

# fDocs for Microsoft Dynamics CRM

Gathering data from Dynamics CRM requires the setup of two elements: a CRM View and an fData DSD data group.

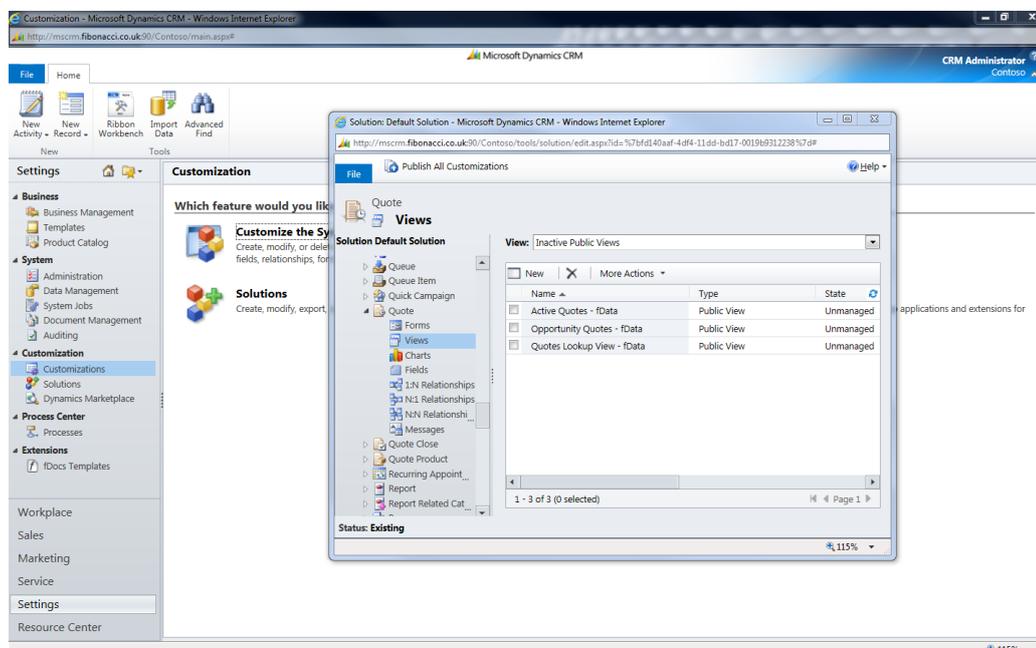
It is always recommended to keep data groups simple. A data group is a single block of data from one entity, or possibly a primary entity with other related fields associated with it. To define the data to be gathered you simply need to create and save an Advanced Find view for the entity from within CRM and mark it for fData's use.

Each such view can be used in many data groups, so the view is a fundamental element in the gathering of CRM data.

This document is a guide to creating the views, marking for fData use and incorporating them in fData data groups. Throughout we will use as an example, a DSD to gather data for a quotation. The examples are all with respect to the Quote entity.

## Creating the View

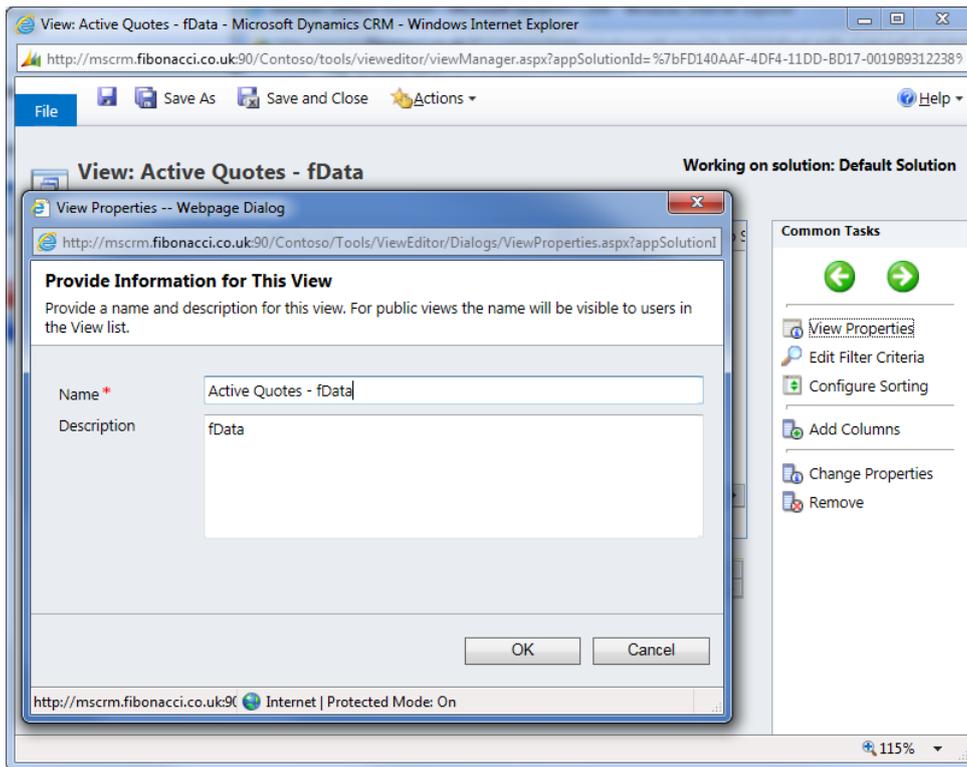
In this example a view has been created at Settings, Customizations, Customize the System, then expand Entities and further expand Quotes and finally click on Views.



Note that in this case the views used are inactive. This need not be the case. You can use any view you wish, even if it is used for other purposes, however, it is useful to be able to hide fData specific views in this way.

The content of the view is whatever you need to be gathered for your DSD. However, it is recommended that these views are kept relatively simple. It's

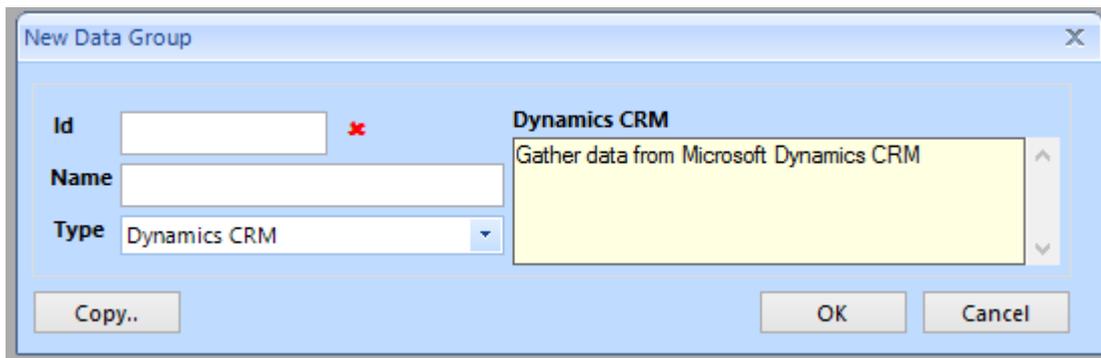
easy to add more views into the DSD.



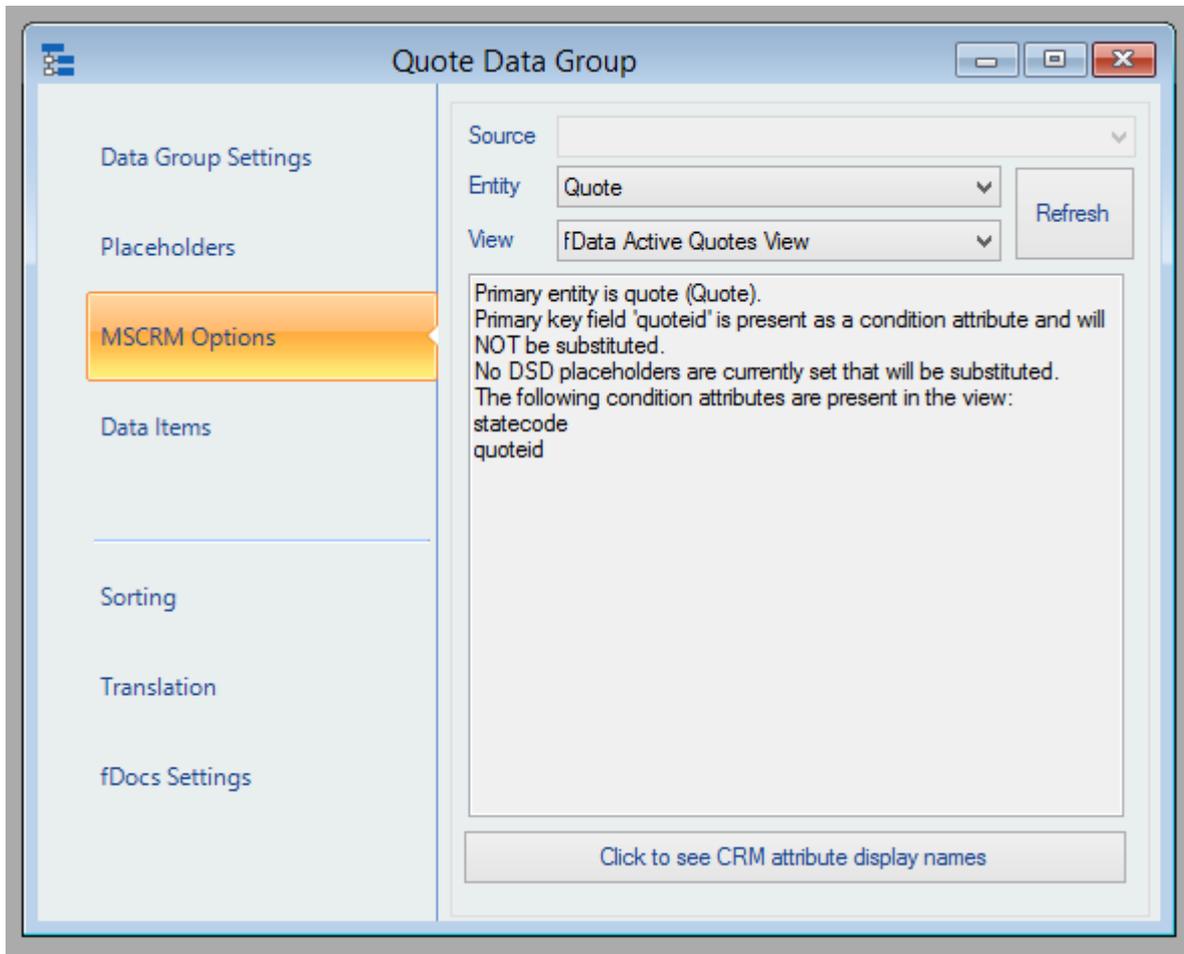
To make fData aware that the view is available for its use, the letters "fData" must be included somewhere in the Description property. In fact this is the default way to mark available views; see the section on adaptation for more information.

## Adding the Data Group

To add a new MSCRM data group to a DSD, enter its id and name and select the type from the list:



Next, select the view from the list available. By default the list shows all views marked for fData's use, so use the Entity option to filter the list.



The text area shows a summary of the view. Most importantly it shows the condition attributes that are applied in the view. In this case the “quoteid” is the primary key and is not currently available as a user entry or placeholder and so will not be substituted. A user entry or placeholder must be added in order to have a relevant value substituted when data is gathered. Please refer to the next section “Which Quote?” on how to specify the data you require.

## Which Quote?

When the DSD is executed it needs to be informed of the quote to retrieve. This is done through a “User Entry”. This doesn’t mean that the user actually enters the value, but that it is a value passed in from some other mechanism such as a URL link or a workflow.

The user entry **MUST** be exactly the same name as the entity’s primary key field, in this case “quoteid”. User entry names are case sensitive.

fSeries will automatically substitute the supplied value into the view query, so that details of the requested quote are returned, instead of whichever quote may have been used when creating and testing the view.

In fact, any condition in any view that specifies a field with the same name as a user entry will have the user entry substituted (this is also the case with placeholders that are discussed later). This means that when creating a

view you can add a condition which will be picked up by fSeries and substituted with a requested value simply by adding a matching user entry (or placeholder).

User entries are used to pass values into a DSD but sometimes you need to pass values from one data group to another. For example, if you wanted to add a data group with information about the account associated with the quote you could add it to the quote view or add a separate data group for the account. To do the latter you will need to tell the account view/data group which account to use. This is done by adding a placeholder to the account data group with the same name as the account primary key (accountid) which specifies getting the account id from the appropriate field in the quote (obviously your quote view must include a column for the account id).

So user entries and placeholders are the mechanism to pass values into a data group and then on to the corresponding view.

## Nested Data Groups

Sometimes you will want to specify that one data group operates within another. For example, if you added a data group for quote products and wanted to add a further data group for each product, you would need to nest the product one inside quote products data group. fSeries will then execute the products data group for every record found in the quote products.

This is done by specifying that the products data group has quote products as its parent. You will normally need to add a placeholder to the "child" data group to get the right record, in this case the quote product's product id.

An important issue here is that you would NOT nest quote products inside the quote data group. This is because both are at the top level, effectively "under" the quoteid user entry. This means that when using the DSD in fDocs Designer, the quote data group and the quote products are available directly from the tag list. If you place the quote products within the quote data group you could only add tags for quote products by first add a data group tag for the quote.

Nesting is actually a quite rarely used feature but is very powerful when it is necessary.

One final word on the way in which data groups are presented in fDocs Designer: if you expect there to be only one record (as in the quote) mark the data group as "Design As Top Level" in the fDocs section of the data group's settings. This means that all tags for the data group will appear at the top level and may be placed into the document directly rather than by including them within a data group (+) tag.