Personalization: Create Audiences with Dynamic Customer Profiles

Skip Ahead

Key Terms
1. Create a DCP Service
2. Create a DCP Table
3. Define Table Attributes
4. Upload your data
5. Create an Audience
6. Build a Campaign

This article will help you:
- Connect your first and third-party data sources to Optimizely
- Create a single, unified view of your customers
- Customize Audience conditions based on behavioral targeting, as well as online and offline data about your customers

A Dynamic Customer Profile (DCP) is a single, actionable view of your customer built from a combination of first and third-party customer data. It enables you to deliver consistent, personalized web and mobile experiences for your most valuable customers, and measure the impact of those experiences.

Here are a few examples of the data sources that you can connect to Optimizely through DCP:
- Customer Relationship Management software
- Email Service Providers
- Data Warehouses
- Analytics platforms
• Point-of-Sale Systems

• Data Management Platforms

For example, use DCP in combination with behavioral targeting to create a personalized experience based on a visitor’s browsing behavior during a session, as well as frequent flyer account information, favorite airports, and average purchase amount.

Other first and third-party data that you can leverage through DCP include:

---

**Key Terms**

Below are a few key terms to help you get started. If attributes, data sources, and data management platforms are intuitive to you, jump straight into setup.

**Attributes**

Attributes describe what you know about your individual customers: their favorite products, lifetime value scores, VIP or loyalty memberships, gender, and other historical data. These attributes help you create tightly tailored experiences for every known customer who comes to your site.

**Data source**

A data source such as your CRM, Email Service Provider, data warehouse, or DMP stores individual customer attributes under corresponding customer IDs. To use these attributes to personalize your customers’ experiences on your site, create a DCP Table and corresponding Table Attributes.
DCP Table

A DCP Table is used to store records extracted from your data sources and make them available for targeting. Each row in the table represents a customer, and each column represents an attribute describing customers.

DCP Table Attribute

You can think of Table Attributes as the column headers in a DCP Table. Each attribute represents a discrete piece of information contained in the customer records extracted from your data source.

DCP Service

Optimizely’s DCP Service collects and handles the data that you use to deliver personalized messaging. The third-party data that you send to Optimizely is stored here.

Optimizely uses a method called “aliasing” to connect all the attributes kept in different tables to a single Optimizely User ID. This single ID allows Optimizely to deliver customized experiences based on the data you have on individual customers across different tables. Read more about aliasing in our developer docs.

Note:

Dynamic Customer Profiles are available with the Enterprise package. To learn more, check out Optimizely Personalization.

You can also read more about DCP on the Optimizely blog.

Important:

Remember, your terms of service prohibit you from collecting or sending any personally identifiable information (such as names, social security numbers, email addresses, or any similar data) to Optimizely's services or systems through Dynamic Customer Profiles or any other feature.

Please read the article on PII to learn more about sending data to Optimizely and handling personally identifiable information.

Currently even when we provision or upgrade an account to Personalization Professional, DCP is not automatically included. You need to separately whitelist DCP as a standalone feature by following the process outlined on the Wiki. The same is true for your own free employee account.

1. Create a DCP Service

To set up Dynamic Customer Profiles, first create a DCP Service in Optimizely. A DCP Service collects, stores, and processes the customer data that you’ll use to deliver personalized messaging.
1. Navigate to Account Settings.

2. Select the Dynamic Customer Profiles tab.

3. Name your new DCP Service > click Create DCP Service.

Note that each Optimizely Account can only be associated with one DCP Service, so you won't see the option to create a DCP Service if one already exists.

4. Link the Optimizely projects that should be able to target based on data stored in this DCP Service. You can add multiple projects to the same service at any time.
2. Create a DCP Table

Next, create a table to store records extracted from your data source.

1. Navigate to the Audiences dashboard.

2. Select the Attributes tab and click Create New Table.

3. Optimizely gives you a preview of the steps required to create a table. Click Create New Table.

4. Next, tell Optimizely how your table is organized. Fill in the name of your table and identify how your web page stores the customer IDs to be used in this table: in a cookie, JavaScript variable, query parameter, or Optimizely End User ID cookie.

In the example above, the customer IDs for the table are stored in a JS variable called “locator_name”.

Optimizely syncs the IDs in your new data source with Optimizely User IDs, so you don’t have to worry about managing identity. As your visitor data updates, Optimizely’s DCP Service automatically connects and adds this information to your customer profiles.
Sometimes, customer IDs are stored in multiple locations. For example, IDs may be stored in cookies on some pages and as JavaScript variables on others. In this situation, use our REST APIs to alias this information. Learn more about this process in our developer docs.

3. Define Table Attributes

Next, identify the customer attributes that you’d like Optimizely to store in this table. Is the attribute stored as text, a boolean value, a number, or a date? Define this value type in the drop-down menu.

Create New Table

Each row in your table represents a user, and each column represents an attribute about that user. You can use these attributes to build audiences and target experiences.

Configure Table  Add Attribute(s)  Upload Table Data

Add Attributes

Attributes are column headers in the table. These attributes will become available as audience conditions.

<table>
<thead>
<tr>
<th>Column Name*</th>
<th>Value Type</th>
<th>Description</th>
<th>Content-enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>customerid</td>
<td>Text</td>
<td>ID that uniquely identifies your customer</td>
<td>☑</td>
</tr>
</tbody>
</table>
| membership level | Text   | gold, silver or bronze | ☑   
| purchased in store | Boolean | | ☑   

Check the Content-enabled box next to any values that you’d like to be able to display directly in the browser. For example, if you’d like to show a visitor her “gold” level membership as part of a personalized experience on your site, check the Content-enabled box for the “membership level” attribute.
Above, we identify “membership level” and “purchased in store” as attributes we’ll use in Optimizely. We’ve also made the membership level value available for display in the browser.

Enter as many attributes as you like by clicking *Add an Attribute.*

**Note:**

Attribute names are case sensitive, so “membership level” and “Membership level” register as two different attributes.

### 4. Upload your data

There are three ways to upload data to your table. The **REST API** lets you stream the data, one visitor at a time. You can also upload data in bulk as a **CSV**. Every table is provisioned an Amazon S3 bucket. When you drop CSVs here, we will automatically begin uploading them into our system. You can also small files (5mb or less) in the browser via **Direct Upload.**

Learn how to [set up a streaming API](#) or the [bulk CSV upload](#) in our developer docs.

**Note:**

The Data Upload page for each table contains a link to download a pre-configured CSV file you can use as a template for uploads to that table. This is a great place to start if you're having trouble with the header formatting!

If you're still having trouble, make sure your CSV file is formatted as follows:
• Each column in the header row must be a registered attribute name. A CSV may contain a subset of the registered attributes.

• The header row must include a customerId column. All rows must contain a valid customer ID under this column.

• If a column header does not correspond to a registered attribute name, the upload will fail.

• If an attribute value does not respect the attribute's datatype/format, the upload will fail.

Check that your upload is successful by navigating to Audiences. Open the Attributes tab and select a table.

Click the Actions icon (…) and select Upload History.

The Data Upload History dialog provides a chronological view of your upload attempts.

If your latest upload wasn’t successful, check the formatting of your CSV and try again.

Common mistakes include:

• Uploading a .xlsx file instead of a .csv or .tsv

• Forgetting to register attributes before uploading data

• Not having a column labeled “customerId” - remember, ALL IDs must take this format

• Having “,” within an Attribute Value and saving the file as a .csv instead of .tsv

5. Create an Audience

Now, you’re ready to create an Audience using the customer attributes you’ve added to your DCP Table.

In the Audiences dashboard, click Create New Audience.
In the Audience Builder, expand the list of External Attributes and find the table you just created. To use a table attribute as an audience condition, drag and drop it into Audience Conditions. Use logical conditions to create your new Audience.

To learn more about building Audiences, check out this article.

6. Build a Campaign

Congratulations! You’re ready to create a personalization campaign with your new audience, powered by your dynamic customer profiles.

Learn more about creating a campaign in this article on campaign setup.

To add 1-to-1 personalization to your campaigns, read this article on exposing content attributes with DCP.