

# Cached Dependent Filters

- [Overview](#)
- [Use](#)
- [Setup](#)

## Overview

[top](#)

In order to ensure that User Prompt filter value selection is as easy as possible for the user, it's a good idea to cache your filters, creating lists, rather than prompting them to enter values manually. It's also helpful to create dependencies between filters that are related.

For example, if you have a Product Type and Product Code filter, making the Product Code dependent on Product Type will allow the user to nominate a type, and only see the products that belong to that type, rather than the whole list.

## Use

[top](#)

In this example we have created a dependency structure based on Camp locations: **Camp Region > Camp Country > Camp Name**

1. Select a **Region = Asia** to see a list of countries we have records for in Asia.
2. Select **Region = Europe** to see a list of countries we have records for in Europe.
3. Select **Camp Country = Spain** to see a list of camps we have records for in Spain, Europe.
4. Select **Camp Country = Germany** to see a list of camps we have records for in Spain, Europe.
5. Select **Camp Country in Austria, Germany, Spain** to see a list of camps we have records for in Austria, Germany, or Spain, in Europe.

Filters

Camp Region

In List

Asia

Australia

Europe

Latin America

North America

✓

✗

Reset

Go

## Setup

[top](#)

1. Add the required filters to your report
  - a. Make sure the filters are listed in order of dependency; this will make it clearer to the user
  - b. Leave the filters as User Prompt so they are available for the user

Filters

Camp Region

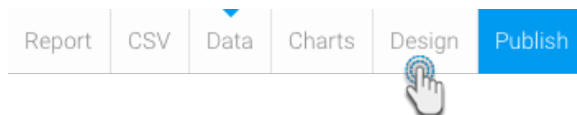
Camp Country

Camp Name

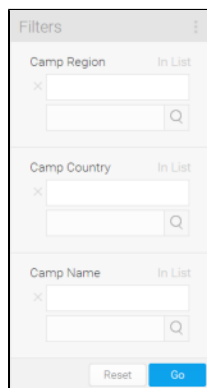
Advanced settings

In this example, we've used the **Camp Region**, **Camp Country**, and **Camp Name** as filters, all set to **In List** in order to allow the user to select multiple values if required.

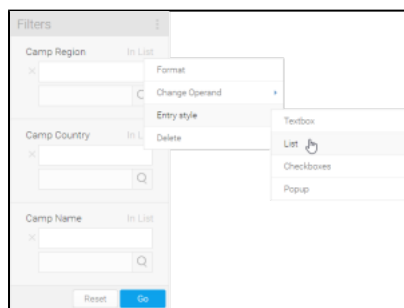
2. Proceed to the Design step of the report builder



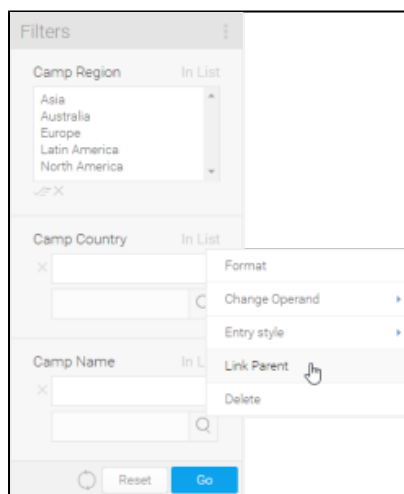
3. Here, you should see all of your filters displayed as User Prompt with no pre-defined values of any kind.



4. Set the first filter in your structure to use **List** as the **Entry Style**. What this does is change the filter to display **cached** values.



5. Now, on the second filter, click on the **Link Parent** option.



6. In the **Set Parent** window, select the first filter in your structure. In this case we set **Camp Country**'s parent to be **Camp Region**.

**Note:** if you don't see the filter you need as the parent in the drop down list your first filter has not been cached correctly or is an incompatible type.

**Set Parent**

Link the **Camp Country** filter to a parent in order to create a relationship as part of a filter hierarchy.

Parent Filter: -- Select --  
 -- Select --  
**Camp Region**

**i** You cannot link to Text Entry, Date Picker or Numeric Filters

**Submit**

7. Once you have defined the parent, you will be able to select the dependency options. In this case we set both to **on**.

- a. **Dependent Values:** this means that your child filter's list of values will change to only display values related to the items selected in your parent filter's list.
- b. **Dependent Display:** this means that your child filter (**Camp Country**) will not be displayed at all until you have selected at least one value in the parent (**Camp Region**).

**Set Parent**

Link the **Camp Country** filter to a parent in order to create a relationship as part of a filter hierarchy.

Parent Filter: Camp Region

**Dependent Values** ☒  
 Only values related to the selected parent values will be displayed.

**Dependent Display** ☒  
 This filter will only be displayed once the parent filter values have been selected.

**Submit**

8. **Submit** your selections.

9. Now, on your third filter, click on the **Link Parent** option.

**Filters**

**Camp Region** In List  
 Asia  
 Australia  
 Europe  
 Latin America  
 North America

**Camp Name** In List  
 × [ ] [ ]  
 Format  
 Change Operand  
 Entry style  
**Link Parent**  
 Delete

Reset

10. In the Set Parent window, select the second filter in your structure, the one directly above the current filter. In this case we set **Camp Name**'s parent to be **Camp Country**.

The reason we select **Camp Country** as the parent, rather than **Camp Region**, is to ensure any selections the user makes in the Country filter have an effect on the **Camp Name** Filter. Setting the dependency this way will mean that **Camp Name** inherits the **Camp Country** dependency on **Camp Region** automatically.

**Set Parent** ✕

Link the **Camp Name** filter to a parent in order to create a relationship as part of a filter hierarchy.

Parent Filter -- Select -- ▼

i You cannot link to Text or Numeric Filters

**Camp Country**

**Submit**

11. Define your dependency options and submit your selections.

**Set Parent** ✕

Link the **Camp Name** filter to a parent in order to create a relationship as part of a filter hierarchy.

Parent Filter Camp Country ▼

Dependent Values ☑  
Only values related to the selected parent values will be displayed.

Dependent Display ☑  
This filter will only be displayed once the parent filter values have been selected.

**Submit**

12. You will now see your **Camp Region** filter.

**Filters** ⋮

**Camp Region** In List

Asia  
Australia  
Europe  
Latin America  
North America

✓✕

⌂ Reset Go

13. Make a selection on the **Camp Region** filter and you will see your **Camp Country** list appear with values related to the region(s) you selected.

**Filters** ⋮

**Camp Region** In List

Asia  
Australia  
**Europe**  
Latin America  
North America

✓✕

**Camp Country** In List

Austria  
Germany  
Spain  
Finland  
France

✓✕

⌂ Reset Go

14. Make a selection on the **Camp Country** filter and you will see your **Camp Name** list appear with values related to the country(ies) you selected.

Filters

Camp Region

In List

Asia

Australia

Europe

Latin America

North America

Camp Country

In List

Austria

Germany

Spain

Finland

France

Camp Name

In List

Mt St Saviour

Mt Tremblant

Nobeyama

Reset

Go

[top](#)