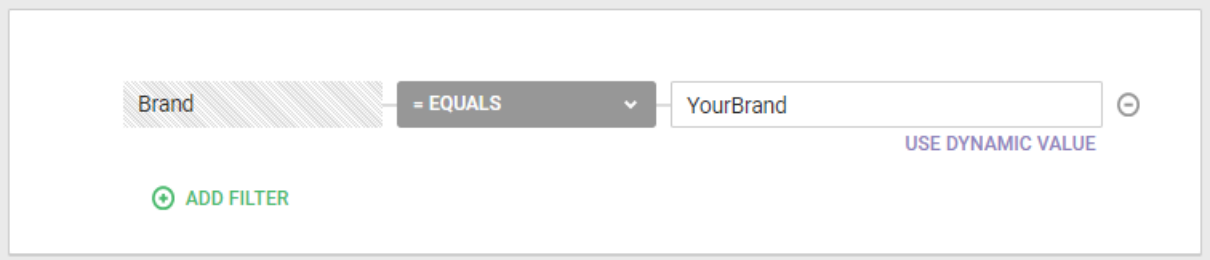
Filters in Recommendations

Attribute-based filters that you create on the **Global Settings** tab of the Product Recommendations page as well as the filters that you build when creating a recommendation strategy or a bundle for Monetate Dynamic Bundles allow you to further refine the products recommended. Each filter is composed of an attribute, an operator (for example, **= equals**, **≠ does not equal**), and a value, which can be static or dynamic.



On the **Global Settings** tab as well as in recommendation strategies, you can build multiple filters and join them with AND/OR logic. Currently, you can only use AND logic when you build multiple filters in a Dynamic Bundle product category or for the bundle itself.

Attribute Options

The options in **ADD FILTER** when you're creating a global filter are the specific attributes that appear in the account's default product catalog, including custom attributes, as well as the optional attributes of Monetate's product catalog specifications.

When you're creating a filter for a recommendation strategy, the options in **ADD FILTER** are those attributes found in the selected product catalog, including custom attributes, along with the optional attributes of Monetate's product catalog specifications. Therefore, ensure that any attribute you use for a filter appears in the product catalog selected.

The **ADD FILTER** options for both product category filters and bundle-level filters are the attributes found in the selected product catalog, including custom attributes, along with the optional attributes of Monetate's product catalog specifications. If you're creating a local bundle that uses a Recommendations dataset, the `lookup_key` and `rank` attributes from the dataset *are not* available to use in a filter.

Product Type Filter

A filter that uses the **Product Type** option is unique because it treats the product type filtering criteria as a comma-separated list. The reason is that the values submitted in the `product_type` attribute in a product catalog can be a comma-separated string and thus contain multiple values for a filter to find matches. Therefore, the Product Type filter is considered to match if *any* of the product types in the comma-separated filter list match.

The **= equals (Starts With)** operator matches any product type with any item in the filter list that starts with the filtering criterion. For example, using this operator with the filtering criterion `ABC` would match with `product_type` attribute values `ABC` as well as `ABC123`. If both of those values appeared in a comma-

separated string as the attribute value (for example, `ABC,ABC123,DGF`), the filter would still match because each individual comma-separated entity is a potential for matching, and both `ABC` and `ABC123` appear in the string.

The **= equals (True Equality)** operator strictly matches on the filtering criterion you enter. For example, it would match `ABC` only with the `product_type` attribute value `ABC`. It would still match if the value `ABC` appeared in the middle of a comma-separated string, such as `123,CGE,ABC,897` because within that comma-separated string `ABC` is recognized as an individual value for matching purposes.

The **contains** operator, naturally, matches on any `product_type` attribute value that contains the filtering criterion.

Finally, the **≠ does not equal** operator sets the matching criterion to "does not start with."

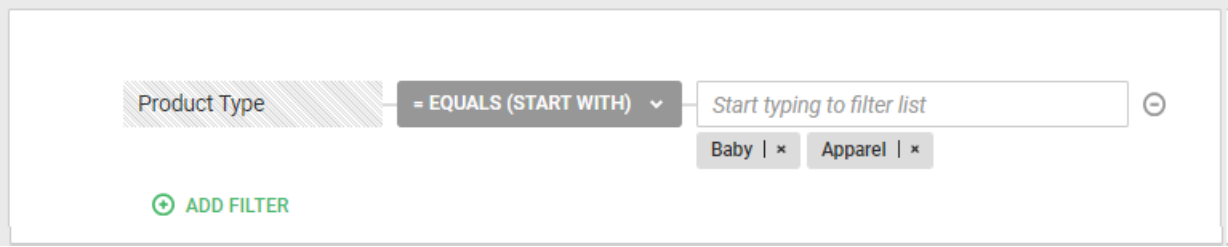
This logic works the same if the filter uses a static value or a dynamic value.

The matching logic is case-sensitive. It also checks for internal punctuation within a word or multi-word value (for example, `t-shirts` or `lace-ups`) and for any spaces within a multi-word string (for example, `soft goods` or `maxi dresses`).

If you add a value to the filtering criteria that contains multiple words that *are not* separated by commas and select either the **= equals (Starts With)** operator or the **≠ does not equal** operator, then the match is based on the full string, not just the first word. For example, if `men's shirts` is the filtering criteria, then any match must contain *men's shirts*.

Consider a Product Type filter using the **= equals (Starts With)** operator and `Automotive` as the filtering criteria. It would match on `Automotive Parts` as well as on `Baby, Automotive` because the comma after `Baby` signals that the `Automotive` that follows is a product type that matches the filter criterion. However, `Auto Parts` wouldn't match because it doesn't contain the full `Automotive` criterion.

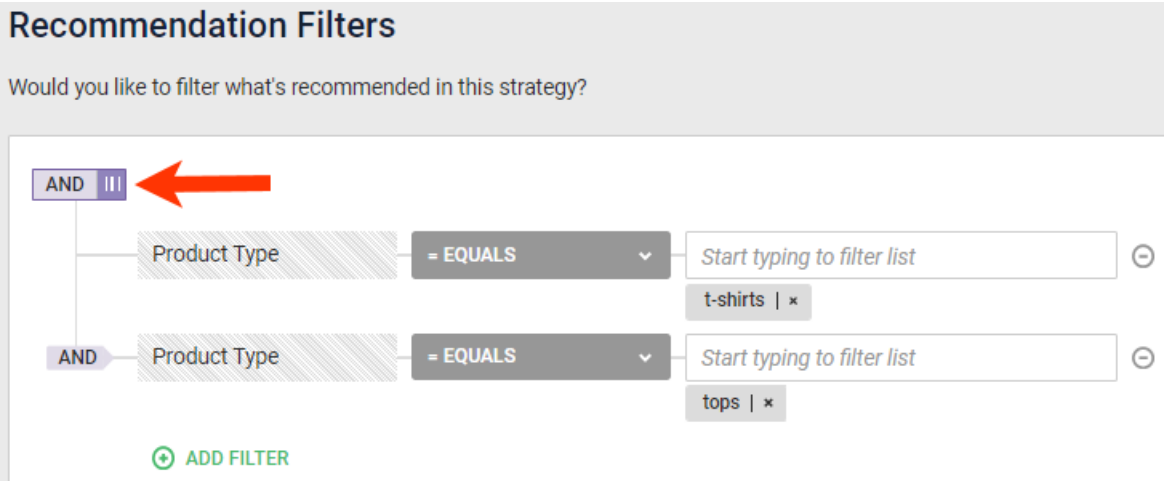
You can include multiple filtering criteria in the filter list, and a match is made if any value in the `product_type` attributes in a product catalog match any of the filtering criteria. For example, if `Baby` and `Apparel` appear in the filter list, then the **= equals (Starts With)** operator in the filter equation looks for product types that start with either term.



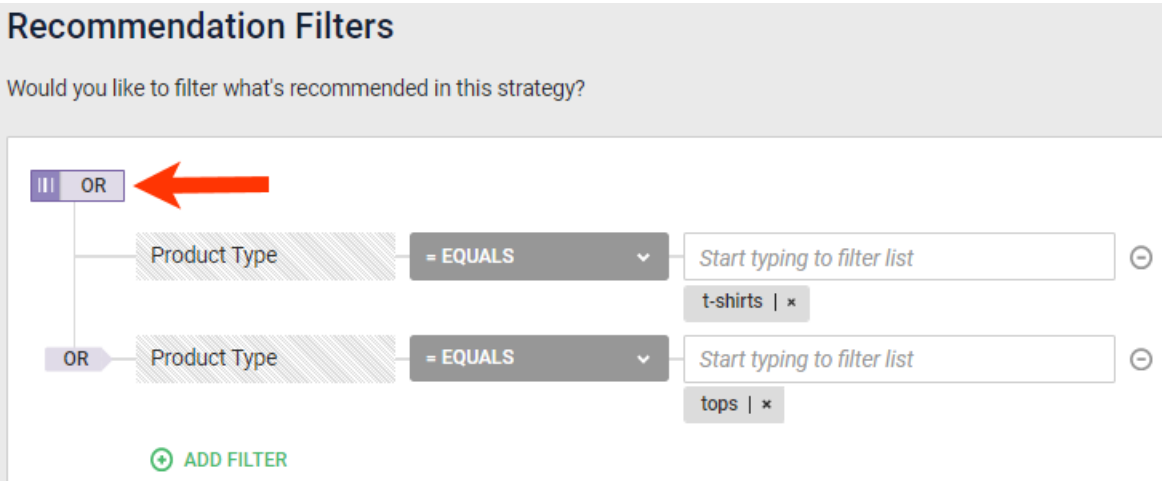
Matches for this filter would thus include *Baby*, *Baby things*, and *Clothing, Apparel* because the comma after *Clothing* indicates that the *Apparel* that follows is its own product type. *Apparel, Women's, Shirts* would also match because the string begins with *Apparel*. However, *Clothing, Men's* wouldn't match this filter because neither of the two values, `Clothing` and `Men's`, matches any product type in the filter list.

For a recommendation strategy, you can use multiple Product Type filters to build a more robust filter since you

connect the filters by setting the operator toggle to **AND** or to **OR**.



If you join multiple filters that use the **= equals (Starts With)** operator with the toggle set to **AND**, then each match must start with *all* the filter list criteria. Therefore, the filter string shown in the screenshot above would match with *apparel, tops, t-shirts* as well as *tops, t-shirts, women's t-shirts* and *tops, t-shirts, men's t-shirts*. However, the filter wouldn't match with *men's t-shirts* or *women's t-shirts* because neither product type includes the `tops` criterion and neither one *starts with* `t-shirts`.



If you join these same multiple filters with the toggle set to **OR**, then the effect is essentially the same as if you had put all the filter list terms in one recommendation filter. Therefore, the filter string shown in the screenshot above works the same as having a single filter with `t-shirts` and `tops` in the filter list.

Static and Dynamic Values

The value in a filter equation can be static or dynamic. Static values allow you to customize recommendations based on defined attribute-based rules. Dynamic values can help ensure recommended products have similar attributes to what interests a customer.

Dynamic Value Options

The options available after you click **USE DYNAMIC VALUE** are determined first by the attribute you selected from **ADD FILTER** and then by the type of recommendation algorithm—collaborative or noncollaborative—that you selected in the basic strategy configuration.

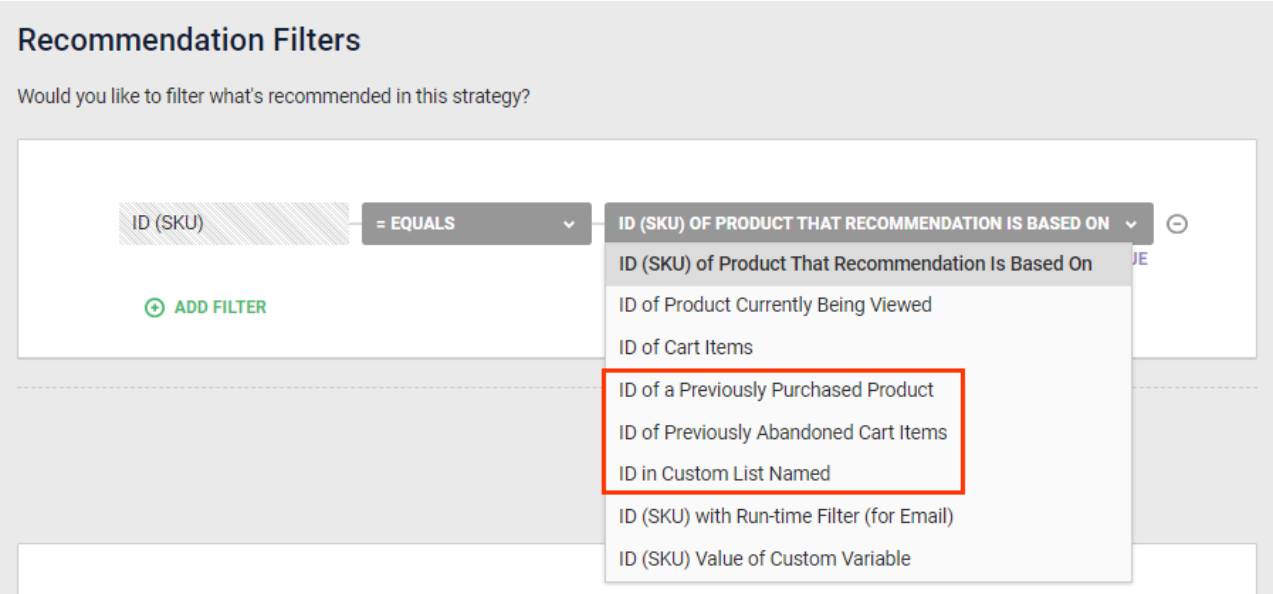
These options appear for most attributes that work with a dynamic value:

- <attribute> of Product Currently Being Viewed
- <attribute> of Cart Items
- <attribute> with Run-Time Filter (for Email)
- <attribute> Value of Custom Variable

If you selected a collaborative recommendation algorithm, then **<attribute> of Product That Recommendation Is Based On** appears with the dynamic value options listed above.

The **ID (SKU)** and **Item Group ID (Product ID)** attribute options have three additional dynamic value options:

- <attribute> of a Previously Purchased Product
- <attribute> of Previously Abandoned Cart Items
- <attribute> in Custom List Named



The **ID in Custom List Named** option allows you to filter by the SKUs or product IDs in a custom list dataset that you select. See Upload a Custom List Dataset for more information.

The **Price** and **Product Price** attributes have only these three dynamic value options:

- <attribute> of Product on Page
- Minimum <attribute> of Cart Items
- Maximum <attribute> of Cart Items

The **Sale Price** attribute has these same three dynamic value options along with a fourth: **Price**.

Run-Time Filter

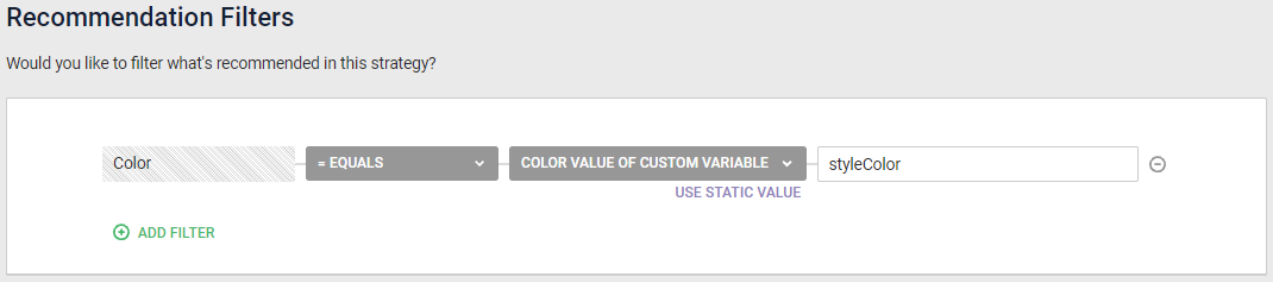
The **with Run-Time Filter (for Email)** dynamic value option makes it possible to create a Product Recommendations for Email experience that filters the products recommended by a specific value passed at run time using a pass-through parameter in the email's HTML code. See Run-Time Context for Recommendations Email Experiences for more information.

Value of Custom Variable Filter

The **Value of Custom Variable** dynamic filter option requires you to enter a custom variable that your site passes to Monetate either using the `setCustomVariables` method call in the Monetate API implementation or using `monetate:context:CustomVariables` in the Engine API implementation.

> You can only use the **= equals** or **≠ does not equal** operator option with the **Value of Custom Variable** dynamic filter option.

The custom variable's value at the time of the recommendations action request is evaluated against the values that appear in the product catalog associated with the recommendation strategy in the field for the attribute that you chose from **SELECT ATTRIBUTE**. Any product that doesn't have the custom variable value in the specified attribute field is excluded from the results.



If the value of the `styleColor` custom variable at run time is `purple`, then the filter as configured in the screenshot excludes from consideration any products in the catalog that don't have `purple` in the `<color>` field.

Tips for Applying Filters

Filters help ensure the recommended products don't include products that are unexpected or outside your brand guidelines. For example, on a Women's Shirts product listing page, you may want to apply a **Gender = women** filter to ensure that only other women's products appear.

Consider too using a broad dynamic filter as a way to better ensure no unexpected products display. For example, you may want to only include products targeted to the same gender as the gender targeted by the products on the page.

Another means of improving the relevance of recommended products is to use collaborative recommendation algorithms, such as Viewed and Also Viewed, Viewed and Later Purchased, and Purchased and Also Purchased. Collaborative recommendation types allow you to leverage the "wisdom of the crowd" to show more relevant product recommendations. By applying relevant broad filters, you can best leverage the wisdom of the crowd.

Keep in mind, though, that the more filters you use, the more you limit the algorithm's potential output. For example, you build a strategy with the Viewed and Also Viewed recommendation type and apply a **Product Type = suits** filter. When a customer views a suit, this configuration means other suits are likely recommended, but the specificity of the static value of the filter may also reduce the likelihood that other products that customers view along with suits, such as dress shoes or ties, are recommended. To avoid this situation, you may want to use a broader filter, such as the dynamic filter **Gender = Gender of Product Currently Being Viewed**, to only show other products whose gender matches the gender of the suit a customer views.

If you find that broad filters are still returning products you don't want recommended, you can always add filters and evaluate how they affect the recommendations.

No matter if they're set on the **Global Settings** tab of the Product Recommendations page, at the recommendation strategy level, at the bundle level, or at the product category level in a bundle, filters that have static values are all applied simultaneously. So too is a filter with the **<attribute> of Product That Recommendation Is Based On** dynamic value option, which only is available if you've selected a collaborative recommendation algorithm.

The following filter best practices apply when you're creating a filter in a recommendation strategy or bundle:

- If you want to filter based on viewed products or carted products to cover more of the products in the catalog, then select a collaborative recommendation algorithm along with one of the options from the VIEWED or CARTED categories of **Base Recommendation on**, and then use the **<attribute> of Product That Recommendation Is Based On** dynamic value option in a filter.
- When working with a collaborative recommendation algorithm, don't select a dynamic value that's similar to the option you selected for **Base Recommendation on**.

Contact your dedicated Customer Success Manager (CSM) if you need assistance creating filters that meet your goals.