



PRODUCT MODELING GUIDE

SITECORE E-COMMERCE DONE RIGHT. ➔ ACTIVECOMMERCE.COM





Active Commerce Product Modeling Guide

[Terminology](#)

[Product Types](#)

[The Product Base Template](#)

[Creating a Simple Product Template](#)

[Using Lookup Values](#)

[Creating Product Family Templates](#)

[Using Color Swatches](#)

[Configuring Product Specs \(Specification Fields\)](#)

[Tips](#)

[The Product Repository](#)

[Adding a Repository](#)

[Assigning a Repository to a Site](#)

[Adding Repository Categories](#)

[The Product Catalog](#)

[Catalog Categories](#)

[The “Product Page” Tab](#)

[Configuring Product Compare](#)

[Configuring Product Filters](#)

Terminology

Product Type – A Sitecore template that defines a specific product.

Product Field – Fields defined when setting up product models under “Templates/Product Types.” These will be the lowest child-level items, e.g., “Product Types/Water Bottle/Water Bottle Data/Material.”

Product Option Field – Same as “Product Field,” but field is also marked as “Is Product Option.” Used when defining Product Variants.

Product Repository (Repository) – Refers to a *back-end* product root item located under Active Commerce/Product Repositories.

Repository Category – Refers to categories under the Product Repository.



Product Catalog (Catalog) – Refers to the *front-end* product root item (i.e., “Products”) within the content structure of the site (e.g., sitecore/Content/MySite/Home/Products).

Catalog Category – Refers to categories visible on the site — under the Product Catalog (e.g., sitecore/Content/MySite/Home/Products/Electronics).

Product Types

Active Commerce uses Sitecore templates to define product characteristics. Therefore, modeling your products should be very familiar to a Sitecore developer; simply build up like you would any other template — using the Template Builder tab, Inheritance, Standard Values, etc.

All product types should be defined under “sitecore/Templates/Product Types.”

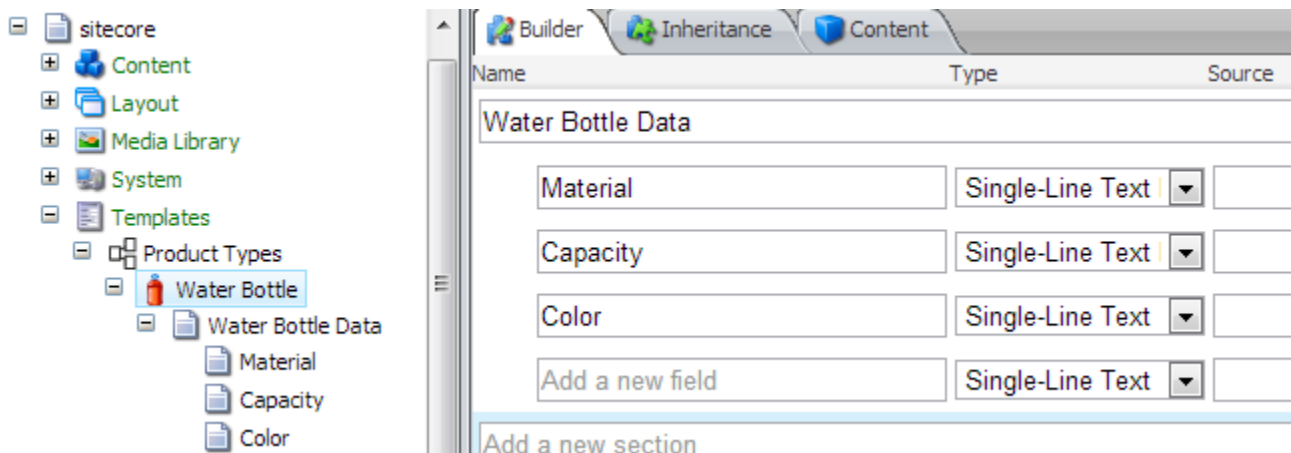
The Product Base Template

All products in Active Commerce must be based off of the product base template: */sitecore/templates/ActiveCommerce/Product/Product*

This template provides base product fields and settings that Active Commerce will expect.

Creating a Simple Product Template

1. In the content tree, go to “sitecore/Templates/Product Types”
2. Insert item > New Template
 - NOTE: You can also add Template Folders here. Organize how you see fit.*
 - a. Enter the name of your product template (e.g., Water Bottle).
 - b. Set the Base template to “Templates/Active Commerce/Product/**Product**.”
 - c. Click Next, Next, Finish.
3. Add product-specific fields using the template Builder tab (e.g., Material, Capacity, Color).
4. Configure a template icon (optional).
5. When complete, it should look something like this:





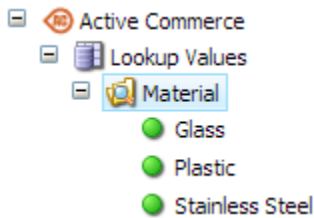
Tip: To make your product type’s template fields appear after the built-in Active Commerce fields in the Content Editor, edit the “Sortorder” field on the template section (here, Water Bottle Data) and set its value to 900 or greater.

Using Lookup Values

Product fields that have a finite set of possible values should not use Single-Line Text. Instead, they should use a Droplink or a Droptree and corresponding Lookup Values. These are configured under “sitecore/Active Commerce/Lookup Values.” Lookup Values are organized into Lookup Folders.

Adding a Lookup Folder and Values

1. In the content tree, go to “sitecore/Active Commerce/Lookup Values.”
2. Insert Item > Lookup Folder.
3. Enter a name for the Lookup Folder (e.g., Material) and click OK.
4. Select the new Lookup Folder > Insert Item > Lookup Value.
5. Enter the value (e.g. Stainless Steel) and click OK.
6. Repeat steps 4-5 until all values have been entered.



7. Modify product template fields to use the Lookup Folder. On the template Builder tab:
 - a. Set Type to Droplink or Droptree.
 - b. Set Source to the Lookup Folder we just created.

Let’s switch over the Material and Capacity fields we defined earlier to use Lookup Values instead.

NOTE: Leave color for now. This will be modified later on to use Color Swatches.

1. Add a Capacity Lookup Folder and corresponding values (e.g., 12 oz, 20 oz). (See steps above.)
2. In the content tree, select the Water Bottle product template.
3. On the template Builder tab:
 - a. Set Material field Type to Droplink and Source to: “/sitecore/commerce/lookup/Material”
 - b. Set Capacity field Type to Droplink and Source to: “/sitecore/commerce/lookup/Capacity”



| Name | Type | Source |
|-------------------|------------------|------------------------------------|
| Water Bottle Data | | |
| Material | Droplink | /sitecore/commerce/lookup/Material |
| Capacity | Droplink | /sitecore/commerce/lookup/Capacity |
| Color | Single-Line Text | |
| Add a new field | Single-Line Text | |

Creating Product Family Templates

Your webshop may need to sell different variations of the same product, for example, a shoe can come in different colors and sizes. In this case, it makes sense to create two templates — one which holds the common product characteristics (*Variable Product*) and one which holds the variable data (*Product Variant*). The product variants will be added as children of the variable product in the Sitecore tree. This structure makes up a product family.

First, create the variable product

1. In the content tree, go to “sitecore/Templates/Product Types.”
2. Insert item > New Template.
 - a. *NOTE: You can also add Template Folders here. Organize how you see fit.*
 - b. Enter the name of your product template (e.g., Shoes).
 - c. Set the Base template to “Templates/Active Commerce/Product/**Variable Product Base**”
 - d. Click Next, Next, Finish.
3. Add common product-specific fields using the template Builder tab (e.g., Brand, Gender, Width, Insole). Use lookup values where appropriate.
4. Configure a template icon (optional).

Second, create the product variant

1. In the content tree, go to “sitecore/Templates/Product Types.”
2. Insert item > New Template.
 - a. Enter the name of your product template (e.g., Shoes Variant).
 - b. Set the Base template to “Templates/Active Commerce/Product/**Product Variant Base**.”
 - c. Click Next, Next, Finish.
3. Add variant data fields using the template Builder tab (e.g., Size, Color). Use lookup values where appropriate.



4. Identify the Product Option Fields (see def). In most cases this will be all of the fields you added in the previous step. These are what drive the Active Commerce product selection UI (product filtering, drop downs on Product Detail page, details in cart, etc.)
 - a. In the content tree, select the product field (e.g., Size).
 - b. In the “Active Commerce” section, check the “Is Product Option” checkbox.
 - c. Repeat for each variant field.
5. Configure a template icon (optional).

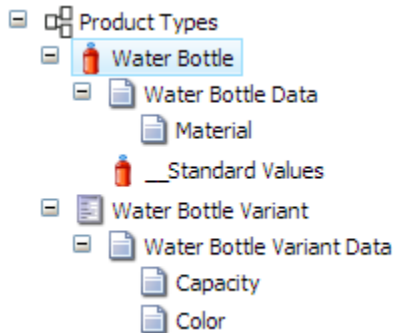
Finally, define insert options

We want to make it clear to product content authors that this is a product family and product variants need to be added.

1. In the content tree, select the variable product (e.g., Shoes).
2. Add Standard Values (Builder tab in the Ribbon > Template section > click Standard Values).
3. Select the __Standard Values item that was added.
4. Configure tab in the Ribbon > Insert Options section > click Assign.
5. Move the product variant template (e.g., Shoes Variant) to the right pane, and click OK.

Let’s convert our Water Bottle to a Product Family, with Capacity and Color fields being our variable data.

1. Switch the Base template of the Water Bottle template, from “Product” to “Variable Product Base.”
2. Add a product variant (see steps above), named “Water Bottle Variant”
 - a. Move the “Capacity” and “Color” fields from the Water Bottle template to this new template. Either delete and re-add, or use Copying > Move To (*NOTE: You’ll have to create a field section first*).
 - b. Mark the “Capacity” and “Color” fields as Product Option fields.
3. Add Standard Values and define “Water Bottle Variant” as insert option for the “Water Bottle” template.
4. When complete, you should have something like this:



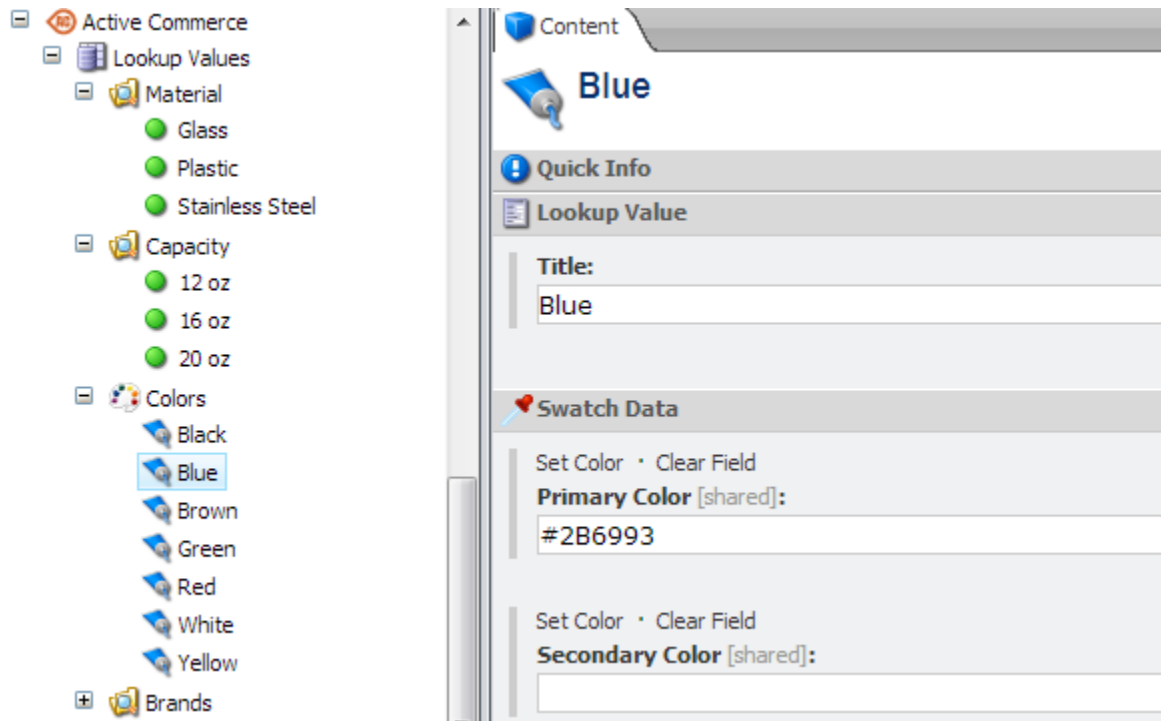
Using Color Swatches



Color swatches make use of a couple techniques we've already covered. Color/swatch values are stored in a special version of a Lookup Folder called a Color Folder, and color fields are simply Product Option Fields with one minor difference — they are named “Color.” This combination enables swatch browsing and selection capabilities throughout Active Commerce (Product List and Product Detail pages).

First, create color library(s)

1. In the content tree, go to “sitecore/Active Commerce/Lookup Values.”
2. Insert Item > Color Folder.
3. Select color folder > Insert Item > Color Value.
4. Enter a color Title. This displays in the product option selection box, and also on the swatch tooltip/hover.
5. Define swatch colors — Primary is required, Secondary is optional. Enter a hex value directly, or use the “Set Color” option to use a color picker.



Then, add color product field(s)

1. Add a product option field (see def) to your desired product — name it “Color” (*IMPORTANT: The field MUST be named “Color”*).
2. For the Type, select Droplink.
3. For the Source, enter the path to the desired color library (e.g., “/sitecore/commerce/lookup/Colors”).

Let’s switch over the Color field we defined earlier to use Color Values instead.



Since we already named our field “Color,” this is all we have to do to get swatch functionality working.

1. Add a Color folder and corresponding values (e.g., Black, Blue), named “Colors” (see steps above).
2. In the content tree, select the Water Bottle Variant product template.
3. On the template Builder tab:
 - a. Set Color field Type to Droplink and Source to: “/sitecore/commerce/lookup/Colors”

Configuring Product Specs (Specification Fields)

1. In the content tree, select the product field (see def) you’d like to set up as a specification.
2. In the “Active Commerce” section, check the “Is Product Specification” checkbox.
3. The field will now show up on the “Specs” tab of the Product Detail page.

Tips

1. Take advantage of template inheritance. Think about the fields you’ll need, and create templates to support them. For example, you may have certain fields that you need on all products in your repository — in which case, you’d create a custom base template. You may also find it useful to separate out common fields to individual templates, and then re-use among product types.
2. Product variants should only vary in values that are stored in dropdown lists (i.e., Lookup Values). Although possible, you generally should not use free-text fields to define variant data.

Adding Fields which Reference Other Products

If your product type needs to include fields which reference other products, we recommend the use of a *Multilist with Search* fields type, especially if your Product Repository utilizes item buckets. A *Multilist with Search* allows the content editor to search for products by SKU or other keyword. When creating the template field, use the following *Source* value to ensure the search is limited to products within /sitecore/commerce/products:

```
StartSearchLocation={0A702337-81CD-45B9-8A72-EC15D2BE1635}
&Filter=_templates:7DD89481AAFA4EC982BF2AC3047049B2
```

The Product Repository

Active Commerce stores product information as items in the Sitecore content tree. These are grouped as a collection of products under a Product Repository. Each repository can contain products and categories. You can create a structure that suits the needs of your product range. All repositories must be located under “sitecore/Active Commerce/Product Repositories.” In general, you will have at least one repository per Active Commerce instance, but you may have

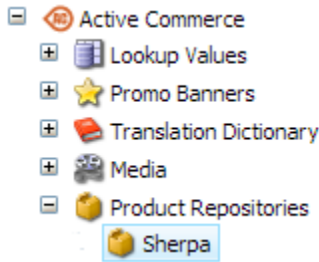


more if additional sites are added.

Bucket Repositories allow you to take advantage of Sitecore Item Buckets to create an unstructured product repository. You can also utilize item buckets on individual repository categories. For more information, review [Sitecore 7 documentation on Item Buckets](#).

Adding a Repository

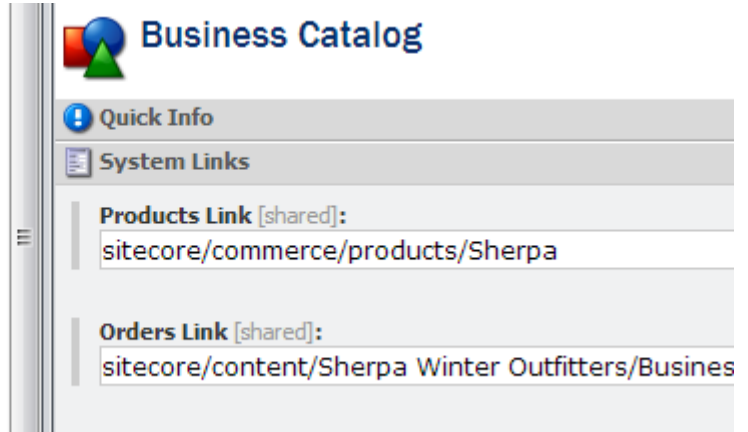
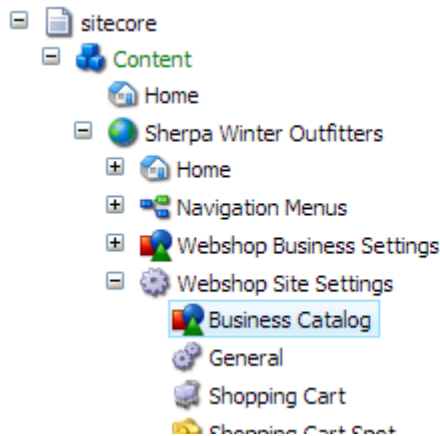
1. In the content tree, select “sitecore/Active Commerce/Product Repositories.”
2. Insert item > Repository.
3. Enter the name of your repository and click OK.



Assigning a Repository to a Site

This determines what products are available when defining product categories on the site.

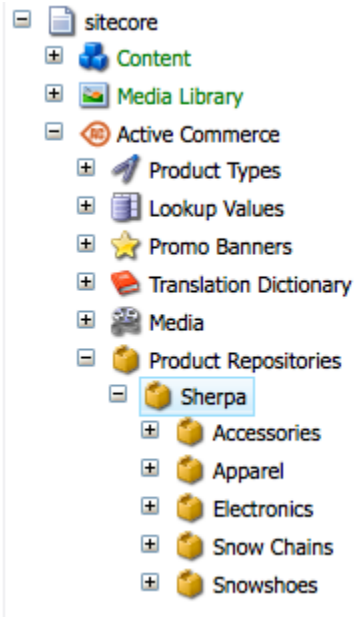
1. In the content tree, select the “Business Catalog” item for your Active Commerce site (e.g., “sitecore/Content/MySite/Webshop Site Settings/Business Catalog”).
2. For the “Products Link” field, select the repository you’d like to use.



Adding Repository Categories



1. In the content tree, select the repository or a category.

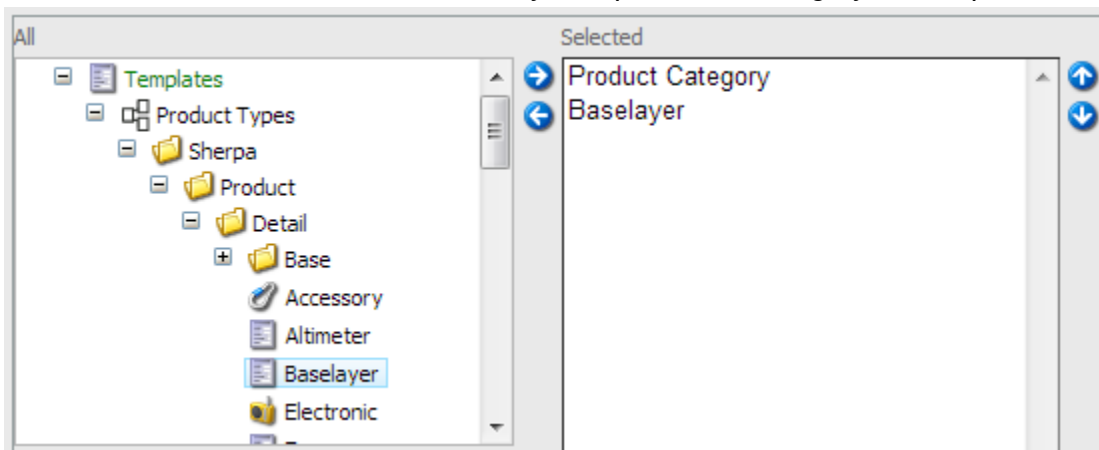


2. Insert item > Product Category.
3. Enter the name of the new category item (e.g., Electronics, Accessories, Spare Parts, etc.) and click OK.
4. Configure the insert options as necessary to define specific Product Types. This is especially important when it comes time for product entry.

NOTE: The Product base template will be added by default. This may be fine, but in most cases, you will have created specific product types that you will want to use instead.

- a. Select the Configure tab on the Ribbon.
- b. Under Insert Options, click Assign.
- c. Move the desired product types to the right pane, and click OK.

NOTE: It's recommended to always keep Product Category as an option.



5. Repeat these steps as many times as you need to build out your category structure.



The Product Catalog

The Product Catalog represents how your products are presented to the world. Here, we define the categories, which products are in those categories, along with how those products can be filtered and compared. When a new Active Commerce site is added, the product catalog item is located directly under the “Home” item (e.g., “sitecore/Content/MySite/Home/Products”).

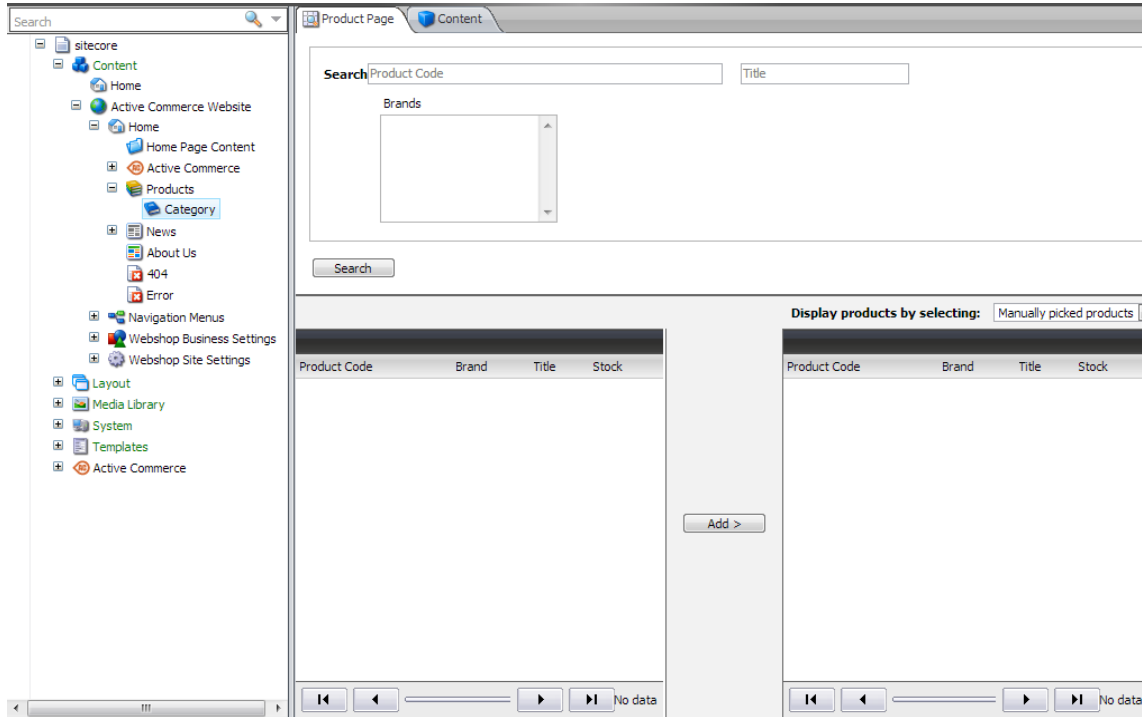
Catalog Categories

- A catalog category is defined by the `/sitecore/templates/ActiveCommerce/Categorization/Category` template.
- A category can take one of 2 forms:
 1. **Category with Products** – This is a category which displays a list of products.
 2. **Category of Categories** – This is a category which displays a list of sub-categories.
- Whether a category is one or the other is determined by whether products are assigned (Category with Products) or not (Category of Categories).
- The default presentation details (defined on Standard Values) uses a conditional rendering to determine which sublayout (“Category - Product List” vs. “Category - Category List”) to render. If it is a category with products, then a product list will display; otherwise (a category of categories), sub-categories will be displayed.

NOTE: See Active Commerce – User Guide, “Assigning Products to the Product Catalog” section, for instructions on adding categories and products to the catalog.

The “Product Page” Tab

The “Product Page” tab is a custom editor that comes packaged with Sitecore SES (defined on “Product Search Group” template — appears on catalog category items). Active Commerce fully supports the use of this editor out of the box as it can be a very powerful tool; however, it’s not without its shortcomings.



The good:

- Built-in product repository search.
- Can customize search form fields to suit client.
- Query-based categories. This makes categories such as specific Brand easy. This also means potential for “auto-updating” product listings (i.e., as products are added to the repository, you don’t need to also add to catalog categories).

The bad:

- Does not support sorting selected products.
- Clunky UI (Paged results, odd resizing, missing “Remove” button).
- Tricky to configure (search doesn’t always display expected results).

NOTE — Not really good or bad, but also worth noting: the “Products” field is used as the backing data store, so it will not work as expected if you modify that field manually.

The alternative is to turn this tab off, and instead, simply use the “Products” field to manage your products. This field (a Treelist) *does* support sorting. Another option is to take this a step further and change the “Products” field to a *Multilist with Search* or *Treelist with Search*.

It is completely up to each Active Commerce implementation to decide if you’d like to use it or not. Here are some guidelines to help you choose:

- If you require query-based categories, then use the “Product Page” tab.
- If being able to *sort* the products is a requirement, then consider removing the “Product Page” tab.



Removing the “Product Page” Tab

1. Ensure that “Standard Fields” is checked under View Ribbon.
2. In the content tree, go to “/sitecore/templates/Ecommerce/Product Categories/Product Search Group/___Standard Values”
NOTE: This can instead be done per category if you do not want it removed from all categories.
3. Under the “Appearance” section > “Editors” field, click “Edit.”
4. Double-click the “Product Page” item (or use left-arrow) to remove from the Selected pane.

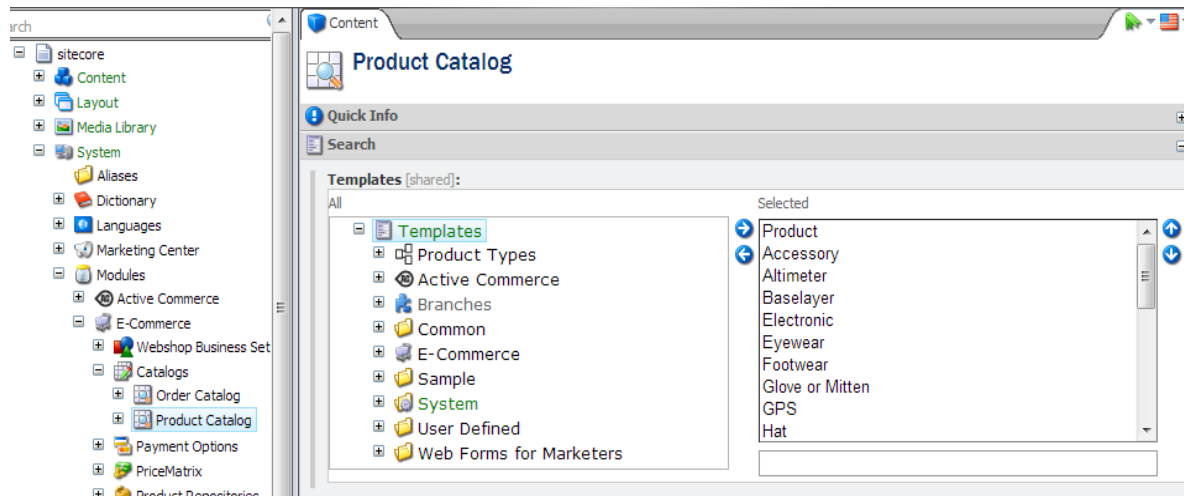
NOTE: The remainder of this section describes configuration of the Product Page tab. You can skip if not using.

Configuring Product Templates

In order for your custom product types to appear on the Product Page tab, you must add them to the “Product Catalog” SES item.

NOTE: This Product Catalog item is also used to display products on the “Product Catalog” tab of a repository category item.

1. In the content tree, go to “/sitecore/system/Modules/Ecommerce/Catalogs/Product Catalog.”
2. Under Product Types, select all your custom product templates. For product families, only add the *variable products (parents)*, NOT the *product variants (children)*.



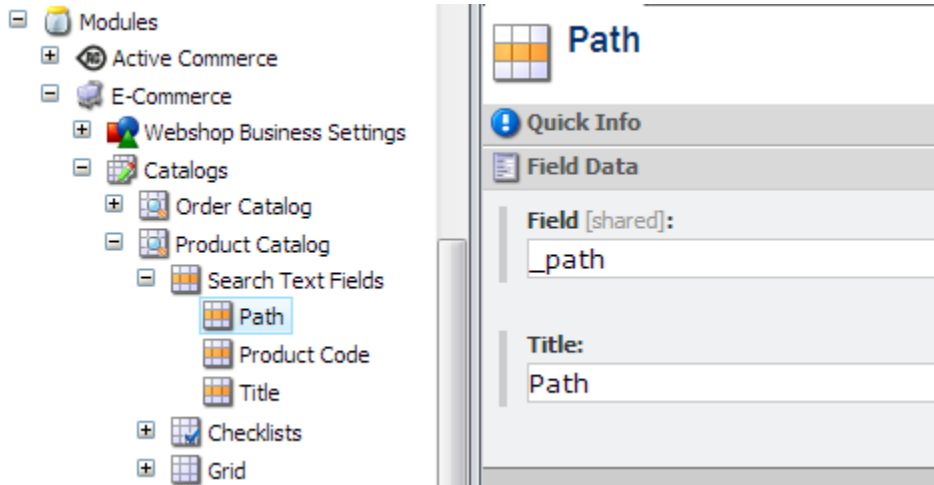
Customizing the “Product Page” Tab

Please refer to the [Sitecore E-Commerce Services 1.2 - E-Commerce Installation and Configuration Guide](#) (section 4.6.1 Catalogs) for complete customization details, including adding search form fields, and modifying the grid columns.

To illustrate what’s possible, let’s run through a useful example: **linking catalog categories to repository categories.**

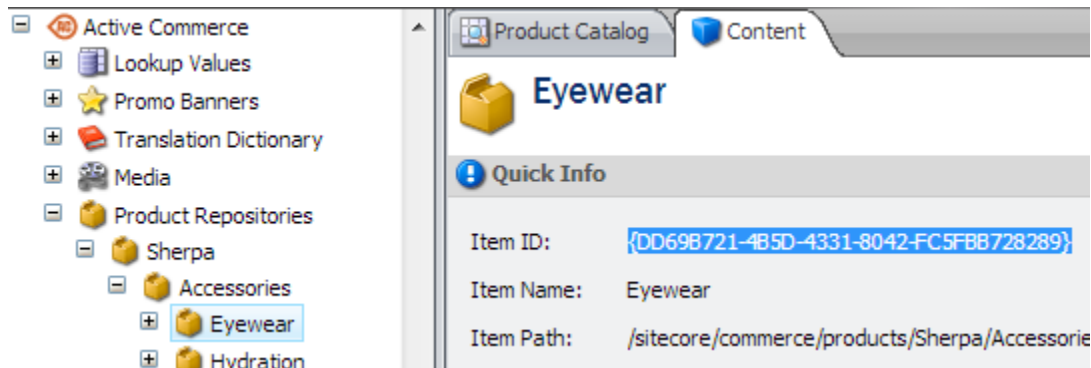


1. In the content tree, go to “/sitecore/system/Modules/Ecommerce/Catalogs/Product Catalog/Search Text Fields.”
2. Insert item > Search Text Field.
3. In the “Field” field, enter “_path”. This is a built-in Sitecore field that *contains ShortIDs for every parent item in the path relative to current item.*
4. In the “Title” field, enter “Path”.

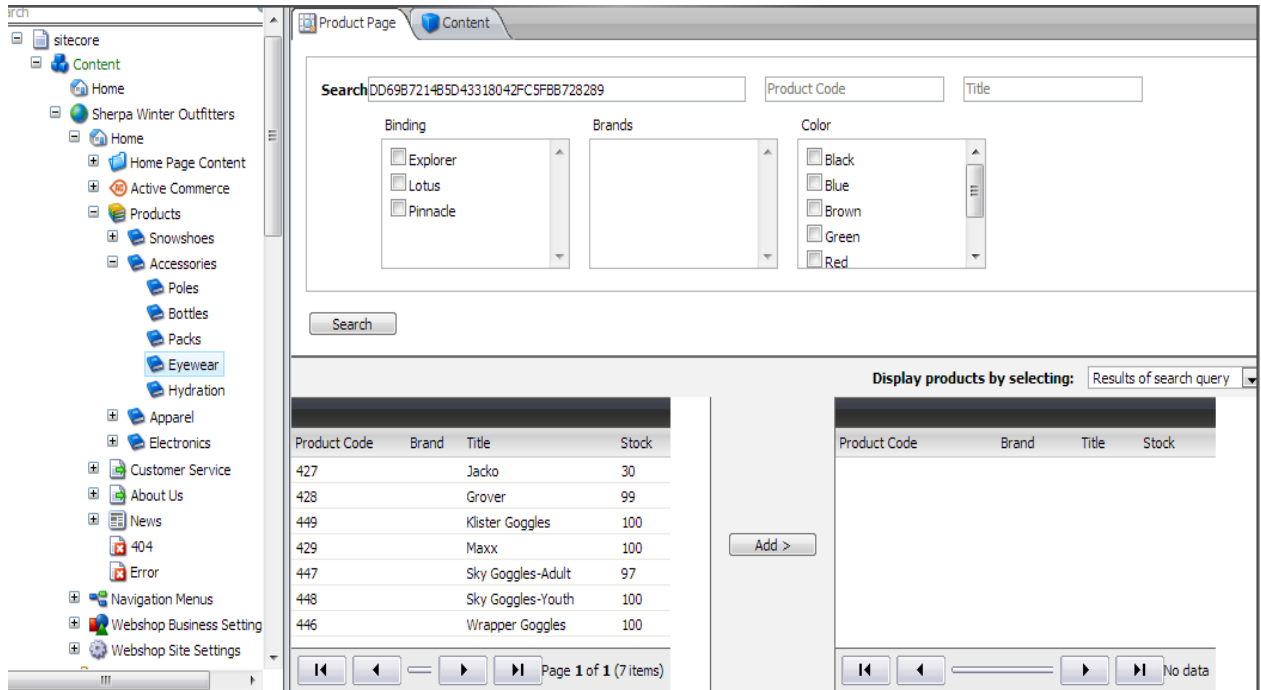


5. Now we can create query-based categories using this new “Path” search field.

For example, let’s say I have a repository category called Eyewear (note the ID of this item):



Now, let’s say I also have a catalog category named Eyewear. I want every product in my Eyewear repository category to appear here. Rather than manually adding each product, we can enter the ShortID of the repository category in the “Path” search field, and set to display products using results of search query.



With this setup, any time products are added or removed from the repository category, they will automatically be included under the Eyewear category on the site. Nice!

Configuring Product Compare

NOTE: This only applies to Categories with Products. If Compare is added to a Category with Categories, it will have no effect.

1. In content tree, select a catalog category > Insert Item > Compare Folder.
2. Select the compare folder > Insert Item > Compare Field.
3. In the “Field” field, select the product field (see def) you’d like to compare on.
4. In the “Display Name” field, enter the display name of the comparison (appears along the left of the product comparison modal window).

Configuring Product Filters

NOTE: This only applies to Categories with Products. If Compare is added to a Category with Categories, it will have no effect.

1. In content tree, select a catalog category > Insert Item > Filter Folder — name it “Filters” (**IMPORTANT: The folder MUST be named ‘Filters’**).
2. Select the filter folder > Insert Item > Filter Field.
3. In the “Field” field, select the product field (see def) you’d like to filter on.
4. In the “Display Name” field, enter the display name of the filter (appears on the left sidebar of the product list page).
5. In the “Tooltip” field, enter a tooltip (appears on hover of the filter display name).



6. Check the “Exclusive” checkbox to enforce only one selected value at a time.

