Audience conditions: Descriptions and examples

Skip Ahead

Available audience conditions in Optimizely X

This article will help you:

• Use Audience conditions to **customize who will be included in your experiments**, such as:
  ◦ Visitors who use certain **devices**, like phones or tablets
  ◦ Visitors in certain **locations**
  ◦ Visitors who come from certain **ad campaigns** or who use certain **query parameters**
  ◦ Visitors who have certain **cookies**, like logged-in

When you set up Audiences to target certain visitors in your experiment, you'll build those audiences using certain conditions. This article describes each audience condition, including common ways to use them to target certain visitors. You can also take a quick look at [how to create audiences in our Optimizely Academy](#).

Most default audience conditions aren't "sticky," meaning the condition is re-evaluated for the visitor on each new page load. For example, a visitor may meet the targeting conditions and be included in the experiment early on. If she returns later, maybe with a different browser or language setting that doesn't meet the experiment's audience conditions, she'll no longer be included in the experiment. The exceptions to this rule are the Ad Campaign and New/Returning Session conditions.

Available audience conditions vary based on your Optimizely plan type, so if you don't have access to an audience condition you need, talk to your Optimizely representative about changing plans.

For information about available audience conditions and other plan-specific features, please refer to our package matrices for [Optimizely X](#).
Available audience conditions in Optimizely X

Ad Campaign

The Ad Campaign condition allows you create audiences based on 'utm_campaign' campaign source types. Visitors who arrive with the parameter 'utm_campaign' will capture the parameter value as a dimension, up to a 20-character maximum (longer values will be truncated). This lets you drill down into campaign keywords such as “holiday_sale.”

You have several options for how to target certain ad campaigns:

- Choose **equals** if you want to target an exact 'utm_campaign' value
- Choose **contains** if you want to target any 'utm_campaign' value that contains the string that you enter (similar to **substring match** in URL targeting)
- Choose **matches** if you want to target a value using a **Regular Expression**
- Choose **has any value** if you want to target any ad campaign, regardless of what the value is

For example, if you want to include visitors from the "spring_15" 'utm_campaign' value, select Visitor **comes from** any of these ad campaigns and the **contains** targeting option, and enter spring_15 in the text field:

For details about how you can optimize based on ad campaigns, see our article on [optimizing SEM campaigns](#).

**Note:**

The *Ad Campaign* audience condition is "**sticky**." In other words, if a visitor lands on a page with a query parameter from an ad campaign and gets targeted in the experiment, the targeting will persist after the initial page view. This stickiness will last for the duration of the campaign. The *Ad Campaign* condition is ideal for tracking visitors across multiple pages.

After a visitor is initially bucketed into an experiment using the *Ad Campaign* condition, Optimizely will place this information in the browser's localStorage so that the visitor continues to be targeted, even after they move to other pages on the site.

Browser/Version

The Browser/Version condition targets your experiment based on a visitor's browser or browser version. You can include or exclude visitors who are using specific browsers or versions of browsers.

For example, suppose you want to target your experiment to visitors who are using Google Chrome. You can use the Browser/Version audience condition to include visitors who are browsing in any version of Google Chrome in the
experiment:

An alternative to targeting based on browser/version is **Custom JavaScript targeting**.

To target based on whether or not the visitor has a cookie from your site, enter the cookie’s name. This will check whether that cookie is set. You also have several options for how to target certain cookie values:

- **Choose equals** if you want to target an exact cookie value
- **Choose contains** if you want to target any cookie that contains the string that you enter (this is similar to substring match in URL targeting)
- **Choose matches** if you want to target a value using a **Regular Expression** (currently, Optimizely does not support flags)
- **Choose has any value** if you want to target based on the presence of a cookie, regardless of what the value is

Cookie targeting is case-sensitive for both names and values. If you need your cookie targeting to be case insensitive, please contact support or let us know in the [Optiverse Community](https://www.optimizely.com/community/). The most common use cases for this condition include setting audiences by their logged-in state, and creating a test cookie to preview your experiment.

For example, if you’re running a pop-up banner that highlights a new-user promotion. You want to run an experiment on this banner, but you also want to ensure that the experiment is only counting visitors for those who actually see the banner. You know that a cookie “seen_promo=true” is set on visitors when the pop-up appears.

First, identify how you’re going to differentiate between visitors who have and have-not seen the promotion. Select the **Cookie** condition, then Visitor **does not have** any of these cookies. Add the cookie name (seen_promo) in the left input field and the cookie value (true) in the right input field:

The experiment will only run for visitors who do not have the cookie “seen_promo=true”. This means your test will only
be seen by visitors who are seeing the promotion for the first time.

As another example, let's say you want to run an experiment with layout changes for customers who are logged in to your site. You know that logged-in customers have a cookie "logged_in=true".

First, identify how you are going to differentiate between logged-in and logged-out customers. Select the **Cookie** audience condition, then Visitor **has** any of these cookies. Add the cookie name (such as "loggedin") in the left input field and the cookie value (true) in the right input field:

The experiment will only run for visitors with the specified cookie that distinguishes them as being logged in.

In this case, a cookie is the differentiator, but in many cases, the "logged-in" state may be a variable on the page and you will need to use a **custom JavaScript condition** to do this. [Custom JavaScript](https://github.com/faisalman/ua-parser-js/tree/0.7.12)

Custom JavaScript conditions let you target complex and customized sets of visitors, based on conditions that you define. Just make sure that the condition evaluates to **true**. To learn more about Custom JavaScript, including use cases, see our **Custom JavaScript audience targeting article**.

**Device**

We use an open source javascript library called UAParser, [https://github.com/faisalman/ua-parser-js/tree/0.7.12](https://github.com/faisalman/ua-parser-js/tree/0.7.12), to define the device type. NOTE: Mobile and non-mobile segments on the results page are not defined using UAParser.

How "Other Tablet" and "Other Mobile" is defined:

- Other tablet- If the device is a tablet but not one of the specified models(iPhone, iPad, etc) on the UI then the device is considered an "other tablet"
- Other mobile -If the device is a tablet but not one of the specified models(iPhone, iPad, etc) on the UI then the device is considered an "other mobile"

Also see Device Type below, which may better fit with how you would like to target.

Optimizely allows you to target your experiment based on device types. You can choose to include or exclude visitors who are using a specific device. Optimizely detects different devices based on the **user agent** provided by the visitor's browser, **so this condition is not evaluated based on device width or screen width**.

For example, if you have a responsive website, you might want to test adding a large image to your landing page. You know that adding an image to the page will not easily scale for smaller screens, so you can use the Device audience
condition to exclude all visitors who are using a mobile phone from your experiment:

This dimension uses the same library as the Device condition above, and was introduced to have a saner set of values to target on than Device. See [https://optimizely.atlassian.net/browse/METRIC-727](https://optimizely.atlassian.net/browse/METRIC-727) for context.

Available on Optimizely X.

Like the Device condition above, this condition allows you to target your experiment based on device type, which is inferred from the user agent provided by the visitor's browser.

Importantly though, it provides a slightly different set of values that you may target:

- Phone
- Tablet
- Desktop / Laptop
- Other (this includes devices like smart TVs, embedded devices, video game consoles, and wearables)

**Dynamic Customer Profiles (DCP)**

Available on [select Optimizely X Web Personalization plans](https://optimizely.com/plans/).

A Dynamic Customer Profile (DCP) is a single, actionable view of your customer built from a combination of Optimizely behavioral data and your first- and third-party customer data. DCP allows you to create audience conditions based on information that you have about your customers, and create 1-1 personalization.

Learn about setting up Dynamic Customer Profiles in [this article](https://optimizely.com/).
You can set audiences based on the IP address of the visitor to your webpage. When you select this feature, you will see a condition box and four matching rules to choose from:

- **Exact Match**: The visitor's IP address must match what is in the box exactly for the targeting condition to be true (so you must enter a full IP address).

- **Prefix Match**: Allows you to define the starting part of the visitor's IP address that must match. Any IP addresses that start with the same numbers as what's in the box should match.

- **Regular Expression**: Advanced type of match that allows you to get very specific in terms of the matching conditions for your visitors. Visit this page for more information about using regular expressions.

- **CIDR Notation**: Advanced type of match that allows you to express an entire subnet more capably than the other match types. If you are not familiar with CIDR Notation and have more advanced technical skills, you can read more about it in this article.

The IP Address audience condition information comes from Akamai, a global content delivery network.

**Language**

Since browsers store their preferred language choice, such as "en-us" (for US English), you can set audience conditions by language preference. If your plan does not include geotargeting, this feature effectively lets you target by country:

However, this assumes that a visitor's language preference will correspond with their location. For example, an US citizen might have his browser set to prefer "en-us" even though he's actually living in Brazil.

**List Attributes**

*Available on select Optimizely X Web Personalization plans.*

List attributes are a type of external attribute that enable you to target visitors that are part of an audience you have already defined somewhere outside of Optimizely. They allow you to import custom lists of users, and then create audience conditions based on those lists. For more information, check out this article on list attributes.

**Location (Geotargeting)**
Location (also known as geotargeting) allows you to only run experiments for visitors coming from specific geographic locations.

To use this feature, start typing the name of a major city, country, state, region, or continent, and a drop-down will appear. Select your location from the drop-down:

You can also add more than one location:

With these targeting settings, the experiment will only run for visitors who come from these specific locations.

The geographic information comes from Akamai, a global content delivery network. This geographic information is derived from the source of the internet connection. This means that under certain conditions, the information may be slightly inaccurate. For example, a visitor who is in a suburb that's close to a city may be identified as coming from the city.

Visitors also using a VPN for another city will be bucketed into an experiment targeting that city. For example, if a Visitor's location is in Boston after VPN, they will be bucketed into an experiment targeting Boston visitors, regardless of their real location.

Mobile visitors are often identified as coming from a location that is different from where they actually are because their internet traffic is routed according to their service provider. For example, a visitor accessing a site on her phone from Livermore, California may appear to come from Sacramento, California, depending on how her service provider routes traffic. You can use the Device audience condition to exclude mobile visitors.

Visitors whose geographic information cannot be determined will not match any locations. This means that if you are targeting specific locations, these visitors will not see the experiment if you are targeting specific locations. However, if you are targeting to exclude locations, these visitors will see the experiment.
Important:

Using the Location audience condition may cause page flashing because the script that obtains the visitor's IP and location information runs asynchronously.

New/Returning Session

The New/Returning Session condition targets experiments towards new or returning visitors to a page where your Optimizely snippet is implemented:

Note:

The New/Returning Session audience condition is "sticky." This condition is session-based, meaning that the first time someone goes to a domain where your Optimizely snippet is implemented, they will be considered "new" for the remainder of the session.

A session is a period of activity for a user. An existing session ends and a new session begins after 30 minutes of inactivity on the site. The maximum session length is 24 hours, at which point a new session automatically begins (even if the user wasn’t inactive for 30 minutes).

For a new visitor, this audience condition is also "sticky" for the duration of that session. In future sessions, those visitors will become "returning" visitors and thus won’t be targeted as "new" visitors.

When interpreting and segmenting results, note that even if a visitor is counted as "new" and later returns as a "returning" visitor, that visitor will count in the "new" segment.

This condition changed as of the release of Audiences. Previously, it was not session-based. Experiments using our old new/returning functionality, before audiences were released, will not migrate automatically to audiences.

Platform/OS

The Platform/OS audience condition allows you to create audiences based on the platform or operating system a visitor is using. Target visitors based on whether they are on mobile or desktop, or if they are using a specific operating systems like Windows or OS X.

For example, if you're running a promotion for a new mobile app that you are launching for iOS and you only want to show that promotion to visitors that who are using iOS, you can use this audience condition to include all visitors who are viewing the specified URL from a device running any version of iOS:
Query Parameters

URLs can contain a lot of information, and one of the most common components is the query string. Query strings contain many pieces of information, and they can be used in Optimizely to personalize your experiments. The most common use case for setting audiences based on query parameters is targeting your experiment for paid ads and search engine marketing (SEM) campaigns.

You also have several options for how to target certain query parameters:

- Choose **equals** from the dropdown if you want to target an exact query parameter value
- Choose **contains** from the dropdown if you want to target any query parameter value that contains the string that you enter (this is similar to **substring match** in URL targeting)
- Choose **matches** from the dropdown if you want to target a value using a **Regular Expression** (note, at this time, Optimizely does not support flags.)
- Choose **has any value** from the dropdown if you want to target based on the **presence of a query parameter**, regardless of what the value is.

Query parameter targeting is case-sensitive for both names and values. If you need your query parameter targeting to be case-insensitive, please contact support or let us know in the [Optiverse Community](https://www.optimizely.com/community/).

Query parameter conditions will not continue to include a visitor in an audience if the visitor returns to the page without the query parameter. In a multi-page experiment, the visitor will fail targeting conditions if they navigate to another page in the experiment without the query parameter that is used in the audience condition.

Let's say you want to test changing a search widget for visitors who are arriving at the page from SEM campaigns. You know that your desired audience will have a query parameter “utm_medium=cpc”.

First, identify the name and value of the query string parameter you care about. Is the value always guaranteed to be the same, or will it change? Are you looking for specific value, or will any value work?

If experiment should run for all visitors with “utm_campaign” present in the URL, but the value does not matter, select Visitor **matches** any of these query parameters, add “utm_campaign” in the left input field as the parameter name, and select **has any value** from the dropdown:
If you do care what the value of the query string parameter is, such as "utm_medium=cpc", follow the same steps, but choose *equals* from the dropdown and enter the value of the query parameter:

Add as many query parameters as needed:

For a more in-depth example of how you can use query parameter audience conditions to deliver more value from your search engine marketing (SEM) and ad campaigns, see our article on [optimizing based on SEM campaigns](#).

The *Referrer URL* audience condition option allows you to trigger an experiment based on the URL that your visitors came from. This is commonly known as the *referrer*.

Typical use cases include setting audiences by the search engine they use, or excluding certain promotional referrers. The match types used for Referrer URL audience conditions are the same as the match types used for *URL targeting*.

For example, suppose you are running several offers on popular discount sites. You want to run an experiment that tests adding promotional banners to your page, but you do not want the visitors coming from a discount site to see these
alternate offers.

First, identify which URL sources should be excluded from the experiment. Then, select Visitor does not come from these URLs and add the specific page URL along with the appropriate match type. In this case, it would be a referral from a substring match to www.groupon.com and www.livingsocial.com:

<table>
<thead>
<tr>
<th>Audience Conditions</th>
<th>Code Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Referrer URL</strong></td>
<td></td>
</tr>
<tr>
<td>Visitor: does not come from these URLs</td>
<td>+</td>
</tr>
<tr>
<td>groupon.com</td>
<td>Substring match</td>
</tr>
<tr>
<td>or</td>
<td>groupon.com</td>
</tr>
<tr>
<td>livingsocial.com</td>
<td>Substring match</td>
</tr>
</tbody>
</table>

This will ensure that the experiment will run for all visitors except for those who had been on a groupon.com or livingsocial.com page immediately prior to arriving at his site.

In another example, suppose you are running an experiment to provide targeted content to visitors arriving from common search engines.

First, identify which search engines should be included in the experiment. Select Visitor comes from these URLs and add the specific page URL along with the appropriate match type. In this case, it would be a referral from a substring match to google.com, yahoo.com, and bing.com:

<table>
<thead>
<tr>
<th>Audience Conditions</th>
<th>Code Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Referrer URL</strong></td>
<td></td>
</tr>
<tr>
<td>Visitor: comes from these URLs</td>
<td>+</td>
</tr>
<tr>
<td>google.com</td>
<td>Substring match</td>
</tr>
<tr>
<td>or</td>
<td>google.com</td>
</tr>
<tr>
<td>or</td>
<td>yahoo.com</td>
</tr>
<tr>
<td>or</td>
<td>bing.com</td>
</tr>
</tbody>
</table>

This ensures that the experiment will run for all visitors who had been on Google, Yahoo, or Bing immediately prior to arriving at his site.

To learn how to use the Referrer URL audience condition to include visitors who came to your site by specific search terms, see our [Referrer URL audience condition article](#).  

**Time/Day of Visit**

The Time/Day of Visit audience condition targets experiments to visitors during certain days and at certain times. Times can be specified to the minute or for the entire day, and they correspond to the visitor's local time zone.
Time/Day of Visit targeting will not continue to include a visitor in an audience if the visitor loads your page outside of the day or time frame. However, the visitor's conversions will count toward whichever variation they were exposed to, even if they convert outside of the day or time frame.

Traffic Source

This condition allows you to create audiences based on the source type in the browser: campaign, search, referral, or direct.

- **Campaign** source type contains users that arrive on a URL containing a 'utm_campaign,' 'utm_source,' 'gclid,' or 'utm_source' query parameter; if the URL contains one of these parameters, the visitor will count as "Campaign" traffic even if they arrived through search

- **Search** source type contains users that visit through a referral URL containing Bing, Google, Yahoo, or Baidu but NOT a specific campaign parameter (otherwise, they would be bucketed as "campaign").

- **Referral** source type includes all users that come from another URL that doesn't count as Campaign or Search

- **Direct** source type includes all users who do not have any external referrer in their URL

These are the same as the source types found in Optimizely's default segments.

The Traffic Source source types (except campaign) are based on the document.referrer value in the browser. Your visitors' source type values may change as they navigate your site.

For example, suppose your site's flow includes a step where visitors leave the page, go to a third-party site or different subdomain, and come back to your site. After they come back to your site, they will count as a referral and be included in your experiment if it is targeted toward referral visitors. Be mindful of using this audience condition if your site includes such a step.