

Filtering data using Data Access Control

(From version 3.4.2)

Often you are able to filter data directly in an SQL query using a WHERE statement, or from Microsoft CRM by adding a condition to your view, but sometimes you are unable to filter the source data coming into fData because it has come from a web service, flat XML file or some other source that does not allow filtering during requests. Or, it may be that because of the complexity of what you want to do, editing the SQL query or setting up MSCRM conditions would get too complicated.

Traditionally, the way to filter data that is returned from queries made is to add on a [Filter type Data Group](#) to your DSD and use that to create a set of records based on filter criteria you specify, and then use the new Data Group to report from.

However, this can be complicated when using multiple rules, or if you are unfamiliar with xPath notation.

From version 3.4.2 forwards, fSeries includes [Data Access Control](#). This facility is intended to give you the option to remove rows of data, or even individual fields within a row, depending on the role/access of the user making the request. However, because DAC can be used with fSeries functions, this gives you the option to also filter rows/fields based on results from other Data Groups, or on User Entries coming into the DSD.

This guide will show you how to create multiple filters from [User Entries](#), and use them to quickly filter a set of records coming back.

Example Scenario

In the examples below, we have a set of data coming back from a source that does not support filtering during data gathering (i.e: a web service). The data contains a set of related family members to a client, that looks like this:

FirstName	Surname	Age	Location
JOHN	SMITH	30	LONDON
KAREN	SMITH	35	LONDON
ANN	OTHER	45	BIRMINGHAM

Switching on Data Access Control

Firstly, ensure DAC is switched on for your copy of fSeries by [logging into fAdmin](#) and go to *Settings > Data Access Control* and ensure the setting for

"DataAccessControl" is checked. You can then open [fData](#) to being working on your DSD.

Working on the DSD

In our DSD we currently have one [Data Group](#) called "RelatedFamily" that returns the data shown in our example scenario. We would like to add three filter options for our end user – the ability to filter by Surname, Location, and Age Range From/To.

User Entries

The first step is to add our [User Entries](#) that will allow the user to enter the criteria. We'll create **four** User Entries – one called "SurnameFilter", one called "LocationFilter", one called "AgeFromFilter" and one called "AgeToFilter" (remember to set the Type of the User Entry for the last two entries to be a *Number*). You do not need the word "Filter" in your User Entry IDs, but it can make them easier to reference in the future.

Adding the Surname/Location Filters

Having opened our Data Group, open the Access Control tab (which is located below Data Items). Now, from the ribbon at the top click the "Access Control" button located in the "Add New" group.

In the condition box, type in the following fSeries function:

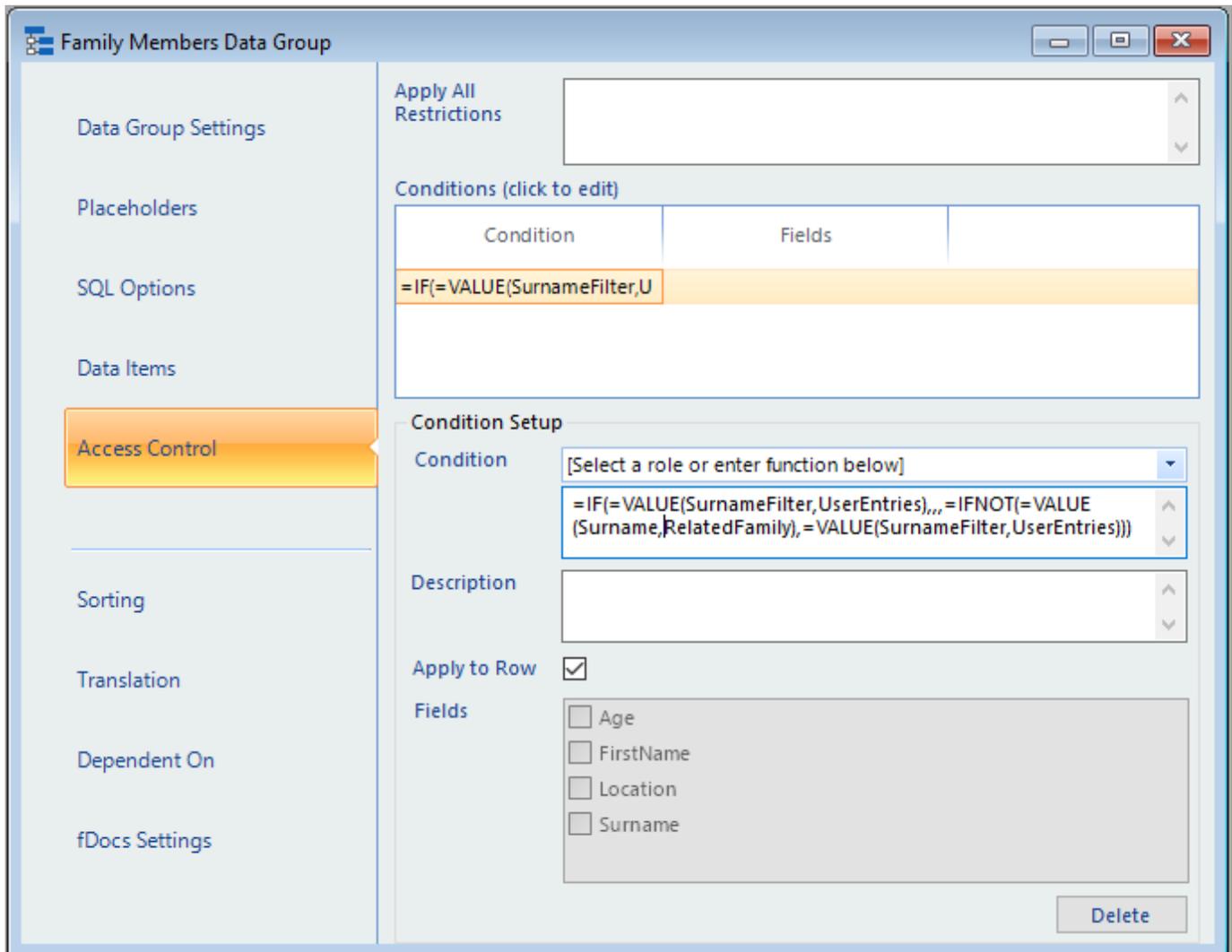
```
=IF(=VALUE(SurnameFilter,UserEntries),,,=IFNOT(=VALUE(Surname,RelatedFamily),=VALUE(SurnameFilter,UserEntries)))
```

Although this function may look intimidating, it can be broken down as follows:

"IF the VALUE of my SurnameFilter field in my UserEntries is blank, do nothing, otherwise IF the VALUE of the Surname field in my RelatedFamily data group does NOT match the VALUE of my SurnameFilter field in my UserEntries, this condition is true and do something!"

What we then need to tell fData to do is what to remove. In this case we don't want the row of data to appear, so click the "Apply to Row" box. If you want to just filter out an individual field/fields, you can check which fields you don't want to appear from the list below. This gives very fine detail over what can be removed.

Your Data Group window should now look like this:



There's nothing else you need to do as any changes immediately take effect. Now we've added the filter for Surname, we can go back and add another Access Control condition by clicking the ribbon in the top, and use the same method to add a filter for Location.

Add a From/To Filter Range

The process for this is very similar to the process for adding regular filters, except here we need to add **two** filters, one to check **from** a value, and one up **to** a value.

The two conditions would be thus:

```
=IF(=VALUE(AgeFromFilter,UserEntries),,,=COMPARE(=VALUE(Age,RelatedFamily),lt,=VALUE(AgeFromFilter,UserEntries),1))
```

"IF the VALUE of the AgeFromFilter field in my UserEntries is blank, do nothing, otherwise COMPARE the VALUE of the Age field in my RelatedFamily data group (the last "1" at the end of the function indicates a numeric comparison) and if it's LESS THAN the VALUE of the AgeFromFilter field in my UserEntries, the condition is valid and do something!"

and:

```
=IF(=VALUE(AgeToFilter,UserEntries),,,=COMPARE(=VALUE(Age,RelatedFamily),gt,=VALUE(AgeToFilter,UserEntries),1))
```

“IF the VALUE of the AgeToFilter field in my UserEntries is blank, do nothing, otherwise COMPARE the VALUE of the Age field in my RelatedFamily data group (the last “1” at the end of the function indicates a numeric comparison) and if it’s GREATER THAN the VALUE of the AgeToFilter field in my UserEntries, the condition is valid and do something!”

You’ll note from the above that the conditions filter out rows/columns of data, not filter in.

Going Further

By now you’ll have your filters all set up, and you can test your DSD. You’ll see that if you enter a value in the relevant user entry, the data returned will have filtered out everything which doesn’t match what you’ve entered.

The examples above will meet most of your requirements, but of course you can use any [fSeries functions](#) you wish to ensure the data filters out what’s needed.

Finally, the user entries above will appear as basic text boxes to the user. You could [turn these into lookups](#) if you wanted so that the user would see a drop down with a unique set of values to choose from. This is achieved by creating a secondary DSD that serves to provide the drop down values needed. See our help guides for more details.