

Database Filter Contracts

Note that a number of these contracts are used by the Data form wizard and should not be modified.

This document explains how to use the database filter contracts by using a real world example.

Note: These contracts are in no way related to the DFW contracts and should not be used interchangeably with the Database Filter contracts that are automatically applied by the DFW. DFW contracts are denoted by “(DFW)” in the plain english phrase.

What are filters?

Filters are used to reduce the number of records displayed in the Recordset to show only the records that meet certain criteria. Suppose you have a Recordset of information about students and their test grades. Each record contains the student's name, age, date the test was taken, and the grade the student received on the test. If you have an AutoTable on a page that is bound to this Recordset, every record in the Recordset will be displayed. But what if you want to see only the records that show all the test scores of the student named Jim? What if you wanted to see all the test scores of a test that was taken on December 9, 1998? Well, you could have two separate Recordsets that contain the records you want, but that is not very convenient. You can, however, apply a filter to the Recordset so only the records you want to see are displayed. The English phrase for our first filter would be, “Show me all the records in the Recordset where the name of the student is Jim.” The English phrase for the second filter would be, “Show me all the records in the Recordset where the date the test was taken is December 9, 1998.” You can even combine phrases to filter the database even further such as, “Show me all the records in the Recordset where the name of the student is Jim *and* the date the test was taken is December 9, 1998.” You can expand your search by applying a filter such as, “Show me all the records in the Recordset where the name of the student is Jim *or* the name of the student is Nancy.” By applying different filters to the same Recordset on the same page, you allow users to filter the Recordset however they like.

How do I implement filters in Drumbeat?

In Drumbeat, you apply a filter to a Recordset by assigning various Database Filter interactions. You can apply simple filters that only filter the Recordset based on one field, or you can assign complex filters using the connecting words “and” and “or” that filter the Recordset based on the values of two or more fields. Suppose we have a simple web application with two pages. The first page is named Search and the second page is named Results. The Results page is just an AutoTable that is populated by the Recordset called Recordset1. The Search page will have various form elements so users can enter values to filter the Recordset.

Important Note: Implementing Filters in Drumbeat is a two step process. First you build the filter by adding the “Add <form element> to filter of <Recordset> when <button> is clicked” interactions for each of the form elements in your search. Second,

you must apply the filter to the Recordset by adding the “Filter <Recordset> when <source> is clicked” interaction.

A simple filter using one field in the Recordset

Put an edit box named Edit1 on the Search page. Also put a button of type Submit on the page and the Recordset. Make sure that the Action of the Form object is set to No Link or to the Search page. Select the edit box, the form button, and the Recordset and assign the interaction “Add Edit1 edit box to filter of Recordset1 when FormButton1 is clicked”. A Parameters dialog box will appear with four different parameters to assign. For this simple filter, all we need to worry about are the Recordset Field and the Operator. For the Recordset Field parameter, select the Name field name from the dropdown list. For the Operator parameter, select the equal sign. Now select the submit button and the Recordset. Assign the interaction “Filter Recordset1 when FormButton1 is clicked”. For the “Go to page” parameter select the Results page.

Now publish your site and browse the Search page. If you enter “Jim” in the edit box and click the button, you will be taken to the Results page and only records for Jim will appear in the AutoTable. If you go back to the Search page and leave the edit box blank and click the button, records for all students will appear on the Results page.

A filter using two fields in the Recordset

Now suppose you want to filter the Recordset based on the student’s name and the date the test was taken. Put another edit box, Edit2, on the Search page. Select the new edit box, the form button, and the Recordset. Assign the interaction “interaction “Add Edit2 edit box to filter of Recordset1 when FormButton1 is clicked”. For the Recordset Field parameter, select the TestDate field name from the dropdown list. For the Operator parameter, select the equal sign. For the Connector parameter, select “And”.

Publish your site and browse the Search page. In the first edit box enter “Jim” and in the second edit box enter “12/9/1998”. Now when you click the button and go to the Results page, you will only see records of tests that Jim took on December 9, 1998. If you go back to the Search page and delete “Jim” and resubmit you will see all of the records of tests taken on December 9, 1998.

Order of execution is important

In the above example, it is very important that the clause of the filter pertaining to the student’s name be added to the filter *before* the clause of the filter pertaining to the date of the test. You can check on the ordering of the execution of interactions on the Recordset by right-clicking on the Recordset and selecting the “Assigned Interactions (ordered)” option. If the interactions are in the wrong order, you can drag them to be in the correct order. If you wanted to, you could add the date clause to the filter first. But then you would have to change its Connector parameter to be blank and change the Connector parameter of the Name clause to be “And”.

Clause Separators

There may be times when you want to separate the different clauses in your filter with parentheses. If this is the case you can assign the Clause separator parameter to be either an open parenthesis or a close parenthesis. However, if you do this, it is recommended that you use the Non-blank Validation contract on the edit boxes. Because if the user leaves an edit box blank that uses Clause separators, you will get an ASP error on the Results page.

Using Check Boxes

Suppose there is a field in our database called “Passed” that is a yes/no field. We could add a Check box to our Search page and add it to the filter. To add it to the filter, select the Check box, the form button, and the Recordset and assign the interaction, “Add Check1 check box to filter of Recordset1 when FormButton1 is clicked”. For the Recordset Field parameter, select “Passed”. For the Connector parameter, select “And”.

Publish your site and browse the Search page. Now the filter will always include a clause for the Passed field of the Recordset. If the Passed Check box is checked, then all of the records displayed will have a Passed value of yes. If it is not checked, all of the records displayed will have a Passed value of no.

Using Lists

Using Dropdown lists with filters is just like using edit boxes. But, if you want to use a list box which allows multiple selections to be submitted, you have to use the contract, “Filter [Recordset] with [value in] list value when [Button] is clicked (Multiple Values)”. Due to limitations in a Recordset’s Filter method, this contract must be used all by itself. You cannot use other filter contracts with this contract.

Using Session Variables

You can also use a Session Variable in your filters. Suppose our small web application is part of a bigger web application. There could be a logon page where the student is asked to enter his name. The student’s name will then be stored in a Session Variable named “StudentName”. Now, when they get to our Search page, we don’t have to ask them to enter their name again because it is already being stored in a Session Variable for us. So, to take advantage of this fact, we do as follows. Delete the edit box where the user used to type their name. Put the Session Object in the basement. Select the Session Object, the form button, and the Recordset. Find and apply the Database Filter contract, “Add value in Session variable to the filter of <Recordset> when <button> is clicked”. For the “Variable Name” parameter, type “StudentName”. Assign all the other necessary parameters.

Now publish and browse. When you browse the Search page, there is no place to enter your name, but when you click the submit button, you will only be able to see records of tests that you took. (To test this new filter, to make sure it works, put the following line in the Document’s Server event of your Search page: *Session(“StudentName”) = “John”*)

