

# Dynamic Category Expressions

The dynamic category expression controls the product membership of the dynamic category. You can find the expression of a dynamic category in the **Expression** section on a dynamic category page.

You can create and edit the dynamic category expression either using the visual expression builder or the advanced expression editor. If you choose to use the visual expression builder, you'll need to create containers and filters using dialog boxes in Admin. If you choose to use the advanced expression editor to create and edit the expression you'll need to write either a JSON styled tree, or write a plain text string.

Refer to following sections for more information about either manually writing the dynamic category expressions, or using the visual expression builder.

## Visual Expression Builder

The visual expression builder allows you to visually create groups and conditions that specify the details of your expression. A group specifies whether any or all of the conditions within it need to evaluate as True in order for products to be added to the dynamic category. A condition contains an expression field, an operator, and a value, for example *Product code is equal to BLUE-SHIRT*.

### Add a Group

To add a group:

1. Under the **Expression** section, expand the actions menu and select **Add Group**.
2. Specify whether **Any** or **All** of the conditions in the container must be True by clicking **Edit**. **Any** acts like an or, and **All** acts like an and. For example, you have a condition that includes the categories Prada and Women's. If you specify **Any**, any products that belong to the Prada or the Women's category will be included in the dynamic category. However, if you specify **All**, only products that belong to both the Prada and Women's categories will be included in the dynamic category.

For example, you add a group that includes four conditions and allow any of the conditions to be True.

### Add a Condition

To add a condition:

1. Under the **Expression** section, expand the actions for the applicable group and select **Add Condition**.
2. Select a condition **Field**.
3. Select an **Operator**.
4. Specify a **Value**.

For example, you want the dynamic category to include any product with a list price of \$50. To accomplish this, you create a condition that states *List price is equal to \$50*. Some fields allow for null, or no values. The asterisk on the **Value**

drop-down menu determines whether the selected field requires a value. For example, you can create a condition that states *Sale price has no value*.

## Add an Existing Condition to a Group

To add an already existing condition to a group, click and drag the condition into the desired group.

## Advanced Expression Editor

The advanced expression editor allows you to manually write either JSON equivalent text or plain text to create the expression.

To create or edit an expression manually:

1. In the **Expression** section, click **Advanced**.
2. In the Advanced Edit Expression window, select either the **JSON** view or the **Text** view to write your expression.

## Preview the Expression Results

Before saving the dynamic category, you can preview the results of the expression. This gives you the opportunity to view the products that will be included in the dynamic category before saving the category. When previewing the results of the expression, you can specify the site and either the Live or Staged version of the site.

If your product publishing settings are set to **Live** for a catalog, you can only preview the live matching results. To view both the live and staged matching results, you need to set your product publishing settings to **Staged**. Refer to [Publishing Settings](#) for more information about setting your product publishing settings.

Only products that appear on your storefront are returned in the preview expression results. If, for whatever reason, including out of stock, a product does not appear on your storefront either currently or on the staged preview date then it will not appear in either the live or staged preview expression results. [Product ranking rules](#) do not affect the preview expression results.

To preview the results of an expression:

1. Under the **Expression** section, click **Preview**.
2. In the **Preview Expression** modal, the results of the expression are shown in the **Preview** grid on the right.
3. Select the **Site** you wish to preview products for as well as the site's **State**. **Tip:** If you select **Staged** as the site's state, you can specify the date you wish to preview. This allows you to preview the products that will be included in the dynamic category on the specified date.
4. If the product results do not match your desired results, you can edit the expression using the **Expression** frame on the left.
5. If you edit the expression in the **Preview Expression** modal and you wish to save your changes, click **Done**. Otherwise, click **Cancel** to disregard any changes and close the **Preview Expression** modal.

The total matched results are shown in the bottom-right corner of the **Preview** grid. You can use this total to quickly determine whether your expression meets your desired criteria.

## Expression Fields

When you write a dynamic category expression, you define values for the supported fields in order to build out the expression.

Refer to the following table for the supported expression fields and their operators:

### Precomputed and Realtime Categories

Field	Supported Operators	Data Type	Allow Null Values?
<p><b>Visual Expression Editor</b> Product code</p> <p><b>Manual Editor</b> ProductCode</p>	<p>EQ (is equal to) NE (does not equal) IN (is one of the following)</p>	String	No
<p><b>Visual Expression Editor</b> Category code</p> <p><b>Manual Editor</b> Categories.CategoryCode (Precomputed categories can reference other precomputed categories; however, Kibo eCommerce recommends limiting the amount of precomputed categories the parent precomputed category references.)</p>	<p>EQ (is equal to) REQ (is equal to and includes child categories) NE (does not equal) IN (is one of the following, does not recursively include selected child categories).</p>	String	No
<p><b>Visual Expression Editor</b> Product Type</p> <p><b>Manual Editor</b> ProductTypeId</p>	<p>EQ (is equal to) NE (does not equal) IN (is one of the following)</p>	Integer	No

Field	Supported Operators	Data Type	Allow Null Values?
<p><b>Visual Expression Editor</b> List price</p> <p><b>Manual Editor</b> Price.CatalogListPrice</p>	<p>EQ (is equal to) NE (does not equal) LT (is less than) LE (is less than or equal to) GT (is greater than) GE (is greater than or equal to) IN (is one of the following)</p>	<p>Decimal</p>	<p>No</p>
<p><b>Visual Expression Editor</b> Sale price</p> <p><b>Manual Editor</b> Price.CatalogSalePrice (This field refers to the sale price of a product entered on the product page in Admin.)</p>	<p>EQ (is equal to) NE (does not equal) LT (is less than) LE (is less than or equal to) GT (is greater than) GE (is greater than or equal to) IN (is one of the following)</p>	<p>Decimal</p>	<p>Yes</p>

Field	Supported Operators	Data Type	Allow Null Values?
<p><b>Visual Expression Editor</b> Days available in catalog</p> <p><b>Manual Editor</b> DaysAvailableInCatalog</p> <p>This field is calculated using the <b>First Available Date</b> field in the product properties.</p>	<p>EQ (is equal to) NE (does not equal) LT (is less than) LE (is less than or equal to) GT (is greater than) GE (is greater than or equal to) IN (is one of the following)</p>	Integer	No
<p><b>Visual Expression Editor</b> Fulfillment types supported</p> <p><b>Manual Editor</b> FulfillmentTypes Supported</p>	<p>EQ (is equal to) NE (does not equal) IN (is one of the following)</p>	String Valid values: "DirectShip", "InStorePickup", "Digital"	No
<p><b>Visual Expression Editor</b> Weight</p> <p><b>Manual Editor</b> Measurements.Package Weight.Value</p>	<p>EQ (is equal to) NE (does not equal) LT (is less than) LE (is less than or equal to) GT (is greater than) GE (is greater than or equal to) IN (is one of the following)</p>	Decimal	Yes

Field	Supported Operators	Data Type	Allow Null Values?
<p><b>Visual Expression Editor</b> Height</p> <p><b>Manual Editor</b> Measurements.Package Height.Value</p>	<p>EQ (is equal to) NE (does not equal) LT (is less than) LE (is less than or equal to) GT (is greater than) GE (is greater than or equal to) IN (is one of the following)</p>	<p>Decimal</p>	<p>Yes</p>
<p><b>Visual Expression Editor</b> Width</p> <p><b>Manual Editor</b> Measurements.Package Width.Value</p>	<p>EQ (is equal to) NE (does not equal) LT (is less than) LE (is less than or equal to) GT (is greater than) GE (is greater than or equal to) IN (is one of the following)</p>	<p>Decimal</p>	<p>Yes</p>

Field	Supported Operators	Data Type	Allow Null Values?
<p><b>Visual Expression Editor</b> Length</p> <p><b>Manual Editor</b> Measurements.Package Length.Value</p>	<p>EQ (is equal to) NE (does not equal) LT (is less than) LE (is less than or equal to) GT (is greater than) GE (is greater than or equal to) IN (is one of the following)</p>	Decimal	Yes
<p><b>Visual Expression Editor</b> Attribute Property</p> <p><b>Manual Editor</b> properties.</p> <p>This field is based on the available product properties and uses the fully-qualified name (FQN) of the property.</p>	The supported operators of this field are dependent on the selected property.	The value of this field is dependent on the selected property's value.	Yes

### Realtime Categories Only

Field	Supported Operators	Data Type	Allow Null Values?
-------	---------------------	-----------	--------------------

Field	Supported Operators	Data Type	Allow Null Values?
<p><b>Visual Expression Editor</b> Post-Discout Price</p> <p><b>Manual Editor</b> Price.SalePrice (this field refers to the post discount price of a product.)</p>	<p>EQ (is equal to)</p> <p>NE (does not equal)</p> <p>LT (is less than)</p> <p>LE (is less than or equal to)</p> <p>GT (is greater than)</p> <p>GE (is greater than or equal to)</p> <p>IN (is one of the following)</p>	Decimal	Yes
<p><b>Visual Expression Editor</b> Sale Type</p> <p><b>Manual Editor</b> Price.SaleType</p>	<p>EQ (is equal to)</p> <p>NE (does not equal)</p> <p>IN (is one of the following)</p>	<p>String</p> <p>Valid values: Catalog Sale Price Discounted List Price Discounted Catalog Sale Price</p> <p>Refer to <a href="#">Sale Type Values</a> for more information about these values.</p>	Yes

### Sale Type Values

The realtime sale type field has three different values that each apply to a product's various sale types: Catalog Sale Price, Discounted List Price, and Discounted Catalog Sale Price. Refer to the following table for more information about each of these values and their relationship to a product's various sale prices.



Value	Description
<b>Visual Expression Editor</b> Catalog Sale Price  <b>Manual Editor</b> CatalogSalePrice	The price specified in a product's <b>Sale Price</b> field.
<b>Visual Expression Editor</b> Discounted List Price  <b>Manual Editor</b> DiscountedList	The product's list price after applicable discounts are applied.
<b>Visual Expression Editor</b> Discounted Catalog Sale Price  <b>Manual Editor</b> DiscountedCatalogSalePrice	The price specified in a product's <b>Sale Price</b> field after applicable discounts are applied.

## Expression Examples

These four examples provide visual builder and advanced editor samples for different expressions.

### Example 1

The following example shows a precomputed dynamic category expression that includes all products that have been in the catalog for 30 days or less:

**Visual Expression Builder**

**Advanced Editor: Tree View**

```
{
  "type": "container",
  "logicalOperator": "or",
  "nodes": [
    {
      "type": "predicate",
      "left": "daysavailableincatalog",
      "right": 30,
      "operator": "le"
    }
  ]
}
```

**Advanced Editor: Text View**

daysavailableincatalog le 30

## Example 2

The following example shows a precomputed dynamic category expression that includes all products with the property color set to red:

**Visual Expression Builder**

**Advanced Editor: Tree View**

```
{
  "type": "container",
  "logicalOperator": "or",
  "nodes": [
    {
      "type": "predicate",
      "left": "properties.tenant~color",
      "operator": "eq",
      "right": "Red"
    }
  ]
}
```

**Advanced Editor: Text View**

properties.color eq "Red"

## Example 3

The following example shows a precomputed dynamic category expression that includes all products with the property color set to red and are in the apparel category.

## Visual Expression Builder

### Advanced Editor: Tree View

```
{
  "type": "container",
  "logicalOperator": "and",
  "nodes": [
    {
      "type": "predicate",
      "left": "Categories.CategoryCode",
      "operator": "eq",
      "right": "apparel"
    },
    {
      "type": "predicate",
      "left": "properties.color",
      "operator": "eq",
      "right": "Red"
    }
  ]
}
```

### Advanced Editor: Text View

Categories.CategoryCode eq "apparel" and properties.color eq "red"

## Example 4

The following example shows a realtime dynamic category expression that includes all products with a discounted sale price and are in the apparel category. You must set the **Product Membership** drop-down menu to **Realtime** in order to access the dynamic realtime fields.

## Visual Expression Builder

### Advanced Editor: Tree View

```
{
  "type": "container",
  "logicalOperator": "and",
  "nodes": [
    {
      "type": "container",
      "logicalOperator": "and",
      "nodes": [
        {
          "type": "predicate",
          "left": "Price.SaleType",
          "operator": "eq",
          "right": "DiscountedList"
        },
        {
          "type": "predicate",
          "left": "Categories.CategoryCode",
          "operator": "eq",
          "right": "apparel"
        }
      ]
    }
  ]
}
```

**Advanced Editor: Text View**

(Price.SaleType eq "DiscountedList" and Categories.CategoryCode eq "apparel")