

# Oracle® Cloud

## Sync Customer and Campaign Data Between Oracle Unity and Salesforce Marketing Cloud



F73275-02  
April 2024



Oracle Cloud Sync Customer and Campaign Data Between Oracle Unity and Salesforce Marketing Cloud,  
F73275-02

Copyright © 2022, 2024, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

# Contents

## 1 About This Recipe

---

Overview	1-1
System and Access Requirements	1-1
Assumptions	1-2

## 2 Before You Install the Recipe

---

Configure Oracle Unity	2-1
Access Your FTP Server and Create a Directory	2-1
Configure Salesforce Marketing Cloud	2-1

## 3 Install and Configure the Recipe

---

Configure the Oracle Unity Connection	3-1
Configure the Oracle FTP Connection	3-2
Configure the Oracle REST SFMC Connection	3-2
Configure the Oracle SOAP SFMC Connection	3-3
Configure the Oracle REST OIC Invoke Connection	3-3
Configure the Oracle REST OIC Trigger Connection	3-4
Configure the Lookup Tables	3-4

## 4 Activate and Run the Recipe

---

### A Appendix: Add or Edit the Attributes in Salesforce Marketing Cloud (Optional)

---

### B Appendix: Add or Delete Attributes in Oracle Unity (Optional)

---

## C Appendix: Add a New Email Campaign Event (Optional)

---

# Preface

This document describes how to install, configure, and run this recipe in Oracle Integration 3.

## Topics:

- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Related Resources](#)
- [Conventions](#)

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

## Related Resources

For more information, see these Oracle resources:

- Oracle Integration documentation in the Oracle Cloud Library on the Oracle Help Center.
- Oracle Cloud at <http://cloud.oracle.com>.

## Conventions

The following text conventions are used in this document.

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

# 1

## About This Recipe

Use this recipe to synchronize customer segments and campaign results between Oracle Unity and Salesforce Marketing Cloud.



### Note:

This recipe is available as **Oracle Unity — Salesforce Marketing Cloud | Sync Customer and Campaign Data** in the Integration Store. Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

## Overview

This recipe synchronizes customer segments and campaign results between Oracle Unity and Salesforce Marketing Cloud. Initially, it exports customer segments from Oracle Unity to Salesforce Marketing Cloud on a scheduled basis. Further, this recipe synchronizes campaign results data between the two systems.

To use the recipe, you must install the recipe and configure the connections and other resources within it.

Initially, you can activate and run the first scheduled integration flow of the recipe (**Oracle Unity FTP Master Customer Export**) manually or specify an execution schedule for it. When triggered, the integration flow fetches a customer segment (with multiple customer records) from Oracle Unity and exports it to a specific location in FTP. Subsequently, run the second scheduled integration flow of the recipe (**Oracle FTP SFMC Data Extension Import**), which reads the customer segment from the FTP location and imports it to a data extension created in Salesforce Marketing Cloud. Finally, run the app-driven integration flow (**Oracle SFMC Automation Setup**) and the last scheduled integration flow of the recipe (**Oracle SFMC Unity Campaign Results Import**) to synchronize the campaign results data between Salesforce Marketing Cloud and Oracle Unity.

Basic data, such as name, email, note, phone, and address, are synchronized between the Oracle Unity and Salesforce Marketing Cloud platforms.

## System and Access Requirements

- Oracle Integration, Version 21.4.3.0.0 or higher
- Oracle Unity
- An account on Oracle Unity with the Administrator role
- Salesforce Marketing Cloud
- An account on Salesforce Marketing Cloud with the Administrator role
- A secure FTP (sFTP) server or File Server

- An FTP client to access the sFTP server

## Assumptions

- Salesforce Marketing API request quota is sufficient to support the number of create and update contact requests from the batch integration.
- Oracle Unity has the customer segment with the required personalization attributes.
- Salesforce Marketing Cloud has the customer data extension already created to which the Oracle Unity customer details must be imported.



# 2

## Before You Install the Recipe

You must perform the following configuration tasks on your Oracle Unity and Salesforce Marketing Cloud instances in order to successfully connect to these external systems using Oracle Integration and synchronize customer and campaign data between them.

### Configure Oracle Unity

Complete the following tasks in your Oracle Unity instance to successfully connect to it from Oracle Integration and export customer segments.

1. Obtain the client ID and client secret. See Prerequisites for Creating a Connection.
2. Create master customer segments. See [Creating Segments](#).
3. Add the following personalization attributes to your master customer segments. See [Managing Personalization Attributes](#).
  - ID, First Name, Last Name, Email, Phone, Country, Age, Gender, Ok To Email, and Ok To Text.

### Access Your FTP Server and Create a Directory

Obtain an sFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory on the server to import and archive the files. Note the path of this directory.

### Configure Salesforce Marketing Cloud

To access Salesforce Marketing Cloud from Oracle Integration and import customer segments, you must perform certain configurations on your Salesforce Marketing Cloud instance.

- Obtain the client ID and client secret for a new package.
- Obtain the client ID and client secret for an existing package.
- Create an import data extension and obtain the external key.
- Obtain the SOAP WSDL endpoint, username, and password.

Log in to your Salesforce Marketing Cloud instance as an **Administrator** and execute the following tasks.

1. Obtain the client ID and client secret for a new package.
  - a. Click **Setup**, select **Apps**, and then select **Installed Packages**.
  - b. Click **New** to create a new package.

- c. In the New Package Details window, enter the name and description for the package and click **Add Component**.
        - d. Select **API Integration** as the component type and click **Next**.
        - e. Select **Server-to-Server** as the integration type and click **Next**.
        - f. In Server-to-Server properties, select **Read and Write Properties** for the Data Extensions property.
        - g. Click **Save**.  
The Components section displays the **Client ID** and **Client Secret** for the package. Note these values.
2. Obtain the client ID and client secret for an existing package.
  - a. Click **Setup**, select **Apps**, and then select **Installed Packages**. The Installed Packages page lists all the packages installed in the Salesforce Marketing Cloud account.
  - b. Select a package and navigate to the Components section.
    - i. To add a new component, click **Add Component**.
    - ii. To edit an existing component, click **Edit**.  
The Components section displays the **Client ID** and **Client Secret** for the existing package. Note these values.
  - c. Click **Save**.
3. Create an import data extension and obtain the external key.
  - a. In **Audience Builder**, choose **Contact Builder**.
  - b. In the Contact Builder window, navigate to the **Data Extensions** tab, and click **Create**.
  - c. Select the appropriate data extension method in the **Creation Method** drop-down menu.
    - i. **Create from New** – To create a data extension and manually insert all attributes.
    - ii. **Create from Existing** – To create a data extension and include all attributes assigned through the existing data extension or add attributes later in the creation process.
    - iii. **Create from Template** – To create a data extension and include all attributes assigned in the template or add attributes later in the creation process.
  - d. Enter a name for the data extension.
  - e. Note the value in the **External Key** field. Each data extension uses a unique external key.
  - f. Enter a brief description of the data extension.
  - g. Click **Change Location** and choose the location for storing the data extension. Choose an existing folder or create a new folder.
  - h. Select the **Is Sendable?** check box to make this data extension available as the source.
  - i. Select the **Is Testable?** check box to make this data extension available for testing and click **Next**.

- j. Set the data retention policy as **OFF**, then click **Next**.
- k. Create attribute fields to match the fields with Oracle Unity. For example, `Unity_MasterCustomer_ID`, `FirstName`, `LastName`, `Email`, `Phone`, `Country`, `Age`, `Gender`, `OkToEmail`, and `OkToText`.

 **Note:**

For the `Unity_MasterCustomer_ID` attribute, select the **Primary Key** check box.

For each attribute:

- Enter a name and choose a data type.
- If the attribute must contain a value for every contact, choose **Required**.
- Enter the maximum number of characters the attribute can have in the **Length** field.
- If the attribute must have a default value, enter the value in the **Default Value** text field.
- For sendable data extensions, map the **Send relationship** from `Unity_MasterCustomer_ID` data extension field to the **Subscriber Key**.

 **Note:**

Attributes assigned as a primary key do not contain default values.


- l. Click **Complete**.
4. Obtain the SOAP WSDL endpoint, username, and password. To do so, you must create a user and assign role and permissions.
    - a. Create a user.
      - i. Click **Setup**, select **Users**, and then **Users**.
      - ii. On the Setup Users page, click **Create**.
      - iii. On the New User page:
        - Enter the name of the user.
        - Enter the email ID of the user in **Reply Email Address** field and verify it.
        - Enter the email address to which run-time exception emails are sent in the **Notification Email Address** field.
        - Enter the username and note the same.
        - Select the appropriate **Time Zone** and **Culture Code**.
        - Enable the **API User** option.
        - Note the **SOAP WSDL URL**.
        - Set a password and note the same.

- b. Assign the required role and permissions to the user.
  - i. On the Setup Users page, select the user you created previously.
  - ii. Click **Manage Roles**.
  - iii. Select whether you assign the role directly to the user or in the context of a specific business unit in the **Business Unit** drop-down menu.
  - iv. Click **Edit Roles**. Select the check box next to the appropriate roles to assign.
  - v. Click **Save**.
  - vi. Click **Edit Permissions** to update specific permissions for the user and select the check boxes next to permissions to assign.
  - vii. Click **Save**.

# 3

## Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

1. On the Oracle Integration Home page, in the **Use a recipe/accelerator** section, click **View all**.
2. Find the recipe you want to install, then click **Get**.  
A message confirms that the recipe was successfully installed, and the recipe card shows **In use**.
3. Click **Configure**  on the recipe to configure its resources.  
The project workspace opens, displaying all the resources of the recipe. Configure the following resources before you activate and run the recipe.


## Configure the Oracle Unity Connection

1. In the Connections section, click the connection name.
2. In the **Oracle Unity URL** field, enter the URL for connecting to Oracle Unity. For example, `https://XXXXXX.cxunity.ocs.oraclecloud.com`.  
The URL is provided when you subscribe to Oracle Unity.
3. In the Security section, enter the following details:

Field	Information to Enter
<b>Security Policy</b>	Select <b>Resource Owner Password Credentials</b> .
<b>Access Token URI</b>	Enter the authorization server that generates the access token: <code>https://IDCS_URL/oauth2/v1/token</code>
<b>Client ID</b>	Enter the unique random string that matches the API console project. See <a href="#">Configure Oracle Unity</a> .
<b>Client Secret</b>	Enter the unique random string that matches the API console project.
<b>Username</b>	Enter the Oracle Identity Cloud Service username of the OAuth token request.
<b>Password</b>	Enter the Oracle Identity Cloud Service password of the OAuth token request.
<b>Scope</b>	Enter the list of authorization permissions for the target application. For example, <code>read write</code> .

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

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

## Configure the Oracle FTP Connection

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


Field	Information to Enter
<b>FTP Server Host Address</b>	Enter the host address of your sFTP server.
<b>FTP Server Port</b>	Enter 22.
<b>SFTP Connection</b>	Select <b>Yes</b> from the list.

- In the Security section, enter the following details:

Field	Information to Enter
<b>Security Policy</b>	Select <b>FTP Server Access Policy</b> .
<b>User Name</b>	Enter the user name to connect to your sFTP server.
<b>Password</b>	Enter the password to connect to your sFTP server.

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

A message confirms if your test is successful.

- To return to the project workspace, click **Go back** .

## Configure the Oracle REST SFMC Connection


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

Field	Information to Enter
<b>Connection Type</b>	Select <b>REST API Base URL</b> .
<b>Connection URL</b>	Paste the URL without trailing (/) copied from Salesforce Marketing Cloud. See <a href="#">Configure Salesforce Marketing Cloud</a> .

- In the Security section, enter the following details:

Field	Information to Enter
<b>Security Policy</b>	Select <b>OAuth Custom Two-Legged Flow</b> .


Field	Information to Enter
<b>Access Token Request</b>	<p>Enter the client id, client secret, account id and auth url copied from Salesforce Marketing Cloud in the below format. Append v2/token in the auth url. To generate client id and secret, see <a href="#">Configure Salesforce Marketing Cloud</a>.</p> <pre>-X POST -H 'Content-Type:application/json' -d '{ "grant_type": "client_credentials", "client_id": "&lt;client id&gt;", "client_secret": "&lt;client secret&gt;", "account_id": "&lt;account id&gt;" }' https://mc4nwk3rs1n1tm8lm4qgd1g448hm.auth.marketingcloudapis.com/v2/token'</pre>

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.  
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

## Configure the Oracle SOAP SFMC Connection

1. In the Connections section, click the connection name.
2. In the Properties section, enter the SOAP WSDL URL in the **WSDL URL** field. See [Configure Salesforce Marketing Cloud](#).
3. In the Security section, enter the following details:

Field	Information to Enter
<b>Security Policy</b>	Select <b>Username Password Token</b> .
<b>Username</b>	Enter the Username. See <a href="#">Configure Salesforce Marketing Cloud</a> .
<b>Password</b>	Enter the Password.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.  
A message confirms if your test is successful.
6. To return to the project workspace, click **Go back** .

## Configure the Oracle REST OIC Invoke Connection

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


Field	Information to Enter
<b>Connection Type</b>	Select <b>REST API Base URL</b> .
<b>Connection URL</b>	Paste the URL of the Oracle Integration instance in the below format.

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

Field	Information to Enter
<b>Security Policy</b>	Select <b>Basic Authentication</b> .
<b>Username</b>	Enter the Username of the Oracle Integration instance.
<b>Password</b>	Enter the Password of the Oracle Integration instance.

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


A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** .

## Configure the Oracle REST OIC Trigger Connection

1. In the Connections section, click the connection name.
2. In the Security section, select the Security Policy as **OAuth 2.0 Or Basic Authentication**.
3. Click **Save**. If prompted, click **Save** again.
4. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

A message confirms if your test is successful.

5. To return to the project workspace, click **Go back** .

## Configure the Lookup Tables


The recipe contains the following four lookup tables. Edit the lookup tables to enter appropriate values for the lookup keys.

- **ORACLE-BRT-UNITY\_SFMC\_PROPERTIES**: Used to configure the FTP file details in which the customer segments are stored.

Key	Description	Example
<b>FileDirectory</b>	Stores the location of the FTP file directory that has the customer segments.	/home/users/org-name / customers/inbound
<b>RetryCount</b>	Stores the number of retries when a transient error occurs.	3



Key	Description	Example
<b>ArchiveEnabled</b>	Stores the flag to identify if the file must be archived after being processed successfully.	Y
<b>ArchiveDirectory</b>	Stores the location of the FTP directory that stores the archived files with customer segments.	/home/users/org-name / customers/archive
<b>CustomerDataExtension Key</b>	Stores the ID of customer data extension in Salesforce Marketing Cloud. It is a unique value that identifies the data extension and is used to identify the data extension with an API call.	5C2EF345-4A12-405C-BDCE-BE358139DC44
<b>SFMCPageSize</b>	Stores the ideal pagination value for the <b>Oracle FTP SFMC DataExtension Import</b> integration.	250
<b>EmailFrom_Success</b>	Stores the email address from which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	no-reply@company.com
<b>EmailTo_Success</b>	Stores the email address to which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	xyz@company.com
<b>EmailSubject_Success</b>	Stores the subject of the email that is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	Unity to SFMC Customer Import Successful
<b>EmailFrom_Failure</b>	Stores the email address from which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	no-reply@company.com
<b>EmailTo_Failure</b>	Stores the email address to which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	xyz@company.com
<b>EmailSubject_Failure</b>	Stores the subject of the email that is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	Unity to SFMC Customer Import Failure
<b>MaxFileCount</b>	Stores the maximum number of files that can be processed in one instance. Ensure that the processing time doesn't exceed six hours limit for a scheduled instance. The maximum possible value is 100.	100

Key	Description	Example
<b>MaxInstanceCount</b>	Stores the maximum number of instances that a run can span across in case the processing is not completed in one instance.	10
 <b>Note:</b> Ensure that processing is complete before next scheduled run.		
<b>ClearBeforeImport</b>	Stores the flag to identify if recipe clears the old data before importing the new data in the customer import data extension.	Y
<b>CustomerSegmentName</b>	Stores the Customer Segment Name.	Unity_SFMC_MasterCustomer
<b>UnityTenantId</b>	Stores the Unity Tenant Id.	100046

- **ORACLE-BRT-SFMC\_UNITY\_PROPERTIES:** Used to configure the data extension that stores the campaign results in Salesforce Marketing Cloud.

Key	Description	Example
<b>ResultsDataExtensionKey</b>	Stores the external key of the results data extension in Salesforce Marketing Cloud. This value is passed in the <b>Automation Setup</b> integration.	ResultsDataExtensionName
<b>ResultsDataExtensionName</b>	Stores the name of the results data extension in Salesforce Marketing Cloud. This value is passed in the <b>Automation Setup</b> integration.	Unity_CampaignResults_DE
<b>MarketingMedium</b>	Stores the medium of the marketing campaign.	Email
<b>MarketingSource</b>	Stores the source application of the marketing campaign.	Salesforce Marketing Cloud
<b>EmailFrom_Success</b>	Stores the email address from which a message is sent whenever the campaign results import from Salesforce Marketing Cloud to Oracle Unity is successful.	no-reply@company.com
<b>EmailTo_Success</b>	Stores the email address to which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	xyz@company.com
<b>EmailSubject_Success</b>	Stores the subject of the email that is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is successful.	Unity to SFMC Customer Import Successful

Key	Description	Example
<b>EmailFrom_Failure</b>	Stores the email address from which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	no-reply@company.com
<b>EmailTo_Failure</b>	Stores the email address to which a message is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	xyz@company.com
<b>EmailSubject_Failure</b>	Stores the subject of the email that is sent whenever the customer import from Oracle Unity to Salesforce Marketing Cloud is unsuccessful.	Unity to SFMC Customer Import Failure
<b>UnityPageSize</b>	Stores the maximum number of records that can be passed to Oracle Unity in an invoke request.	500
<b>UnityRetryCount</b>	Stores the maximum number of retries in case Oracle Unity invoke request fails.	3
<b>MaxIterationCount</b>	Stores the maximum number of Salesforce Marketing Cloud iterations with an iteration size of 2500 records. Ensure that the processing time doesn't exceed six hours limit for a scheduled instance.	100
<b>MaxInstanceCount</b>	Stores the maximum number of instances that a run can span across in case the processing is incomplete in one instance.	20


 **Note:**

Ensure that the processing gets completed before next scheduled run.

- **ORACLE-BRT-SFMC\_UNITY\_EVENT\_TYPE:** Represents the mapping between event types in the instances. For instance, SentEvent is mapped with Sent.

Event Type in Salesforce Marketing Cloud	Event Type in Oracle Unity
<b>SentEvent</b>	Sent
<b>OpenEvent</b>	Opened
<b>ClickEvent</b>	Clicked
<b>BounceEvent</b>	Bounced


- **ORACLE-BRT-SFMC\_UNITY\_MONTH\_OF\_YEAR:** Used to map the calendar months between the instances.

1. In the Lookups section, click the lookup name.
2. Edit the lookup table as necessary.
3. Click **Save**. If prompted, click **Save** again.
4. To return to the project workspace, click **Go back** .

# 4


## Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. In the project workspace, click **Activate**. In the Activate project panel, with the default project deployment selected, choose an appropriate tracing option, then click **Activate**.  
A message confirms that the integration has been activated. Refresh the page to view the updated status of the integration.
2. Run the recipe.
  - a. Export a customer segment from Oracle Unity to an FTP server.
    - i. In the Integrations section of the project workspace, click **Actions**  on the **Oracle Unity FTP MasterCustomer Export** integration flow, then select **Run**.
    - ii. On the Configure and run page, click **Run**.  
The integration gets triggered, and the customer segment records are exported to the FTP server.

 **Note:**

You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

- b. Import a customer segment from the FTP server to Salesforce Marketing Cloud.
  - i. In the Integrations section of the project workspace, click **Actions**  on the **Oracle FTP SFMC DataExtension Import** integration flow, then select **Run**.  
The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.
    - **continueInstance**: This parameter stores the flag to identify a new instance. **Y** indicates continued process from previous instance. **N** indicates a new instance and is the default value.
    - **InstanceNo**: This parameter stores the count of the running instance. The default value is 1.
    - **triggerTimeStamp**: This parameter stores the created time stamp of the trigger file that started the instance.
  - ii. On the Configure and run page, click **Run**.  
The integration gets triggered, the customer segment data is imported to Salesforce Marketing Cloud data extension.

 **Note:**

You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

- c. Run the **Oracle SFMC Automation Setup** integration flow to create the initial customization required in Salesforce Marketing Cloud. This is a one-time setup that configures the automation that queries and stores campaign results in a data extension.

- i. In the Integrations section of the project workspace, click **Actions** **...** on the integration flow, and select **Run**, then click **Test**.

The page to test the integration with a sample file is displayed.

- ii. In the Request section of the test page, on the **Body** tab, provide the following parameters:

Name	Description	Example
<b>customerDataExtensionName</b>	Stores the name of the data extension to which customer segments are imported from Oracle Unity to Salesforce Marketing Cloud.	Unity_MasterCustomer_DE
<b>automationKey</b>	Stores the external key of the automation created in Salesforce Marketing Cloud.	Unity_CampaignResults_Automation
<b>automationName</b>	Stores the name of the automation created in Salesforce Marketing Cloud.	Unity_CampaignResults_Automation
<b>resultsDataExtensionKey</b>	Stores the external key of the data extension with the campaign result events in Salesforce Marketing Cloud.	Unity_CampaignResults_DE
<b>resultsDataExtensionName</b>	Stores the name of the data extension with the campaign result events in Salesforce Marketing Cloud.	Unity_CampaignResults_D
<b>automationScheduleStartTimeDateTime</b>	Stores the DateTimeStamp when the automation schedule starts in CST Time Zone.	YYYY-MM-DDT00:00:00
<b>automationScheduleOccurrences</b>	Stores the number of occurrences of the automation schedule runs.	1000
<b>automationScheduleInterval</b>	Stores the number of days interval between two occurrences of the scheduled run.	1

 **Note:**

The automation runs on a daily schedule.

- iii. Click **Test**.

- d. Synchronize campaign results data from Salesforce Marketing Cloud to Oracle Unity.

- i. In the Integrations section of the project workspace, click **Actions** **...** on the **Oracle SFMC Unity CampaignResults Import** integration flow, then select **Run**.

The Configure and run page is displayed with the following parameters. Update the parameters with appropriate values.

- **continueInstance:** This parameter stores the flag to identify a new instance. **Y** indicates continued process from previous instance. **N** indicates a new instance and is the default value.
  - **InstanceNo:** This parameter stores the count of the running instance. The default value is 1.
  - **continueRequestId:** This parameter stores the Request ID of the Salesforce Marketing Cloud request which needs to be continued in upcoming instances.
- ii. On the Configure and run page, click **Run**.  
The integration gets triggered, and the email campaign results are imported to Oracle Unity.

 **Note:**

You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

3. Monitor the running of the integration flow in Oracle Integration.
  - a. In the project workspace, click **Observe**. You'll see the integration flow being triggered and running successfully.
  - b. To manage errors in your project, see Manage Errors in a Project.
4. Log in to Oracle Unity and check for the new campaign results imported.

**Related Documentation**

- *Using the Oracle Unity Adapter with Oracle Integration 3*
- *Using the FTP Adapter with Oracle Integration 3*
- *Using the Salesforce Adapter with Oracle Integration 3*

# A

## Appendix: Add or Edit the Attributes in Salesforce Marketing Cloud (Optional)

To add or edit attributes in Salesforce Marketing Cloud (other than what is pre-configured in this recipe), you must update the data extension you created previously and update the schema in REST endpoint configuration of the **Oracle FTP SFMC DataExtension Import** integration flow.

### Update the Data Extension in Salesforce Marketing Cloud

1. Log in to Salesforce Marketing Cloud.
2. In **Audience Builder**, select **Contact Builder**.
3. In the Contact Builder window, navigate to the **Data Extensions** tab.
4. Select the data extension you created previously from the list, then click **Open**.
5. To add a new attribute, click **Create Attribute**.
6. To edit an attribute, click the **Edit** icon next to the attribute.

 **Note:**

You cannot select the data type while editing the attribute.

7. Click **Save**.

### Update the Schema in REST Endpoint Configuration

1. In the Integrations section, click the **Oracle FTP SFMC DataExtension Import** integration flow.
2. On the integration canvas, expand **FileScope**, then expand **PageScope**.
3. In PageScope, select the **UpsertInSFMCDataExtension** element, and then click **Actions . . .** and select **Edit**.
4. On the Request page, update the JSON schema with the new fields.
5. On the Summary page, click **Done**.
6. In the resulting window, click **Update**.  
The updated attribute appears as a separate field on the mapping canvas.

### Update the REST Endpoint's Mapper

Map the new attribute to the relevant source attribute on the mapping canvas.

1. On the integration canvas, in **PageScope**, select the **Map to UpsertInSFMCDataExtension** element, then click **Actions . . .** and select **Edit** to open the mapper.
2. Map the new attribute to the relevant source attribute on the mapping canvas.



# B

## Appendix: Add or Delete Attributes in Oracle Unity (Optional)

To add or delete attributes in Oracle Unity (other than what is pre-configured in this recipe), you must perform the following tasks:

1. Add or delete the personalization attributes in Oracle Unity.
2. In the **Oracle FTP SFMC DataExtension Import** integration flow of the recipe:
  - a. Update the FTP adapter with the new file schema.
  - b. Update the mapper element corresponding to the FTP adapter.
  - c. Update the Stage File element with the new file schema.
  - d. Update the mapper element corresponding to the Stage File element.
  - e. Update the Stage File element in the fault handler with the new file schema.
  - f. Update the mapper element corresponding to the Stage File element in the fault handler.

### Add or Delete the Personalization Attributes in Oracle Unity

To add or delete the personalization attributes (other than what is pre-configured in this recipe) from the base object to the segment, see [Managing Personalization Attributes](#).

### Update the FTP Adapter with the New Schema



#### Note:

You get the updated file schema when you run the **Oracle Unity FTP Master Customer Export** integration flow.

1. In the Integrations section, click the **Oracle FTP SFMC DataExtension Import** integration flow.
2. On the integration canvas, expand the **FileScope** element, select the **ReadFileFromFTP** element, and then click **Actions** and select **Edit**.
3. Upload the updated schema.
4. On the Summary page, click **Done**.
5. In the resulting window, click **Update**.  
The updated attribute appears as a separate field on the mapping canvas.

### Update the FTP Adapter's Mapper

1. On the integration canvas, in **FileScope**, select the **Map to ReadFileFromFTP** element, and then click **Actions** and select **Edit** to open the mapper.
2. Map the new attribute to the relevant target attribute on the mapping canvas.

### Update the Stage File Element with the New Schema

1. On the integration canvas, in **FileScope**, select the **InitializeStageErrorFile** element, and then click **Actions** **...** and select **Edit**.
2. Upload the updated schema.
3. On the Summary page, click **Done**.
4. In the resulting window, click **Update**.  
The updated attribute appears as a separate field on the mapping canvas.  
To know more about configuring stage file activity, see [Configure a Stage File Action](#).

### Update the Stage File Element's Mapper

1. On the integration canvas, in **FileScope**, select the **Map to InitializeStageErrorFile** element, and then click **Actions** **...** and select **Edit** to open the mapper.
2. Map the new attribute to the relevant target attribute on the mapping canvas.

### Update the Fault Handler's Stage File Element with the New Schema

1. On the integration canvas, expand **FileScope**, and then expand **PageScope**.
2. In **PageScope**, click the **Fault Handler** element, and select **Default Handler**.
3. Select the **AppendToStageErrorFile** element, and then click **Actions** **...** and select **Edit**.
4. On the Format Definition page, upload the updated schema.
5. On the Summary page, click **Done**.
6. In the resulting window, click **Update**.  
The updated attribute appears as a separate field on the mapping canvas.  
To know more about configuring stage file activity, see [Configure a Stage File Action](#).

### Update the Stage File Element's Mapper

1. On the integration canvas, within **DefaultHandler**, select the **Map to AppendToStageErrorFile** element, and then click **Actions** **...** and select **Edit** to open the mapper.
2. Map the new attribute to the relevant target attribute on the mapping canvas.

# C

## Appendix: Add a New Email Campaign Event (Optional)

In this recipe, the following email campaign events are pre-configured:

- Sent Event
- Bounce Event
- Click Event
- Open Event

To add a new email campaign event other than what is pre-configured in the recipe, you must:

1. Add a new event in the lookup table **ORACLE-BRT-SFMC\_UNITY\_EVENT\_TYPE**. See [Configure the Lookup Tables](#).
2. In the Integration section, click the **Oracle SFMC Automation Setup** integration flow and perform the following tasks on the integration canvas.
  - a. Assign global variables for the new event.
    - i. On the integration canvas, select the **AssignGlobalVariables** element, and then click **Actions** ⋮ and select **Edit**.
    - ii. Add new variables for the event.
  - b. Add a SOAP adapter endpoint and a mapper for the new event.
    - i. In the **Main Scope**, expand **DataExtensions Scope**.
    - ii. Add a SOAP adapter endpoint along with a mapper element.
    - iii. Update the mapper with the details specific to the new event.
  - c. Add a status check for the data extension created.
    - i. In the **DataExtensions Scope**, select the switch case (**IF DataExtensionsCreated**) element, and then click **Actions** ⋮ and select **Edit**.
    - ii. Add a **StatusCode** for the new data extension created.
  - d. Add query definitions for the event.
    - i. In the **Main Scope**, expand **QueryDefinitions Scope**.
    - ii. Add a SOAP adapter endpoint and a mapper for a query definition to query events data from the Salesforce Marketing Cloud data view and import the same to the data extension created for the event.
    - iii. Update the query definition mapper with the details and query specific to the new event.
    - iv. Add another SOAP adapter endpoint and a mapper for a query definition to copy data from the event-specific data extension to the results data extension.
    - v. Update the query definition mapper with the details and query specific to the new event.

- 
- vi. Add switch case conditions and assign variables for the data extensions created.
    - i. In the **QueryDefinitions Scope**, select the switch case (**IF QueryDefinitionsCreated**) element, and then click **Actions . . .** and select **Edit**.
    - ii. Add a **StatusCheck** for the new data extension created.
    - iii. To assign variables, select the **AssignQueryDefinition** element, and then click **Actions . . .** and select **Edit**.
    - iv. Add a new variable in the similar format you added for other events.
  - e. In the **Automation Scope**, update the mapper with the query definition created.
    - i. In the **Main Scope**, expand **AutomationScope**.
    - ii. Select the **Map to createAutomation** element, and then click **Actions . . .** and select **Edit**.
    - iii. On the mapping canvas, add a new **AutomationTask** and map the new query definition created with the relevant target attribute.