SITECORE®

Experience Rollout

Sitecore Discover

Experience Rollout Specification Document V 4.0.1 20190709

Table of Contents

Chapter 1 **Introduction**

When Sitecore Discover is ready to launch a feature, we ramp our experience to 100% gradually. This gives us time to start our optimizations and monitor our experience's performance on your site. To facilitate this traffic ramp, Discover splits your site into two streams using a random number generator.

- One stream of traffic is presented with the Discover experience (A) while the other includes the customer's default site experience (B).

- The site traffic split will start with 0% (A) and 100% (B) to ensure Discover and your systems are working end-to-end. We also do basic quality checks.

- The traffic split is then set to 1% (A) and 99% (B) to make sure the split in traffic has no biases and the performance metrics are properly presented in the customer's analytics platform.

- Traffic is then increased to 5% (A) and 95% (B). Cookies are shuffled to ensure no bias is present in either site stream.

- Once it is established that the performance metrics are positive and there are no adverse effects resulting from the Discover experience, traffic is increased to the next interval.

- The process of performance monitoring and traffic-setting is continued until we have reached 100% Discover experience.

The traffic can be split by Discover or you may choose to do your own traffic split.

Chapter 2 Discover manages traffic split

If you choose to have Discover manage the traffic split, we configure it using cookies. We use the following process to allow for the traffic ramp:

1. Discover will configure an `__rslct` cookie and set it to denote which feature is active in our experience. The following features may be listed in a comma-separated and sorted list to represent active features for the client:

   o `rw`: recommendation

   o `sb`: preview search

   o `si`: photo search

   o `sp`: search page

   o `sv`: voice search

   o `cb`: content block

   For example, the cookie `__rslct=rw,si,sp` represents that the recommendation, photo search, and search page features are active for the client.

2. As we ramp from 0% to the next traffic interval, the cookie value will be set to include the active feature that is being ramped up.

3. Note that multiple features may be ramped up independently. For example, if `__rslct=rw,sb` then it is possible that recommendation is at 75% Discover experience, and preview search is at 10%.

4. The features listed in the cookie will always be in alphabetical order.

5. You can determine if a feature is on for a given user by using a simple JavaScript function like `s.indexOf('sp')>=0` where `s` is the value of cookie `__rslct`.

Chapter 3 Customer manages traffic split

If you decide to manage your own traffic split, and you are using Discover widgets, then the Discover experience can be activated using the `__rsu` cookie. To split your traffic, configure an `__rsu` cookie and set it as either `1` or `0` to indicate whether the Discover experience is active or not.

Discover reads the `__rsu` cookie, and if `__rsu=1`, then the user is presented with the Discover experience as configured.

Chapter 4 Performance Metrics

Before the start of the rollout, it is important to agree on

1. Performance metrics. These could be conversion rate, revenue per visitor, etc.

2. Analytics system. This could be Google Analytics, Omniture, etc.

If you are rendering the UI experience using Discover APIs, then you are responsible for sending events to the analytics system for both the Discover and non-Discover experiences for views, clicks, add-to-cart, and orders. Include the flag `__rsu=1` or `0` to indicate if the event is part of the Discover experience or not.

If Discover is rendering the UI experience, then we will send the events to the analytics system. We will include a flag `__rsu=1` or `0` to indicate if the event was part of the Discover experience or not.