle Sales Cloud. This action should automatically sync the record to Oracle Service Cloud. Additionally, create or update an organization or contact record in the Oracle Service Cloud. This action should automatically synchronize the record to Oracle Sales Cloud.

7. Repeat the activate steps for the following integration flows in the Oracle Service Cloud-to-Oracle Sales Cloud direction.
   - SVC_OSC_ORGANIZATION_CREATED
   - SVC_OSC_ORGANIZATION_UPDATED
   - SVC_OSC_CONTACT_CREATED
   - SVC_OSC_CONTACT_UPDATED

Verify that the flows have been activated by signing in to Oracle Service Cloud and creating and updating organizations and contacts.
# Field Mapping

## About Field Mapping

This chapter details field mapping between Oracle Sales Cloud and Oracle Service Cloud.

## Account Field Mapping

The following table lists the Account field mapping between Oracle Sales Cloud and Oracle Service Cloud.

<table>
<thead>
<tr>
<th>Oracle Service Cloud Attribute</th>
<th>Oracle Service Cloud Data Type</th>
<th>Oracle Sales Cloud Attribute</th>
<th>Oracle Sales Cloud Data Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String (80)</td>
<td>Organization Name</td>
<td>OrganizationName</td>
<td>VARCHAR2 (360) Not applicable.</td>
</tr>
<tr>
<td>Address.Street</td>
<td>String (240)</td>
<td>PrimaryAddress. AddressLine1 + ' ' + PrimaryAddress. AddressLine2 + ' ' + PrimaryAddress. AddressLine3 + ' ' + PrimaryAddress. AddressLine4</td>
<td>VARCHAR2 (240) VARCHAR2 (240) VARCHAR2 (240) VARCHAR2 (240)</td>
<td>if PrimaryAddress. Country != null</td>
</tr>
<tr>
<td>Address.StateOrProvince. Name</td>
<td>String (255)</td>
<td>PrimaryAddress. Province</td>
<td>VARCHAR2 (60)</td>
<td>If PrimaryAddress. Country != null then</td>
</tr>
<tr>
<td>Address.City</td>
<td>String (80)</td>
<td>PrimaryAddress. City</td>
<td>VARCHAR (60)</td>
<td>If PrimaryAddress. Country != null</td>
</tr>
<tr>
<td>Address.PostalCode</td>
<td>String (10)</td>
<td>PrimaryAddress. PostalCode</td>
<td>VARCHAR2 (60)</td>
<td>If PrimaryAddress. Country != null</td>
</tr>
</tbody>
</table>
### Oracle Sales Cloud Attribute | Oracle Service Cloud Attribute | Oracle Sales Cloud Data Type | Oracle Service Cloud Data Type | Condition
---|---|---|---|---
AddressType. Name | AddressType. Name | String (255) | String (255) | Not applicable. Not applicable.

The default value in the UI is Billing.

### Address.Action | Address.Action | Not applicable. | Not applicable. | Not applicable.

The default value in the UI is Add.

---

**Contact Field Mapping**

The following table lists the Account field mapping between Oracle Sales Cloud and Oracle Service Cloud.

<table>
<thead>
<tr>
<th>Oracle Service Cloud Attribute</th>
<th>Oracle Service Cloud Data Type</th>
<th>Oracle Sales Cloud Attribute and Data Type</th>
<th>Oracle Sales Cloud Data Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name.First</td>
<td>PersonName. First</td>
<td>FirstName</td>
<td>VARCHAR2 (150)</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Name.Last</td>
<td>PersonName. Last</td>
<td>LastName</td>
<td>VARCHAR2 (150)</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Address.Street</td>
<td>Address.Street</td>
<td>PrimaryAddress. AddressLine1 + &quot;,&quot; + PrimaryAddress. AddressLine2 + &quot;,&quot; + PrimaryAddress. AddressLine3 + &quot;,&quot; + PrimaryAddress. AddressLine4</td>
<td>VARCHAR2 (240)</td>
<td>if PrimaryAddress. Country != null</td>
</tr>
<tr>
<td>Address.StateOrProvince. Name</td>
<td>Address. StateOrProvince. NamedID.Name</td>
<td>PrimaryAddress. Province</td>
<td>VARCHAR2 (60)</td>
<td>If PrimaryAddress. Country != null</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PrimaryAddress. State</td>
<td>VARCHAR2 (60)</td>
<td>If Province != null then Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Constraints: Oracle Sales Cloud can accept any number of characters for the state field but Oracle Service Cloud accepts only two letters, such as OR, or CA.</td>
<td>If State != null then State</td>
</tr>
<tr>
<td>Oracle Service Cloud Attribute</td>
<td>Oracle Service Cloud Data Type</td>
<td>Oracle Sales Cloud Attribute and Data Type</td>
<td>Oracle Sales Cloud Data Type</td>
<td>Condition</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------------</td>
<td>-------------------------------------------</td>
<td>-----------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Address.City</td>
<td>Address.City String (80)</td>
<td>PrimaryAddress. City VARCHAR (60)</td>
<td>If PrimaryAddress. Country != null</td>
<td></td>
</tr>
<tr>
<td>Emails. EmailList. Address</td>
<td>EmailList. Email. Address String (80)</td>
<td>EmailAddress VARCHAR (320)</td>
<td>if EmailAddress != null</td>
<td></td>
</tr>
<tr>
<td>Emails. EmailList. AddressType.ID</td>
<td>EmailList. Email. AddressType. NamedID.ID Long</td>
<td>Not applicable Not applicable</td>
<td>if EmailAddress != null</td>
<td></td>
</tr>
<tr>
<td>Phones. PhoneList. Number</td>
<td>Phones. PhoneList. Number String (40)</td>
<td>FormattedWorkPhoneNumber VARCHAR2 (40)</td>
<td>if FormattedWorkPhoneNumber != null</td>
<td></td>
</tr>
<tr>
<td>Phones. PhoneList. PhoneNumber</td>
<td>PhoneList. Phone. PhoneNumber String (40)</td>
<td>Not applicable Not applicable</td>
<td>if FormattedWorkPhoneNumber != null</td>
<td></td>
</tr>
<tr>
<td>Phones. PhoneList. PhoneType</td>
<td>PhoneList. Phone. PhoneType. NamedID.ID Long</td>
<td>Not applicable Not applicable</td>
<td>if FormattedWorkPhoneNumber != null</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: The PHONE_COUNTRY_CODE must be valid and can contain up to three characters. Other fields can accept up to maximum length. The FormattedWorkPhoneNumber field can have a maximum size of up to 40 characters and be accepted in Oracle Service Cloud.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: The PHONE_EXTENSION field is not available in the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle Service Cloud Attribute</td>
<td>Oracle Service Cloud Data Type</td>
<td>Oracle Sales Cloud Attribute and Data Type</td>
<td>Oracle Sales Cloud Data Type</td>
<td>Condition</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------</td>
<td>------------------------------------------</td>
<td>-----------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Phones. PhoneList. PhoneType</td>
<td>Long</td>
<td>Not applicable</td>
<td>Long</td>
<td>if FormattedMobile Number != null</td>
</tr>
<tr>
<td>The default value in the UI is 1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phones. PhoneList. Number</td>
<td>Phones. PhoneList. Number</td>
<td>FormattedFaxNumber</td>
<td>VARCHAR2(40)</td>
<td>if FormattedFaxNumber != null = null</td>
</tr>
<tr>
<td></td>
<td>String (40)</td>
<td>Constraints: The same as those of FormattedWorkPhoneNumber</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phones. PhoneList. PhoneType</td>
<td>PhoneList. Phone. PhoneType. NamedId.ID</td>
<td>Not applicable</td>
<td>Long</td>
<td>if FormattedFaxNumber != null = null</td>
</tr>
<tr>
<td>The default value in the UI is 2.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phones. PhoneList. Number</td>
<td>Phones. PhoneList. Number</td>
<td>FormattedHomePhoneNumber</td>
<td>VARCHAR2(40)</td>
<td>if FormattedHome PhoneNumber != null = null</td>
</tr>
<tr>
<td></td>
<td>String (40)</td>
<td>Constraints: The same as those of FormattedWorkPhoneNumber</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VARCHAR2(40)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: The same as those of FormattedWorkPhoneNumber</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phones. PhoneList. PhoneType</td>
<td>PhoneList. Phone. PhoneType. NamedId.ID</td>
<td>Not applicable</td>
<td>Long</td>
<td>if FormattedHome PhoneNumber != null = null</td>
</tr>
<tr>
<td>The default value in the UI is 4.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization. ID</td>
<td>NameId.id</td>
<td>Enrich (Account PartyId)</td>
<td>Long</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>Long</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Reference</td>
<td>Long</td>
<td>PartyId</td>
<td>NUMBER (18)</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>
Customizing the Integration

Customizing the CountryWithProvince ICS Lookup

The Oracle Sales Cloud to Oracle Service Cloud prebuilt integration makes use of a lookup which contains the list of countries configured to store, in Oracle Sales Cloud, Address Region data under the Province field rather than the State field.

If you have customized address formats in Oracle Sales Cloud then this default lookup must be updated to reflect your customizations so that addresses coming from Oracle Service Cloud are properly stored in Oracle Sales Cloud.

1. In your install package, access the following lookup customization script.
   - (Linux) Scripts/ICS_Customization/Linux/getCountriesWithProvince.sh
   - (Windows) Scripts/ICS_Customization/Windows/getCountriesWithProvince.ps1

2. Run the Linux script as follows:
   ```bash
   sh getCountriesWithProvince.sh <CRM domain hostname> FUSION_APPS_ICS_APPID <FUSION_APPS_ICS_APPID password>
   ```

3. Run the Windows script as follows:
   ```powershell
   .\getCountriesWithProvince.ps1 <CRM domain hostname> FUSION_APPS_ICS_APPID <FUSION_APPS_ICS_APPID password>
   ```

4. The script creates a file called: CountryWithProvince.csv.

5. Save this file locally.

6. Sign in to your ICS instance and navigate to the Lookups section.

7. Verify that the CountryWithProvince lookup is present.

8. Click Import Lookup, then do the following:
   - Select the CountryWithProvince.csv file you saved, and then click Import.
   - Click Yes on the Import Confirmation prompt to overwrite the existing lookup.
     If the import was successful, a confirmation message appears.

Adding Validation Rules to a Field

To accommodate data model differences between Oracle Sales Cloud and Oracle Service Cloud, validation rules can be added to Oracle Sales Cloud fields to avoid possible truncation of attributes.

1. Navigate to Navigator, Tools, click Customization, and then select Application Composer.

2. Select the Common application from the Application drop-down list.

3. Under the Objects menu, expand Standard Objects, then expand the Account object for which you want to create a validation rule, and then click Server Scripts.
   In the Server Scripts window, the Validation Rules tab is shown by default.

4. Under Object Rules, click the Add a new validation rule icon.

5. In the Create Object Validation Rule window, create validation rules for the available fields using the information contained in the tables:
### OrganizationName Field Options

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_INT_SVC_OrganizationNameValidation</td>
<td></td>
</tr>
</tbody>
</table>

**Error Message**
The first name has more than 80 characters which is the maximum allowed.

**Definition**

\[
\text{return } (\text{length(OrganizationName)} \leq 80)
\]

---

### Contact FirstName Field Options

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_INT_SVC_FirstNameValidation</td>
<td></td>
</tr>
</tbody>
</table>

**Error Message**
The first name has more than 80 characters, which is the maximum allowed.

**Definition**

\[
\text{return } (\text{length(PersonFirstName)} \leq 80)
\]

---

### Contact LastName Field Options

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_INT_SVC_LastNameValidation</td>
<td></td>
</tr>
</tbody>
</table>

**Error Message**
The last name has more than 80 characters, which is the maximum allowed.

**Definition**

\[
\text{return } (\text{length(PersonLastName)} \leq 80)
\]

---

### EmailAddress Field Options

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_INT_SVC_EmailAddressValidation</td>
<td></td>
</tr>
</tbody>
</table>

**Error Message**
The e-mail address has more than 80 characters, which is the maximum allowed.

**Definition**

\[
\text{return } (\text{length(PrimaryEmailAddress)} \leq 80)
\]

---

### FormattedWorkPhoneNumber Field Options

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_INT_SVC_FormattedWorkPhoneNumberValidation</td>
<td></td>
</tr>
<tr>
<td>Field Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FormattedWorkPhoneNumber Field</td>
<td>Description</td>
</tr>
<tr>
<td>Options</td>
<td>The work phone has more than 40 characters, which is the maximum allowed.</td>
</tr>
</tbody>
</table>

**Definition**

```java
def phones = Phone

while (phones.hasNext()) {
  def phone = phones.next()

  if (phone?.PhoneType == 'WORK' && phone?.PhoneNumber != null &&
      length(phone?.FormattedPhoneNumber) > 40)
    return false
}
return true
```

<table>
<thead>
<tr>
<th>FormattedMobilePhoneNumber Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td>Rule Name</td>
</tr>
<tr>
<td>Options</td>
<td>O_INT_SVC_FormattedMobileNumberValidation</td>
</tr>
<tr>
<td>Error Message</td>
<td>The mobile number has more than 40 characters, which is the maximum allowed.</td>
</tr>
</tbody>
</table>

**Definition**

```java
def phones = Phone

while (phones.hasNext()) {
  def phone = phones.next()

  if (phone?.PhoneType == 'MOBILE' && phone?.PhoneNumber != null &&
      length(phone?.FormattedPhoneNumber) > 40)
    return false
}
return true
```

<table>
<thead>
<tr>
<th>FormattedFaxNumber Field Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td>Rule Name</td>
</tr>
<tr>
<td>Options</td>
<td>O_INT_SVC_FormattedFaxNumberValidation</td>
</tr>
<tr>
<td>Error Message</td>
<td>The fax has more than 40 characters, which is the maximum allowed.</td>
</tr>
</tbody>
</table>

**Definition**

```java
def phones = Phone

while (phones.hasNext()) {
  def phone = phones.next()

  if (phone?.PhoneType == 'FAX' && phone?.PhoneNumber != null &&
      length(phone?.FormattedPhoneNumber) > 40)
    return false
```
### FormattedFaxNumber Field Options

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>return true</td>
</tr>
</tbody>
</table>

### FormattedHomePhoneNumber Field Options

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_INT_SVC_FormattedHomePhoneNumberValidation</td>
<td>The home phone has more than 40 characters, which is the maximum allowed.</td>
</tr>
</tbody>
</table>

#### Definition

```java
def phones = Phone
while (phones.hasNext()) {
    def phone = phones.next()
    if (phone?.PhoneType == 'HOME' && phone?.PhoneNumber != null && length(phone?.FormattedPhoneNumber) > 40)
        return false
}
return true
```

### PostalCode Field Options

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_INT_SVC_PostalCodeValidation</td>
<td>The postal code has more than 10 characters, which is the maximum allowed.</td>
</tr>
</tbody>
</table>

#### Definition

```java
return(length(PrimaryAddressPostalCode) <= 10)
```

6. When you are completed, click **Save and Close**.
# Appendix: Required Files

## Overview of Required Files

The following table lists the files required to perform the Oracle Sales Cloud to Oracle Service Cloud integration.

<table>
<thead>
<tr>
<th>File Description</th>
<th>File Name</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICS Integration Package Archive.</td>
<td>OSC_SVC.par</td>
<td>ICS:\OSC_SVC.par.</td>
</tr>
<tr>
<td>Account Bulk import post processing script.</td>
<td>Account.sh</td>
<td>Scripts\Bulk\Loading\Account \OSC_SVC\Account.sh</td>
</tr>
<tr>
<td>Organization bulk import post processing script.</td>
<td>Mapping.sh</td>
<td>Scripts\Bulk\Loading\Account\SVC_OSC\StateProvinceMapping.sh</td>
</tr>
<tr>
<td>Country With Province .CSV file.</td>
<td>CountryWithProvince. csv</td>
<td>Scripts\Bulk\Loading\Account\SVC_OSC\CountryWithProvince.csv</td>
</tr>
<tr>
<td>Contact Bulk loading post processing script.</td>
<td>Contact.sh</td>
<td>Scripts\Bulk\Loading\Contact \OSC_SVC\Contact.sh</td>
</tr>
<tr>
<td>Prepare Contact Source Reference script.</td>
<td>prepareContactSourceReference.sh</td>
<td>Scripts\Bulk\Loading\Contact\OSC_SVC\prepareContactSourceReference.sh</td>
</tr>
<tr>
<td>Contact Bulk Loading State Province Mapping script.</td>
<td>StateProvinceMapping.sh</td>
<td>Scripts\Bulk\Loading\Contact\SVC_OSC\StateProvinceMapping.sh</td>
</tr>
<tr>
<td>Contact Bulk Loading post processing script.</td>
<td>ContactSSR.sh</td>
<td>Scripts\Bulk\Loading\Contact\SVC_OSC\ContactSSR.sh</td>
</tr>
<tr>
<td>File</td>
<td>Description</td>
<td>File Name</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Service Cloud Contact ID and External Reference Report Definition.</td>
<td>A report definition which is imported into Oracle Service Cloud and used for exporting Contact IDs and their respective External References. Used during bulk loading of Contacts from Oracle Sales Cloud to Oracle Service Cloud.</td>
<td>Contact_ID_ExtReference.xml</td>
</tr>
<tr>
<td>Country and Province(State) ID Report Definition.</td>
<td>A report definition which is imported into Oracle Service Cloud and used for exporting CountryIDs and their respective Province(State) IDs. Used during bulk loading of Accounts and Contacts from Oracle Sales Cloud to Oracle Service Cloud.</td>
<td>Country and Province (State) ID Report.xml</td>
</tr>
<tr>
<td>Country ID Report Definition.</td>
<td>A report definition which is imported into Oracle Service Cloud for exporting Country and their respective Country IDs. Used during bulk loading of Accounts and Contacts from Oracle Sales Cloud to Oracle Service Cloud.</td>
<td>CountryIDMapping. xml</td>
</tr>
<tr>
<td>Service Cloud Organization ID and External Reference Report Definition.</td>
<td>A report definition which is imported into Oracle Service Cloud for exporting Organization IDs and their respective External References. Used during bulk loading of Accounts from Oracle Sales Cloud to Oracle Service Cloud.</td>
<td>Organization_ID_ExtReference.xml</td>
</tr>
<tr>
<td>Linux script to get Countries with Province.</td>
<td>A Linux script to create CountriesWithProvince. csv file. Used to customize the CountryWithProvince ICS Lookup.</td>
<td>getCountriesWithProvince. sh</td>
</tr>
<tr>
<td>Windows script to get Countries with Province.</td>
<td>A Windows script to create the CountriesWithProvince. csv file. Used to customize the CountryWithProvince ICS Lookup.</td>
<td>getCountriesWithProvince. ps1</td>
</tr>
<tr>
<td>Service Cloud Organization Export Reference Report Definition.</td>
<td>A report definition which is imported into Oracle Service Cloud for Exporting Organization data to be imported into Oracle Sales Cloud during bulk loading.</td>
<td>Unsynchronized Orgs_ By_ Create_ Date.xml</td>
</tr>
<tr>
<td>Service Cloud Contact Export Reference Report Definition.</td>
<td>A report definition which is imported into Oracle Service Cloud for exporting contact data</td>
<td>Unsynchronized Contacts_ By_ Create_ Date.xml</td>
</tr>
</tbody>
</table>
### Appendix: Required Files

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>File Name</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>to be imported into Oracle Sales Cloud during bulk loading.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9 Appendix: Performing Bulk Import

Overview of the Bulk Import Process

Bulk import of Contact and Account data is optional depending on your implementation of either Oracle Sales Cloud and Oracle Service Cloud.

- If you are new to both Oracle Sales Cloud and Oracle Service Cloud, you do not need to perform the steps in this appendix.
- If you have an existing implementation of Oracle Sales Cloud (but not Oracle Service Cloud), you must perform the steps shown in this appendix.
- If you have an existing implementation of Oracle Service Cloud (but not Oracle Sales Cloud), you must perform the steps shown in this appendix.
- If you have an existing implementation of both Oracle Sales Cloud and Oracle Service Cloud, you must perform all steps in this chapter.

Overview of Bulk Import of Account Records from Oracle Sales Cloud to Oracle Service Cloud

This topic presents a high-level overview of bulk import of account records.

**Note:** If you must implement both Oracle Sales Cloud to Oracle Service Cloud, and Oracle Service Cloud to Oracle Sales Cloud bulk loads, you must first perform the Oracle Sales Cloud to Oracle Service Cloud bulk load operation, then when performing the Oracle Service Cloud to Oracle Sales Cloud bulk load, you must exclude all records that have the `ExternalReference` parameter set to null. Performing bulk import of account records from Oracle Sales Cloud to Oracle Service Cloud involves the following general steps:

1. Import the report definitions provided by Oracle Service Cloud (Country and Province (State) ID Report.xml and Country ID Report.xml) into Oracle Service Cloud and export the report generated as CSV files (`StateIdMapping.csv` and `CountryIdMapping.csv`).
2. Export the data from Oracle Sales Cloud.
3. Combine the exported data from Oracle Sales Cloud using the `RegistryID`.
4. Map the respective `StateId` and `CountryId` columns to the `StateorProvince` and `Country` columns based on the exported reports from Oracle Service Cloud.
5. Create an import mapping in Oracle Service Cloud to import exported data from Oracle Sales Cloud.
6. Import the final `Organization.csv` file into Oracle Service Cloud using the import mapping you created.
7. Create a report in Oracle Service Cloud which contains the imported Org ID and respective External Reference.
8. Export the report created in the above step from Oracle Service Cloud as a CSV file.
9. Create an import mapping in Oracle Sales Cloud (based on the `SourceSystemReference` object) to import the exported CSV file from Oracle Service Cloud.
10. Import the report (containing Org ID and External Reference) from Oracle Service Cloud into Oracle Sales Cloud.
Overview of Bulk Import of Contact Records from Oracle Sales Cloud to Oracle Service Cloud

This topic presents a high-level overview of bulk import of contact records. Performing bulk import of account records from Oracle Sales Cloud to Oracle Service Cloud involves the following general steps:

**Note:** You must perform bulk import of account records prior to importing contact records.

1. Import the report definitions provided by Oracle Service Cloud (Country and Province (State) ID Report.xml and Country ID Report.xml) into Oracle Service Cloud, and export the report generated as .CSV files (StateIdMapping.csv and CountryIdMapping.csv).
2. Export data from Oracle Sales Cloud.
3. Combine the exported data from Oracle Sales Cloud using the RegistryID.
4. Map the respective StateId and CountryId columns to the StateorProvince and Country columns based on the exported reports from Oracle Service Cloud.
5. Create an import mapping in Oracle Service Cloud to import exported data from Oracle Sales Cloud.
6. Import the final Contact.csv file into Oracle Service Cloud using the created import mapping.
7. Create a report in Oracle Service Cloud which contains the imported Contact ID and respective External Reference.
8. Export the newly created report from Oracle Service Cloud as a CSV file.
9. Create an import mapping in Oracle Sales Cloud (based on the SourceSystemReference object) to import the exported CSV file from Oracle Service Cloud.
10. Import the report (containing Contact ID and External Reference) from Oracle Service Cloud into Oracle Sales Cloud.

About the Account and Contact Bulk Import Process

The process of bulk importing accounts and contacts data involves the following four main steps:

1. Exporting Country and State or Province Mapping Files from Oracle Service Cloud.
2. Performing Bulk Import of Account Records from Oracle Sales Cloud to Oracle Service Cloud.
3. Performing Bulk Import of Contact Records from Oracle Sales Cloud to Oracle Service Cloud.
4. Performing Original System Record Import.

Exporting Country and State or Province Mapping Files from Oracle Service Cloud

To export county and state or province mapping files from Oracle Service Cloud to Oracle Sales Cloud, perform the following tasks in order.
Checking Privileges
You must first ensure that you have the correct privileges in the Oracle Service Cloud client to create and export reports.

1. In Oracle Service Cloud, navigate to Configuration, Staff Management, and then Profiles.
2. Double-click the profile records to open for edit, and then select Permissions on the top toolbar.
3. Make sure the profile has the Business Process Setting check box enabled.
4. Select Analytics, and make sure the following check boxes are enabled: Create/Edit Reports, Customize Reports, and Create/Edit Public Reports.

Creating and Exporting Reports
You create and export reports using the report definition provided by Oracle Service Cloud, and located in the OSC_SVC.zip file.

1. Sign in to Oracle Service Cloud.
3. Click the Import Existing Report Definition link.
4. Select the report definition: County_StateProvince_IDMap.xml, then save the report.
5. Click the Home menu, and then select Report View
6. Click Export, and select delimited, comma, and from the Export Options dialog box, set the location, then click OK.
7. Repeat the previous steps for the following report: CountryIDMapping.xml.

Performing Bulk Import of Account Records from Oracle Sales Cloud to Oracle Service Cloud
Use the following tasks to perform bulk import of account records from Oracle Sales Cloud to Oracle Service Cloud.

Performing Bulk Export of Account Records from Oracle Sales Cloud
Use this task to perform bulk export of account records from Oracle Sales Cloud to Oracle Service Cloud.

1. In Oracle Sales Cloud, navigate to Setup and Maintenance.
2. In the Search field, enter Manage File Export Activities.
3. In the Search results, select the task, and then click Go to Task.
4. Click the Create button to create a new file export.
5. In the Edit Export Process Definition page, enter the following required information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the file export template.</td>
</tr>
<tr>
<td>Parent Object</td>
<td>Account.</td>
</tr>
<tr>
<td>File Name</td>
<td>The name of the file to be exported.</td>
</tr>
</tbody>
</table>
6. Click Next, then under Edit Export Process Definition, locate Account Profile in the list and disable (deselect) each box except Account Address.

7. Select Edit Filter Criteria, then in the dialog box, click AFTER from the LastUpdateDate list of values, and enter a date which includes all records that are not currently synchronized to Oracle Service Cloud.

8. Select Account Profile in the Export Objects list, then under Details deselect each box except the following: Organization Name, and PartyId.


10. Select Edit Filter Criteria, then in the dialog box, set the OverallPrimaryFlag value to Y.

11. Select Account Address in the Export Objects list, then under Details check each of the following (RegistryId and SiteNumber are selected by default):
   - Country
   - Address line 1
   - Address line 2
   - Address line 3
   - Address line 4
   - City
   - PostalCode
   - State
   - Province

12. Click Next, and then in the Create Schedule view, choose the schedule type of Immediate.

13. Click Next, and then from the Review view, click Activate.

14. In the Overview page, click the Refresh button to view your status.

15. After the Status field has changed to Succeeded, locate the export file in the Exported data file column, and then click the zip file to view.

Performing Post Processing of Account Bulk Import

For post processing of Account, you can use the Account.sh script, located in the Scripts/Bulk_Loading/Account/OSC_SVC/ folder, as a reference implementation.

Run the script as follows:

```
sh Account.sh Address.csv OrganizationProfile.csv CountryStateMapping.csv
```

1. Locate and open the Address.csv file.

2. Insert a new column before Address line 1 and name it as Street.

3. Merge the following columns: Address line 1, Address Line 2, Address Line 3, Address Line 4, and add the value to Street, similar to the following example:

```
<table>
<thead>
<tr>
<th>Address line 1</th>
<th>Address line 2</th>
<th>Address line 3</th>
<th>Address line 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>#24823</td>
<td>Alcoa Drive</td>
<td>Redmond</td>
<td>CA</td>
</tr>
</tbody>
</table>
```

```
#24823, Alcoa Drive, Redmond, CA
```

4. Delete the following columns: Address line 1, Address Line 2, address Line 3 and Address Line 4.
5. Insert a new column before State and name it StateorProvince.
6. Enter the value of Province in StateorProvince if Province is not empty else enter the value of State in StateorProvince similar to the following example:

<table>
<thead>
<tr>
<th>State or Province</th>
<th>State or Province</th>
</tr>
</thead>
<tbody>
<tr>
<td>KA</td>
<td>Redmond</td>
</tr>
<tr>
<td>BC</td>
<td></td>
</tr>
<tr>
<td>KA</td>
<td>BC</td>
</tr>
<tr>
<td>BC</td>
<td></td>
</tr>
</tbody>
</table>

7. Note in the following table the maximum lengths of fields. If any of the fields shown below, in samples from the Address.csv file and the OrganizationProfile.csv file, exceed the length shown, they must be truncated.

<table>
<thead>
<tr>
<th>Address.csv Fields</th>
<th>Maximum Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Street</td>
<td>80</td>
</tr>
<tr>
<td>StateorProvince</td>
<td>255</td>
</tr>
<tr>
<td>City</td>
<td>80</td>
</tr>
<tr>
<td>Postal Code</td>
<td>10</td>
</tr>
<tr>
<td>Country</td>
<td>255</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OrganizationProfile.csv Fields</th>
<th>Maximum Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrganizationName</td>
<td>80</td>
</tr>
</tbody>
</table>

8. Now, merge the Organization.csv file with the Address.csv file based on the common Registry ID column.
9. After merging, open the OrganizationProfile.csv file, then do the following:
   a. Create a new column before the Country column called Country ID.
b. Use the CountryIDMapping.csv file exported from Oracle Service Cloud by importing the Report Definition provided by Oracle Service Cloud "CountryID Mapping.xml" to map the values in Country with their corresponding CountryId to Country ID column.

10. Map the StateorProvince column to Oracle Service Cloud State ID, then do the following:
   a. Create a new column before the StateorProvince column called State ID.
   b. Use the StateIDMapping.csv file exported from Oracle Service Cloud by importing the Report Definition provided by Oracle Service Cloud (Country and Province (State) ID Report.xml) to map the values in Country with their corresponding StateId to State ID column.

11. Split the Organization.csv file into multiple files (Organization-1.csv, Organization-2.csv, Organization-3.csv, and so on) of the number of records exceed one million such that each split file has maximum number of records equal to or less than one million records.

Importing Accounts into Oracle Service Cloud

💡 Note: The Org_ID_ExtRef.csv file is the file that is exported during Account import from Oracle Sales Cloud to Oracle Service Cloud.

1. In Oracle Service Cloud, navigate to Configuration, Database, and then Data Import Templates.
2. In the Date Import Templates list, select Organization, and then click the New button.
3. In the Data Import Template - Edit form, enter a name for the template, and provide the names of the map columns in the CSV file to the Oracle Service Cloud fields.
4. In the Duplicate Criteria field, enter the following: `ext_ref={PartyId} AND name={Organization Name}`
5. Save the template.

Importing the Merged CSV File

1. Navigate to Configuration, Database, and then Data Import Wizard.
2. From the Data Record Type drop-down list, choose Organization.
3. For the Data File, navigate to the merged Organization.csv file.
4. Select the newly created template, and make sure each column is mapped to the correct field, then click Next to initiate the scan.
   All records are scanned.
5. Ensure all records are imported.

Creating an Organization ID External Reference Report

1. In Oracle Service Cloud, click File, then select Report.
2. In the New Report screen, click the Import Existing Report Definition link.
4. Save the report locally, then click the Home menu, then Views, and the Report View.
5. Click the Export button, then choose Delimited, and then Comma to export the report as a CSV file.

Importing Account External References into Oracle Sales Cloud

1. Log in to Oracle Sales Cloud, and click Setup and Maintenance.
2. In the Search view, enter Manage File Import Mappings, then in the Search Results, click Go to Task.
3. Create a new import map with the following values:
### Importing the Original System Reference into Oracle Sales Cloud

Use this task to import the Original System Reference into Oracle Sales Cloud.

1. In Oracle Sales Cloud, click Setup and Maintenance.
2. In the Search view, enter Manage File Import Activities, then in the Search Results, click Go to Task.
3. In the Manage Import Activities view, click the Create button.
4. In the Create Import Activity: Enter Import Options view, enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>ORG_OSR</td>
</tr>
<tr>
<td>Object</td>
<td>Source System Reference</td>
</tr>
<tr>
<td>File Type</td>
<td>Text File</td>
</tr>
<tr>
<td>Upload From</td>
<td>Desktop</td>
</tr>
<tr>
<td>File Name</td>
<td>The name of the CSV file you exported from Oracle Service Cloud.</td>
</tr>
<tr>
<td>Data Type</td>
<td>Comma Separated</td>
</tr>
<tr>
<td>Import Mapping</td>
<td>ORG_OSR_Import_Map (this is the mapping you created previously).</td>
</tr>
</tbody>
</table>

5. Click Next.
6. In the Map Fields screen, select the previously created import mapping shown below:

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Object</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>External ID</td>
<td>SourceSystemReference</td>
<td>OrigSystemReference</td>
</tr>
<tr>
<td>External Reference</td>
<td>SourceSystemReference</td>
<td>ExistingOwnerTableID</td>
</tr>
</tbody>
</table>

7. Under the Set Constant Values area, specify the following information:

<table>
<thead>
<tr>
<th>Object</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceSystemReference</td>
<td>OwnerTableName</td>
<td>HZ_PARTIES</td>
</tr>
<tr>
<td>SourceSystemReference</td>
<td>OrigSystem</td>
<td>RNOW</td>
</tr>
</tbody>
</table>

8. Click Next.
9. In the Create Schedule view, select Immediate and click Next.
10. In the Review and Activate view, review, then click Activate.
11. Wait until the status of the import activity changes from Scheduled to Completed, and once the status has changed, all records were successfully imported.
12. If the status changes to “completed with errors” this indicates that not all records were successfully imported. View errors by clicking the completed with errors link.

Performing Bulk Import of Contacts Records from Oracle Sales Cloud

Perform the following tasks in the order shown.
First you perform bulk import of contact records from Oracle Sales Cloud.

> **Note:** You must perform bulk import of account records prior to importing contact records.

1. In Oracle Sales Cloud, click Setup and Maintenance, then under Search: Tasks, search for Manage File Export Activities.
2. In the Search Results field, click Go to Task.
3. In the Overview page, click the Create button to create a new file export.
4. In the Edit Export Process Definition page, enter the following required information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the file export template.</td>
</tr>
<tr>
<td>Parent Object</td>
<td>Contact</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>File Name</td>
<td>The name of the file to be exported.</td>
</tr>
</tbody>
</table>

5. Click Next.
6. In the Edit Export Process Definition view, locate Contact Profile in the list and disable (deselect) each box except the following:
   - Contact Address
   - Contact Email
   - Contact Fax
   - Contact Mobile
   - Contact Phone
7. Enable the required Attribute Name, and set the view criteria for each object using the following table:

<table>
<thead>
<tr>
<th>Export Objects</th>
<th>Attribute Name</th>
<th>View Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Profile</td>
<td>PartyId, PersonFirstName, PersonLastName, PartyNumber, PrimaryCustomerld</td>
<td>LastUpdateDate AFTER date which includes all records that are not currently synchronized to Oracle Service Cloud.</td>
</tr>
<tr>
<td>Contact Address</td>
<td>PartyId, Country, Address1, Address2, Address3, Address4, City, Postal Code, State, Province, PartySiteNumber, PartyNumber</td>
<td>Set the OverallPrimaryFlag to Y</td>
</tr>
<tr>
<td>Contact Email</td>
<td>PartyId, EmailAddress, PartyNumber</td>
<td>Set the OverallPrimaryFlag to Y</td>
</tr>
<tr>
<td>Contact Fax</td>
<td>PhoneNumber, PhoneExtension, PhoneAreaCode, PhoneCountryCode, PartyNumber, PartyId, CreationDate</td>
<td>LastUpdateDate AFTER date which includes all records that are not currently synchronized to Oracle Service Cloud.</td>
</tr>
<tr>
<td>Contact Mobile</td>
<td>PhoneNumber, PhoneExtension, PhoneAreaCode, PhoneCountryCode, PartyNumber, PartyId, CreationDate</td>
<td>LastUpdateDate AFTER date which includes all records that are not currently synchronized to Oracle Service Cloud.</td>
</tr>
<tr>
<td>Contact Phone</td>
<td>ContactPointPurpose, OverallPrimaryFlag, PhoneNumber, PhoneExtension, PhoneAreaCode, PhoneCountryCode, PartyNumber, PartyId, CreationDate</td>
<td>LastUpdateDate AFTER date which includes all records that are not currently synchronized to Oracle Service Cloud.</td>
</tr>
</tbody>
</table>
8. Activate the export process.
9. In the History view, click the link under Exported data file.
10. Decompress and locally save the archive.

Performing Post Processing of Contact Bulk Import
For post processing tasks, you can use an included script as reference. The script name is: Contact.sh and you run it as follows:
To manually process, use the following tasks:

Adding and Populating the Street Column

1. Locate and open the Address.csv file.
2. Insert a new column before AddressLine1 and name it Street.
3. Merge the following columns: AddressLine1, AddressLine2, AddressLine3, AddressLine4, and add the value to Street, similar to the following example:

<table>
<thead>
<tr>
<th>AddressLine1</th>
<th>AddressLine2</th>
<th>AddressLine3</th>
<th>AddressLine4</th>
</tr>
</thead>
<tbody>
<tr>
<td>123 Main St.</td>
<td>Suite 303</td>
<td>3rd Floor</td>
<td>Building #2</td>
</tr>
</tbody>
</table>

After the merge the Street column looks like this:

<table>
<thead>
<tr>
<th>Street</th>
</tr>
</thead>
<tbody>
<tr>
<td>123 Main St., Suite 303, 3rd Floor, Building #2</td>
</tr>
</tbody>
</table>

4. Delete the following columns: AddressLine1, AddressLine2, AddressLine3, AddressLine4

Adding and Populating a StateorProvince Column

1. In the Address.csv file, insert a new column before State and name it StateorProvince.
2. Populate the StateorProvince column by merging State and Province fields similar to the following example:

<table>
<thead>
<tr>
<th>State</th>
<th>Province</th>
</tr>
</thead>
<tbody>
<tr>
<td>KA</td>
<td>Redmond</td>
</tr>
<tr>
<td>BC</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Province</th>
</tr>
</thead>
<tbody>
<tr>
<td>KA</td>
<td>BC</td>
</tr>
</tbody>
</table>

After the merge, the StateorProvince field looks like this:

<table>
<thead>
<tr>
<th>S. No</th>
<th>StateorProvince</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>KA</td>
</tr>
<tr>
<td>2.</td>
<td>BC</td>
</tr>
</tbody>
</table>
Adding and Populating a Fax Column

1. Open the Fax.csv file.
2. Insert a new column before the Registry ID column and call it Fax.
3. Populate the Fax field value by merging the following fields: Phone Country Code, Area Code, Phone, and Extension similar to the following example:

<table>
<thead>
<tr>
<th>Phone</th>
<th>Extension</th>
<th>Area Code</th>
<th>Phone Country Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>4345</td>
<td>12</td>
<td>1</td>
</tr>
</tbody>
</table>

After the merge, the Fax field looks like this:

Fax

+1 (12) 23 x4345

4. Delete the Phone Country Code, Area Code, Phone and Extension columns.

Adding and Populating a Mobile Column

1. Open the Mobile.csv file.
2. Insert a new column before the Registry ID column and call it Mobile.
3. Populate the Mobile field value by merging the following fields: Phone Country Code, Area Code, Phone, and Extension similar to the following example:

<table>
<thead>
<tr>
<th>Phone</th>
<th>Extension</th>
<th>Area Code</th>
<th>Phone Country Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>23-3456</td>
<td>2345</td>
<td>123</td>
<td>1</td>
</tr>
</tbody>
</table>

After the merge, the Fax field looks like this:

Mobile

+1 (123) 234-3456 x2345
4. Delete the Phone Country Code, Area Code, Phone and Extension columns.

Mapping the OrganizationId Column to the PersonProfile.csv File

1. Open the PersonProfile.csv file.
2. Add an OrganizationId column.
3. Populate OrganizationId based on the field OrganizationId from the Org_ID_ExtRef.csv file using the following example:

<table>
<thead>
<tr>
<th>Organization ID</th>
<th>ExternalReference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1687</td>
<td>300100051268389</td>
</tr>
</tbody>
</table>

PersonProfile.csv:

<table>
<thead>
<tr>
<th>Registry ID</th>
<th>First Name</th>
<th>Last Name</th>
<th>PartyId</th>
<th>PrimaryCustomerId</th>
<th>OrganizationId</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDRM_81437</td>
<td>Kristen</td>
<td>Patrick</td>
<td>300100051217620</td>
<td>300100051268389</td>
<td>1687</td>
</tr>
</tbody>
</table>

After the merge, the PersonProfile.csv file looks like this:

<table>
<thead>
<tr>
<th>Registry ID</th>
<th>First Name</th>
<th>Last Name</th>
<th>PartyId</th>
<th>PrimaryCustomerId</th>
<th>OrganizationId</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDRM_81437</td>
<td>Kristen</td>
<td>Patrick</td>
<td>300100051217620</td>
<td>300100051268389</td>
<td>1687</td>
</tr>
</tbody>
</table>

Merging the PersonProfile, Address, Email, Fax, Phone, and Mobile Export Files

1. Open the PersonProfile.csv file.
2. Add the following columns:
   - Country
   - Street
   - City
   - Postal Code
   - State/Province
   - Work Phone
   - Home
   - Mobile
   - Fax
   - Email
3. Merge the PersonProfile with Address, Email, Fax, Phone and Mobile using the common Registry ID field.
If a given contact record has more than one Fax number, Mobile number, Work Phone number or Home Phone number, then merge only the record with the oldest creation date. The two newly created fields (Home and Work Phone) of PersonProfile.csv file are populated based on the Phone.csv file. For a record (in the Phone.csv file), enter the value of from the Phone column (in the Phone.csv file), or in the Home column if the ContactPointPurpose value is set to PERSONAL and the OverallPrimaryFlag is set to N. Otherwise, enter the value of Phone in Work Phone column.

4. Use the Phone.csv file to add the values from the Phone column to the Home column in the PersonProfile.csv file.

5. Add the Phone column values, from the Phone.csv file to the Home column in the PersonProfile.csv file if the following are true in the Phone.csv file. If Purpose=PERSONAL and Primary=N. If not, map the column values to the Work Phone column in the PersonProfile.csv file.

6. Use the following example from the PersonProfile.csv file:

Person Profile.csv

<table>
<thead>
<tr>
<th>Registry ID</th>
<th>First Name</th>
<th>Last Name</th>
<th>PartyId</th>
<th>PrimaryCustomerId</th>
<th>OrganizationId</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDRM_81437</td>
<td>Kristen</td>
<td>Patrick</td>
<td>300100051217620</td>
<td>300100051268389</td>
<td>1687</td>
</tr>
</tbody>
</table>

Address.csv

<table>
<thead>
<tr>
<th>Registry ID</th>
<th>Country</th>
<th>Street</th>
<th>City</th>
<th>Postal Code</th>
<th>State/Province</th>
<th>PartyId</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDRM_81437</td>
<td>US</td>
<td>2106 Sinton Ave Apt 105</td>
<td>Cincinnati</td>
<td>45206</td>
<td>OH</td>
<td>300100051217620</td>
</tr>
</tbody>
</table>

Email.csv

<table>
<thead>
<tr>
<th>Registry ID</th>
<th>PartyId</th>
<th>E-Mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDRM_81437</td>
<td>300100051217620</td>
<td><a href="mailto:kristen.patrick@futaba.com">kristen.patrick@futaba.com</a></td>
</tr>
</tbody>
</table>

Phone.csv

<table>
<thead>
<tr>
<th>Registry ID</th>
<th>PartyId</th>
<th>Purpose</th>
<th>Primary</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDRM_81437</td>
<td>300100051217620</td>
<td>Business</td>
<td>2106 Sinton Ave Apt 105</td>
<td>Cincinnati</td>
</tr>
</tbody>
</table>

After the merge, the PersonProfile.csv file looks like this:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registry ID</td>
<td>CDRM_81437</td>
</tr>
<tr>
<td>First Name</td>
<td>Kristin</td>
</tr>
</tbody>
</table>
Adding Columns to the PersonProfile.csv File

Use these topics to add additional columns to the PersonProfile.csv file.

Adding a Country ID column to the PersonProfile.csv File

1. Open the PersonProfile.csv file.
2. Insert a new column after the Country column and call it Country ID.
3. Use the CountryIDMapping.csv file exported from Oracle Service Cloud by importing the Report Definition provided by Oracle Service Cloud CountryID Mapping.xml to map the values in the Country column with their corresponding CountryId to Country ID column. See the following example:
Adding a State/Province Column to the PersonProfile.csv File

1. Open the PersonProfile.csv file.
2. Insert a new column after the Country column and call it State/Province ID.
3. Use the StateIDMapping.csv file exported from Oracle Service Cloud by importing the Report Definition provided by Oracle Service Cloud Country and Province (State) ID.xml to map the values in the Country column with the corresponding StateId to State ID column. See the following example:

<table>
<thead>
<tr>
<th>State/Province</th>
<th>State/Province ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>OH</td>
<td>41</td>
</tr>
<tr>
<td>NY</td>
<td>40</td>
</tr>
</tbody>
</table>

Truncating Fields

1. Open the PersonProfile.csv file.
2. If any of the fields in the file exceed the maximum limit as shown below, truncate the field value to its maximum length.

<table>
<thead>
<tr>
<th>Field</th>
<th>Maximum Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>80</td>
</tr>
<tr>
<td>Last Name</td>
<td>80</td>
</tr>
<tr>
<td>Street</td>
<td>240</td>
</tr>
<tr>
<td>State/Province</td>
<td>255</td>
</tr>
<tr>
<td>Postal Code</td>
<td>10</td>
</tr>
<tr>
<td>EmailAddress</td>
<td>80</td>
</tr>
<tr>
<td>Field</td>
<td>Maximum Length</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Work Phone</td>
<td>40</td>
</tr>
<tr>
<td>Home Phone</td>
<td>40</td>
</tr>
<tr>
<td>Mobile</td>
<td>40</td>
</tr>
<tr>
<td>Fax</td>
<td>40</td>
</tr>
</tbody>
</table>

3. Save the file as Contact.csv.
4. Split the Contact.csv file into multiple files: Contact_1.csv, Contact_2.csv and so on.

Creating an Import Template for Oracle Service Cloud

The following is a one-time requirement.

1. Sign in to Oracle Service Cloud.
2. Navigate to Configuration, Database, and then Data Import Templates.
3. From the Data Import Templates menu, choose Contact, and then click the New tab to create a new template.
4. Provide a name for the template and under Column Mappings map the columns from the Contact.csv file to the Oracle Service Cloud field.
5. The field names include the following:
   - External Reference(ext_ref)
   - First Name(first_name)
   - Last Name(last_name)
   - Country(country_id)
   - Street(street)
   - City(city)
   - Postal Code(postal_code)
   - State/province(prov_id)
   - Office Phone(ph_office)
   - Home Phone(ph_home)
   - Mobile Phone(ph_mobile)
   - Fax Phone(ph_fax)
   - Email Address(email)
   - Organization(org_id)

6. In the Duplicate Criteria field, enter the following:
ext_ref={PartyId} AND any_email={Email}

7. Save the template.

Process of Importing Merged Files into Oracle Service Cloud

Importing Merged Files

1. Sign in to Oracle Service Cloud.
2. Navigate to Configuration, Database, and Data Import Wizard.
3. Click the Data Record Type drop-down list, and choose Contact, and under Data File, select the final merged Contact.csv file.
4. Select the template you created, and ensure all columns are properly mapped, then click Next.
5. Click the Ignore Errors check box, then wait until all records have been imported.

Creating a Report

Now you create a report which exports IDs and ExternalReference pairs for newly imported records. The integration package includes a reference script which you can use to add the CON_ prefix as follows assuming you have saved the export under Contact_RNOW_ID_ExtRef.csv.

The script is as follows:

```
sh prepareContactSourceReference.sh Contact_RNOW_ID_ExtRef.csv
```

1. Sign in to Oracle Service Cloud, click File, and then select Report.
2. Click the Import Existing Report Definition link.
3. Select the Contact_ID_ExtReference.xml report.
4. Save the report locally, and then click the Home menu, then Views, and then Report View.
5. Click the Export button, then choose Delimited, and then Comma to export the report as a CSV file. Under Report Options make sure Add report name to output is not checked.
6. Run the `sh prepareContactSourceReference.sh Contact_RNOW_ID_ExtRef.csv` script.

Importing the Original System Reference (OSR)

Now you import the original system reference using the following tasks:

1. Sign in to Oracle Sales Cloud, and click Setup and Maintenance.
2. Under Tasks, search for Manage File Import Activities, then in the Search Results, click Go to Task.
3. Create a new import activity using the following information, then click Next.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>CON_OSR</td>
</tr>
<tr>
<td>Object</td>
<td>Source System Reference</td>
</tr>
</tbody>
</table>
Field | Value
---|---
File Type | Text File
Upload From | Desktop
File Name | Contact_RNOW_ID_ExtRef.csv
Data Type | Comma Separated
Import Mapping | ORG_OSR_Import_Map

4. In the Map Fields view, under Select Import mapping, select the import mapping you previously created during the account import step and all fields are automatically filled.

5. Under the Set Constant Values area, specify the following information:

<table>
<thead>
<tr>
<th>Object</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceSystemReference</td>
<td>OwnerTableName</td>
<td>HZ_PARTIES</td>
</tr>
<tr>
<td>SourceSystemReference</td>
<td>OrigSystem</td>
<td>RNOW</td>
</tr>
</tbody>
</table>

6. Click Next.

7. In the Schedule view, select Immediate, and then click Next.

8. In the Review and Activate view, click Activate.

9. Follow the status value until the job status changes from Scheduled to Completed.

Performing Bulk Export then Import of Organizations from Oracle Service Cloud to Oracle Sales Cloud

The following topics provide a step-by-step description of how to extract Organization and Contact data from Oracle Service Cloud and upload it to Oracle Sales Cloud. This requirement is explained with two sample scenarios.

When you are performing the initial set-up of an ongoing data synchronization between an existing Oracle Sales Cloud implementation and existing Oracle Service Cloud applications, you must perform an initial synchronization of existing Organization and Contacts between the two applications before enabling an on-going synchronization.

Note: If you are importing any Contact data from Oracle Service Cloud to Oracle Sales cloud, you must first import all Organization data (from Oracle Service Cloud to Oracle Sales Cloud). This occurs because associations between Organizations and Contacts might exist. Importing Organizations prior to Contacts maintain the existing associations between the two.
There are several steps involved in exporting Organizations data from Oracle Service Cloud and then importing the data into Oracle Sales Cloud, where Organizations become Sales Accounts. Here are the high-level steps:

1. Create a custom report in Oracle Service Cloud to export Organization data. You can use one of two tasks shown below to create this custom report. Either create the report from scratch, or use an included script.
2. Run the custom report, export the data, and save it locally.
3. Transform the organization data exported in the previous step.
4. Upload or import the transformed data into Oracle Sales Cloud to create Sales Accounts.

Creating and Running a New Custom Report in Oracle Service Cloud

The first step to exporting data out of Oracle Service Cloud is to create a report to display requisite data after applying the appropriate filters. In the specific case of exporting Organization data from Oracle Service Cloud, you only export a selected number of attributes such as Organization ID, Organization Name, External Reference (also known as the Fusion Party Reference ID) and Address details of an Organization. This requires you to create a custom report to include just those required attributes.

*Note:* The integration package includes a reference report definition which can be imported into Oracle Service Cloud (Unsynched_Orgs_By_Create_Data.xml) for this purpose. To use the script, see the task which follows. Alternatively you can use the following task to create the custom report.

Using a Script to Create a Custom Report

1. In Oracle Service Cloud, click File, and then Report.
2. Click the Import Existing Report Definition link.
3. Select the Unsynched_Orgs_By_Create_Data.xml file from the dialog box.
4. Make any required changes, then save the file.

Creating a New Custom Report

1. In Oracle Service Cloud, click File, and then Report.
2. Click the Standard Report link.
3. In the New Report Designer, toward the bottom of the page, under Data Dictionary, select Organization from the list of objects.
4. From the list of fields, select Organization ID, and Organization Name, and drag and drop them up to the Data area.
5. Now, back in the Data Dictionary, click the Organization_Addresses object and drag and drop it in the Data area.

A join is automatically created between the Organization object and the Organization_Addresses object.

6. Optionally, filter your data by doing the following:
   a. Click the Sort button on the ribbon.
   b. In the Sort dialog box, select the primary sort attribute, and choose whether you want the sort order to be Ascending or Descending. You can set four sort variables.
   c. Click OK when your sort variables are set.
   d. Apply filter criteria by clicking the Level Filter icon, under the Level tab, on the ribbon.
e. In the Add Filter dialog box, use filters such as Date Created or Address Type to limit the amount of data returned. Note, only include Organizations the ExternalReference (orgs.ext_ref) parameter is null, if you want only unsynchronized organizations to be exported.

f. You can also use expressions to enhance your filtering.

g. Click OK when finished.

7. Now save the report by clicking the Save icon above the File menu, and specify a name for the report.

Running the Newly Created Custom Report

You generate your required data by running the newly created custom report.

1. Locate the custom report by name in the Navigation panel. If you do not see the custom report in the Navigation panel, click the Settings (gear wheel) icon to add your custom report to the list of displayed reports. After opening the report, you have to option to change your filters.

2. After modifying your report filters, if necessary, click Search. A complete list of organizations matching your search criteria is displayed.

3. Click the Export button on the ribbon, and select Delimited, then Comma to export a CSV file to your local computer.

4. In the Export Options dialog box, specify where you want to save the file, then click OK.

Performing Post-Processing of the Data File

The CSV data file that you have just created is not yet ready to be imported into Oracle Sales Cloud. You must perform some postprocessing tasks to remove extraneous data and also to introduce some additional information. Note that the transformation of the exported CSV file can be done through any appropriate tool which enables you to manipulate a CSV file. One way to perform the transformation is through scripts.

For post processing of organization data exported from Oracle Service Cloud, you can use, as a reference implementation, the Account.sh script, located in the following folder:

Scripts/Bulk_Loading/Account/SVC_OSC/ folder

Run the script as follows:

For Linux: Convert the mode of the file to executable through the following command:

```bash
chmod 755 Account.sh
```

Then, execute the script by typing the following:

```bash
./Account.sh
```

For Window:

Execute the script by typing the following:

```bash
Account.sh
```

The script begins executing and prompts you to enter certain parameters which are listed below. After supplying the parameters, the data is transformed and committed to the AccountUpdated.csv file.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename</td>
<td>File to be transformed. The file must be in the same folder.</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
Delimiter | Character used as a delimiter in the file to be transformed. Since comma is usually part of most data, Oracle recommends that a character other than comma or space be used as a delimiter (for example: ~).

ColumnIndex for State/Province | The index of StateorProvince column number in your file.

Note: All the required files (the domain value map (DVM) and the file to be transformed) must be in the same folder as the script that is executed. The name of the DVM file must be CountryWithProvince.csv.

1. Open the CountryWithProvince.csv file.
2. Ensure that the State and Country codes match the definitions in Oracle Sales Cloud.
   For example, if the Country column lists United States, you must change the values to US to match Oracle Sales Cloud. See the following tables for a sample representation of pretransformation and posttransformation values.
   - Before Transformation: The Country column shows United States (US)
   - After Transformation: The Country column shows US.

   The Oracle Sales Cloud data model requires State and Province to be in different columns. For countries that support State, that data must be in State column, for others, the data must be in Province column. See the following tables for a sample representation of pretransformation and posttransformation values.

   Before Transformation:

<table>
<thead>
<tr>
<th>StateorProvince</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>US</td>
</tr>
<tr>
<td>NE</td>
<td>US</td>
</tr>
<tr>
<td>SK</td>
<td>CA</td>
</tr>
<tr>
<td>QC</td>
<td>CA</td>
</tr>
</tbody>
</table>

   After Transformation:

<table>
<thead>
<tr>
<th>State</th>
<th>Province</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td></td>
<td>US</td>
</tr>
<tr>
<td>NE</td>
<td></td>
<td>US</td>
</tr>
<tr>
<td></td>
<td>SK</td>
<td>CA</td>
</tr>
</tbody>
</table>
3. After verifying the file, save it.

Importing Post-Processed Organization Data into Oracle Sales Cloud

Importing data into Oracle Sales Cloud is a multistep process. In this step you set up the mapping that will be used when the transformed data is imported. Generally this is done one time.

1. In Oracle Sales Cloud, click Setup and Maintenance.
2. In the Search: Tasks area, search for Manage File Import Mappings.
3. In the Search Results field, click the Go to Task icon.
4. On the Manage File Import Mappings page, click the New button to create a new mapping.
5. In the Create Import Mapping dialog box, provide the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Mapping</td>
<td>Provide a name.</td>
</tr>
<tr>
<td>Object</td>
<td>Account</td>
</tr>
<tr>
<td>File Type</td>
<td>Text file.</td>
</tr>
</tbody>
</table>

6. Click Save and Close.
7. In the Manage File Import Mappings page, click the name of the newly created mapping link.
8. On the Edit Import Mapping page, click the Add (+) icon and provide the following information:

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Object</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization ID</td>
<td>OrganizationProfile</td>
<td>PartyOrigSystemRefNo</td>
</tr>
<tr>
<td>Organization Name</td>
<td>OrganizationProfile</td>
<td>OrganizationName</td>
</tr>
<tr>
<td>Street</td>
<td>SellToAddress</td>
<td>Address1</td>
</tr>
<tr>
<td>City</td>
<td>SellToAddress</td>
<td>City</td>
</tr>
<tr>
<td>State/Province</td>
<td>SellToAddress</td>
<td>State</td>
</tr>
<tr>
<td>Column Header</td>
<td>Object</td>
<td>Attribute</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Postal Code</td>
<td>SellToAddress</td>
<td>PostalCode</td>
</tr>
<tr>
<td>Country</td>
<td>SellToAddress</td>
<td>Country</td>
</tr>
</tbody>
</table>

9. Click Save and Close.

The mapping is complete. You can reuse this task to insert new or update existing rows into the Account object in Oracle Sales Cloud. This mapping can be used repeatedly during the actual import process.

10. In the Search: Tasks area, enter Manage File Import Activities and click Search.

11. In the Search Results, select the task, then click the Go to Task icon.

12. In the Manage Import Activities page, click the New icon.

13. In the Manage File Import Objects wizard, specify the following information:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name for the import activity.</td>
</tr>
<tr>
<td>Object</td>
<td>Account</td>
</tr>
<tr>
<td>Upload From</td>
<td>Select the Desktop button.</td>
</tr>
<tr>
<td>Import Mapping</td>
<td>Click the drop-down list, and select the newly created mappings.</td>
</tr>
</tbody>
</table>

14. Click Next.

The Edit Import Activity: Map Fields page appears. This page displays the mapping of the attributes to the transformed data.

15. Under the Set Constant Values area, specify the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>OrganizationProfile</td>
</tr>
<tr>
<td>Attribute</td>
<td>PartyOrigSystem</td>
</tr>
<tr>
<td>Value</td>
<td>RNOW</td>
</tr>
</tbody>
</table>

16. Click Next, then in the Create Schedule page, choose Immediate from the Schedule drop-down list, then click Next.

17. In the Review and Activate page, confirm the accuracy of all the information, then click Activate.

18. In the Manage File Import Objects page, monitor the progress from Scheduled to Completed.

19. If the status indicates Completed with Errors, then one or more errors occurred during the import process.

20. Refer to the appropriate log to see which data failed. View the log file by opening the attachment associated with the File Import object item.

21. After the process completes, navigate to Sales, then choose Accounts and verify that Accounts were successfully created.
Importing Account PartyID Values into Oracle Service Cloud

After you have verified that the Sales Accounts have been successfully created in Oracle Sales Cloud, the next step is to update the original Organization records (in Oracle Service Cloud) with the PartyID that was assigned by Oracle Sales Cloud. This finishes the process, and both applications (Oracle Sales Cloud and Oracle Service Cloud) have a mapping of unique IDs assigned by both applications.

1. After your import task completes successfully, the unique IDs that were assigned by Oracle Sales Cloud to each Account are committed to a log file. To view the log file, do the following:
   a. Navigate to the View Import Status.
   b. Locate the attachment column, and click the attachment for the wanted import item.

2. In the Attachments column, click the appropriate CSV link, and download the file to your local computer.

   The CSV file contains the unique IDs assigned by Oracle Sales Cloud to each Organization that was successfully created. The IDs appear in the ObjectKey column.

3. Sign in Oracle Service Cloud (though the CX Console) and from the Navigation area, choose Configuration, Database, then Data Import Wizard.

4. In the Data Import Wizard, enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Record Type</td>
<td>Organization</td>
</tr>
<tr>
<td>Header Location</td>
<td>Click the drop-down arrow and select First Line of Data File.</td>
</tr>
</tbody>
</table>

5. Click Next, then confirm that all information in the Column in File column conforms to the data exported from Oracle Sales Cloud, then click Next.

6. In the Column Mappings area, remove all rows except for the following:

   o ObjectKey
   o Organization ID

   Remove rows by selecting a row, then clicking the Remove icon.

7. Click Next, and view the prescan data file, then click Next.

   The actual import is completed and the appropriate Organizations updated with the PartyID (referred to as the Fusion Party Reference ID) assigned by Oracle Sales Cloud. Ensure that the record counts are accurate and as expected.

Overview of Performing Bulk Export then Import of Contacts from Oracle Service Cloud to Oracle Sales Cloud

There are several steps involved in exporting Organizations data from Oracle Service Cloud and then importing the data into Oracle Sales Cloud, where Organizations become Sales Accounts.
Note: If you are importing any Contact data from Oracle Service Cloud to Oracle Sales cloud, you must first import all Organization data (from Oracle Service Cloud to Oracle Sales Cloud). This is due to the fact that associations between Organizations and Contacts might exist. Importing Organizations prior to Contacts maintains the existing associations between the two.

The following shows the high-level steps:

1. Create a custom report in Oracle Service Cloud to export Contacts data.
2. Run the custom report, export the data, and save it locally.
3. Transform the Contact data exported in the previous step.
4. Upload or import the transformed data into Oracle Sales Cloud to create Contacts.

Creating a New Custom Report in Oracle Sales Cloud

The first step in exporting data out of Oracle Service Cloud is to create a report to display requisite data after applying the appropriate filters. In the specific case of exporting Contact data from Oracle Service Cloud, you will only export a selected number of attributes such as the following details of a contact:

- Contact ID
- First Name
- Last Name
- Email Address
- Phone Numbers
- Address

The integration package includes a reference report definition which can be imported into Oracle Service Cloud (Unsynced_Contacts_By_Create_Date.xml) for this purpose. Alternatively the report can be created from scratch using the following instructions.

Using a Script to Create a Custom Report

1. In Oracle Service Cloud, click File, and then Report.
2. Click the Import Existing Report Definition link.
3. Select the Unsynced_Contacts_By_Create_Date.xml file from the dialog box.
4. Make any required changes, then save the file.

Newly Creating a Custom Report

1. In Oracle Service Cloud, click File, and then Report.
2. Click the Standard Report link.
3. In the New Report Designer, toward the bottom of the page, under Data Dictionary, select Contacts from the list of objects.
4. From the list of fields, select the following, and drag and drop the into the Data Area:
   - Contact ID (a mandatory field)
   - Organization ID (a mandatory field)
First Name (a mandatory field)
Last Name (a mandatory field)
Street
City
State
Postal Code
Country
Email Address
Office Phone
Mobile Phone
Fax Phone

5. Optionally filter your data by doing the following:
   a. Click the Sort button on the ribbon.
   b. In the Sort dialog box, select the primary sort attribute, such as Last Name, and choose whether you want the sort order to be Ascending or Descending. There are four sort variables you can set.
   c. Click OK when your sort variables are set.
   d. Apply filter criteria by clicking the Level Filter icon, under the Level tab, on the ribbon.
   e. In the Add Filter dialog box, use filters such as Date Created or Address Type to limit the amount of data returned. Note, it is recommended you filter only contacts for which the ExternalReference parameter is null.
   f. You can also use expressions to enhance your filtering.
   g. Click OK when finished.

6. Now save the report by clicking the Save icon above the File menu, and specify a name for the report.

Running the Newly Created Custom Report
You generate your required data by running the newly created custom report.

1. Locate the custom report by name in the Navigation panel. If you do not see the custom report in the Navigation panel, click the Settings (gear wheel) icon to add your custom report to the list of displayed reports. After opening the report, you have the option to change your filters.
2. After modifying your report filters, if necessary, click Search. A complete list of contacts matching your search criteria is displayed.
3. Click the Export button on the ribbon, and select Delimited, then Comma to export a CSV file to your local computer. If there are more than 1 million records, you must perform multiple exports.
4. In the Export Options dialog box, uncheck the Add report name to the output checkbox, then specify where you want to save the file, and then click OK.

Performing Post-Processing of the Contacts Data File
The CSV data file that you have just created is not yet ready to be imported into Oracle Sales Cloud. You must perform some post-processing tasks to remove extraneous data and also to introduce some additional information. Note that the transformation of the exported CSV file can be done through any appropriate tool which will allow you to manipulate a CSV file. One way to transform the CSV file is through scripts.
In this task you perform two transformations on the Contacts data, shown in the following task:

- Split the StateorProvince column to make it compatible with the Country Code in Oracle Sales Cloud.
- Add the CON_ prefix to the ContactID values

For transforming the contact data exported from Oracle Service Cloud, use as reference implementation, the Contact.sh script, located in the following folder:

scripts/Bulk_Loading/Contact/SVC_OSC/

Run the script as follows:

For Linux:

Convert the mode of the file to executable through the following command:

   chmod 755 Contact.sh

Then execute the script by entering the following:

   ./Contact.sh

For Windows:

Execute the script by entering the following:

   Contact.sh

The script begins executing and prompts you to enter certain parameters which are listed below. After supplying the parameters, the data is transformed and committed to the ContactUpdated.csv file.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename</td>
<td>File to be transformed. The file must be in the same folder.</td>
</tr>
<tr>
<td>Delimiter</td>
<td>Character used as a delimiter in the file to be transformed. Since comma is usually part of most data, it is recommended that a character other than comma or space be used as a delimiter (for example: ~).</td>
</tr>
<tr>
<td>ColumnIndex for Contact</td>
<td>The index of ContactID column in your file. The first column is 1.</td>
</tr>
<tr>
<td>Direction</td>
<td>SvcToOsc.</td>
</tr>
<tr>
<td>ColumnIndex for State/Province</td>
<td>The index of StateorProvince column numbers in your file.</td>
</tr>
</tbody>
</table>

**Note:** All the required files (the domain value map (DVM) and the file to be transformed) must be in the same folder as the script that is executed. The name of the DVM file must be CountryWithProvince.csv.

Performing Post-Processing of the Data File

1. Open the CountryWithProvince.csv file.
2. Ensure that the State and Country codes match the definitions in Oracle Sales Cloud.
For example, if the Country column lists United States, you would need to change the values to US to match Oracle Sales Cloud. See the following tables for a sample representation of pre-transformation and post-transformation values.

- Before Transformation: The Country column shows United States (US)
- After Transformation: The Country column shows US.

The Oracle Sales Cloud data model requires State and Province to be in different columns. For countries that support State, that data must be in State column, for others, the data must be in Province column. See the following tables for a sample representation of pre-transformation and post-transformation values.

Before Transformation:

<table>
<thead>
<tr>
<th>State or Province</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>US</td>
</tr>
<tr>
<td>NE</td>
<td>US</td>
</tr>
<tr>
<td>SK</td>
<td>CA</td>
</tr>
<tr>
<td>QC</td>
<td>CA</td>
</tr>
</tbody>
</table>

After Transformation:

<table>
<thead>
<tr>
<th>State</th>
<th>Province</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td></td>
<td>US</td>
</tr>
<tr>
<td>NE</td>
<td></td>
<td>US</td>
</tr>
<tr>
<td>SK</td>
<td></td>
<td>CA</td>
</tr>
<tr>
<td>QC</td>
<td></td>
<td>CA</td>
</tr>
</tbody>
</table>

3. Use shell scripting or a text editor to apply a prefix to the Contact ID values that were exported out of Oracle Service Cloud. The required prefix is CON_. Thus a contact ID of 1036 will appear as CON_1036 as shown below:

- Before Transformation: The Contact ID value is 1036
- After Transformation: The Contact ID value is CON_1036.

4. After verifying the file, save it.
Importing Post-Processed Contacts Data into Oracle Sales Cloud

Importing data into Oracle Sales Cloud is a multi-step process. In this step you set up the mapping that will be used when the transformed data is imported. Generally this is done one time.

1. In Oracle Sales Cloud, click Setup and Maintenance.
2. In the Search: Tasks area, search for Manage File Import Mappings.
3. In the Search Results field, click the Go to Task icon.
4. On the Manage File Import Mappings page, click the New button to create a new mapping.
5. In the Create Import Mapping dialog box, provide the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Mapping</td>
<td>Provide a name.</td>
</tr>
<tr>
<td>Object</td>
<td>Contact</td>
</tr>
</tbody>
</table>

6. Click Save and Close.
7. In the Manage File Import Mappings page, click the name of the newly created mapping link.
8. On the Edit Import Mapping page, click the Add (+) icon and provide the following information:

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Object</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact ID</td>
<td>PersonProfile</td>
<td>PartyOrigSystemRefNo</td>
</tr>
<tr>
<td>Organization ID</td>
<td>ContactRelationship</td>
<td>ObjOrigSystemRefNo</td>
</tr>
<tr>
<td>First Name</td>
<td>PersonProfile</td>
<td>PersonFirstName</td>
</tr>
<tr>
<td>Last Name</td>
<td>PersonProfile</td>
<td>PersonLastName</td>
</tr>
<tr>
<td>Street</td>
<td>Address</td>
<td>Address1</td>
</tr>
<tr>
<td>City</td>
<td>Address</td>
<td>City</td>
</tr>
<tr>
<td>State</td>
<td>Address</td>
<td>State</td>
</tr>
<tr>
<td>Province</td>
<td>SellToAddress</td>
<td>Province</td>
</tr>
<tr>
<td>Postal Code</td>
<td>Address</td>
<td>PostalCode</td>
</tr>
<tr>
<td>Country</td>
<td>Address</td>
<td>Country</td>
</tr>
</tbody>
</table>
Click Save and Close.

The mapping is complete. You can reuse this task to insert new or update existing rows into the Account object in Oracle Sales Cloud. This mapping can be used repeatedly during the actual import process.

10. In the Search: Tasks area, enter Manage File Import Activities and click Search.

11. In the Search Results, select the task, then click the Go to Task icon.

12. In the Manage Import Activities page, click the New icon.

13. In the Manage File Import Objects wizard, specify the following information:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name for the import activity.</td>
</tr>
<tr>
<td>Object</td>
<td>Contact</td>
</tr>
<tr>
<td>Upload From</td>
<td>Select the Desktop button.</td>
</tr>
<tr>
<td>Import Mapping</td>
<td>Click the drop-down list, and select the newly created CSV file.</td>
</tr>
</tbody>
</table>

14. Click Next.

The Edit Import Activity: Map Fields page appears. This page displays the mapping of the attributes to the transformed data.

15. Under the Set Constant Values area, specify the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContactRelationship</td>
<td>ObjOrigSystem</td>
<td>RNOW</td>
</tr>
<tr>
<td>PersonProfile</td>
<td>PartyOrigSystem</td>
<td>RNOW</td>
</tr>
</tbody>
</table>

16. Click Next, then in the Create Schedule page, choose Immediate from the Schedule drop-down list, then click Next.

17. In the Review and Activate page, confirm the accuracy of all the information, then click Activate.

18. In the Manage File Import Objects page, monitor the progress from Scheduled to Completed.

19. If the status indicates Completed with Errors, then one or more errors occurred during the import process.

20. Refer to the appropriate log to see which data failed. View the log file by opening the attachment associated with the File Import object item.

21. After the process completes, navigate to Sales, then choose Contacts and verify that Contacts were successfully created and associated with the correct Organization (if necessary).
Importing Contact PartyID Values into Oracle Service Cloud

After you have verified that the Contacts have been successfully created in Oracle Sales Cloud, the next step is to update the original Contact records (in Oracle Service Cloud) with the PartyID that was assigned by Oracle Sales Cloud. This finishes the process, and both applications (Oracle Sales Cloud and Oracle Service Cloud) have a mapping of unique IDs assigned by both applications.

In this task you perform one transformation on the Contacts data, shown in the following task, and that is remove the CON_prefix from the ContactID value.

You can use as a reference implementation, the Contact.sh script, located in the following folder:

Scripts/Bulk_Loading/Contact/SVC_OSC/ folder

Run the script as follows:

For Linux:

Convert the mode of the file to executable through the following command:

```bash
chmod 755 Contact.sh
```

Then execute the script by typing the following:

```bash
./Contact.sh
```

For Windows:

Execute the script by entering the following:

```bash
Contact.sh
```

The script begins executing and prompts you to enter certain parameters which are listed below. After supplying the parameters, the data is transformed and committed to the ContactUpdated.csv file.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename</td>
<td>File to be transformed. The file must be in the same folder.</td>
</tr>
<tr>
<td>Delimiter</td>
<td>Character used as a delimiter in the file to be transformed. Since comma is usually part of most data, Oracle recommends that a character other than comma or space be used as a delimiter (for example: ~).</td>
</tr>
<tr>
<td>ColumnIndex for Contact</td>
<td>The index of ContactID column in your file. The first column is 1.</td>
</tr>
<tr>
<td>Direction</td>
<td>Oracle Sales Cloud to Oracle Service Cloud.</td>
</tr>
</tbody>
</table>

Importing Contact PartyID Values into Oracle Service Cloud

1. View unique IDs in Oracle Sales Cloud for each Contact imported by navigating to the View Import Status page.
2. In the Attachments column, click the appropriate CSV link, and download the file to your local computer.
The CSV file contains the unique IDs assigned by Oracle Sales Cloud to each Contact that was successfully created. The IDs appear in the ObjectKey column.

3. Perform one final data transformation. In the Contact ID column, you must now remove the CON_ prefix.

For example, what appears as CON_1036 must be transformed to 1036. Use either shell scripting or a text editor to do this transformation. See the following example:

Before transformation the Contact ID value is 1036.

After transformation the Contact ID value is CON_1036.

4. Once finished with the transformation, log in to Oracle Service Cloud (though the CX Console) and from the Navigation area, choose Configuration, Database, then Data Import Wizard.

5. In the Data Import Wizard, enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Record Type</td>
<td>Contact</td>
</tr>
<tr>
<td>Header Location</td>
<td>Click the drop-down list arrow and select First Line of Data File.</td>
</tr>
<tr>
<td>Duplicate Records</td>
<td>Update Existing Data</td>
</tr>
</tbody>
</table>

6. Click Next, then confirm that all information in the Column in File column conforms to the data exported from Oracle Sales Cloud, then click Next.

7. In the Column Mappings area, remove all rows except for the following:
   - ObjectKey
   - Contact ID

Remove rows by selecting a row, then clicking the Remove icon.

8. Click Next, and view the prescan data file, then click Next.

The actual import is completed and the appropriate Organizations updated with the PartyID (referred to as the Fusion Party Reference ID) assigned by Oracle Sales Cloud. Ensure that the record counts are accurate and as expected.