



## Cobalt Intelligent Query

### Those Pesky “Does Not Have” and “Not In” Queries in Advanced Find

Compatible with Dynamics CRM 2013, 2015, 2016, and Dynamics 365

Anyone who has used Dynamics CRM/365 for a significant amount of time has run across a limitation that has plagued the product since its introduction. It is the [number 1 most voted item](#) on the Microsoft Connect community site and although its status is currently "Resolved" it is still not possible to build a "Not In" or "Does Not Have" query in the native Advanced Find interface. With the release of Dynamics CRM 2013, Microsoft added the ability to execute a left outer join using FetchXML, but there is no reasonable way for an end user to leverage this new functionality.

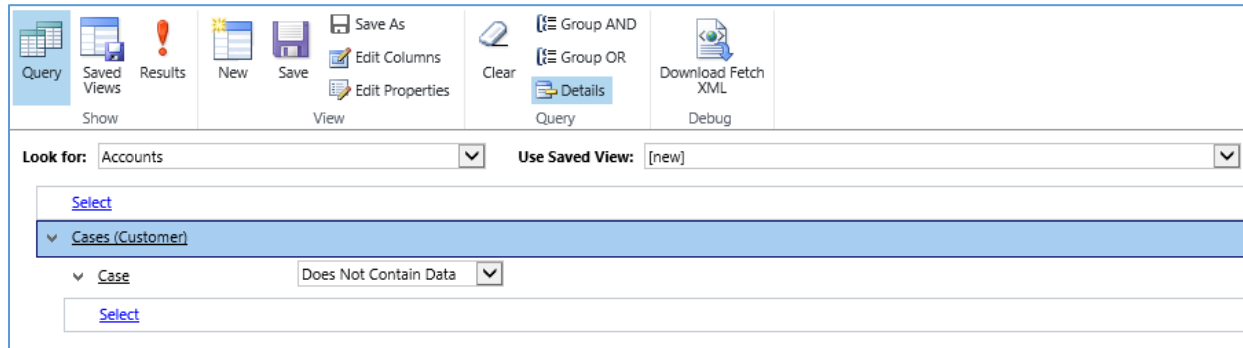
However, there are several workarounds.

1. You can [programmatically create views in CRM](#), but this requires a developer's assistance or an external application to modify and upload the XML and so you cannot create or update views on the fly.
2. You can create a marketing list that contains all of the records for the entity you're searching for and then filter out the records you don't want using a second query. Unfortunately, this will limit you to only accounts, contacts and leads and doesn't provide you with the ability to have the view change with time.
3. Finally, there is a [JavaScript workaround](#) that you can employ to modify the FetchXML that the advanced find window creates prior to executing the query, but you'll need to be comfortable with using JavaScript in your browser and this is technically not a supported CRM customization.

Seeing the limited options available and the vast need for a better solution, Cobalt decided to come up with a solution that was flexible and accessible for end users. We've named the solution Intelligent Query for CRM (<https://www.cobalt.net/category/crm-lab/intelligent-query>) and it's free to the CRM community. Intelligent Query (IQ) gives you the ability to create these queries using the native Advanced Find capabilities to infer a "Not In" query when you use a specific filter on a child record. The solution works out of the box on both custom and standard entities and allows for nested links and more complex conditions. It also works with Marketing Lists as well as ad-hoc Advanced Finds.

## How It Works

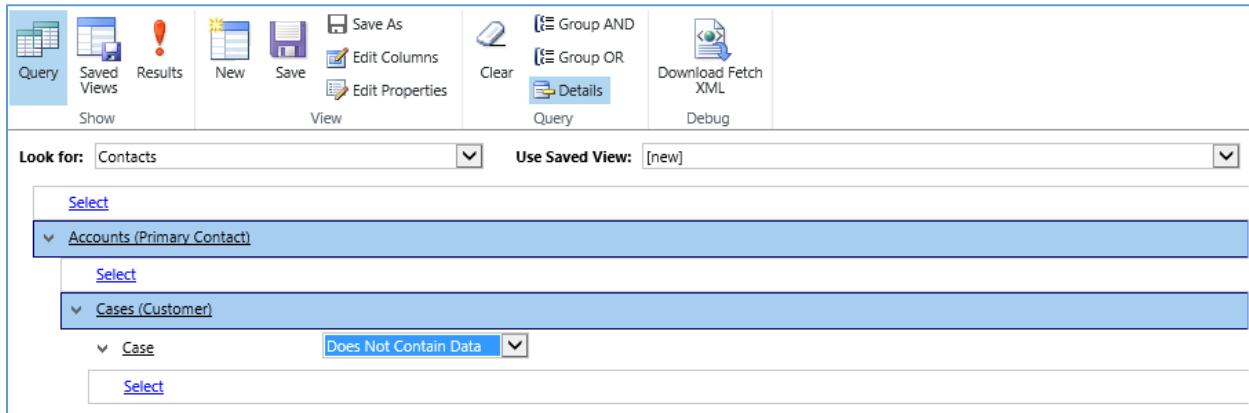
The simplest type of "Not In" query is asking the system to retrieve All Accounts with No Cases. For example, you are in charge of the support team at your company and need to find all customers (accounts) in CRM that have never submitted a support case so you can reach out to those accounts to find out if they are using our product and if they are satisfied with it. This can be achieved using CRM Intelligent Query and building the following Advanced Find.



Notice that we are specifying that the primary key of the entity we've linked to (Case), "Does Not Contain Data". This is the trigger that CRM Intelligent Query uses to infer a "Does Not Have" or "Not In" condition. Because this query would always return zero results (it's impossible for a case record to exist without the primary key) it is effectively useless under normal circumstances, but we're happy Microsoft included it. You can try it yourself on any type of entity in CRM and you will always get zero results. However, once you've installed CRM Intelligent Query, it will infer that what you intended by this query was to return all Accounts with No Cases, since retrieving all Accounts tied to a Case where there is no Case doesn't make sense.

## Using Nested Links

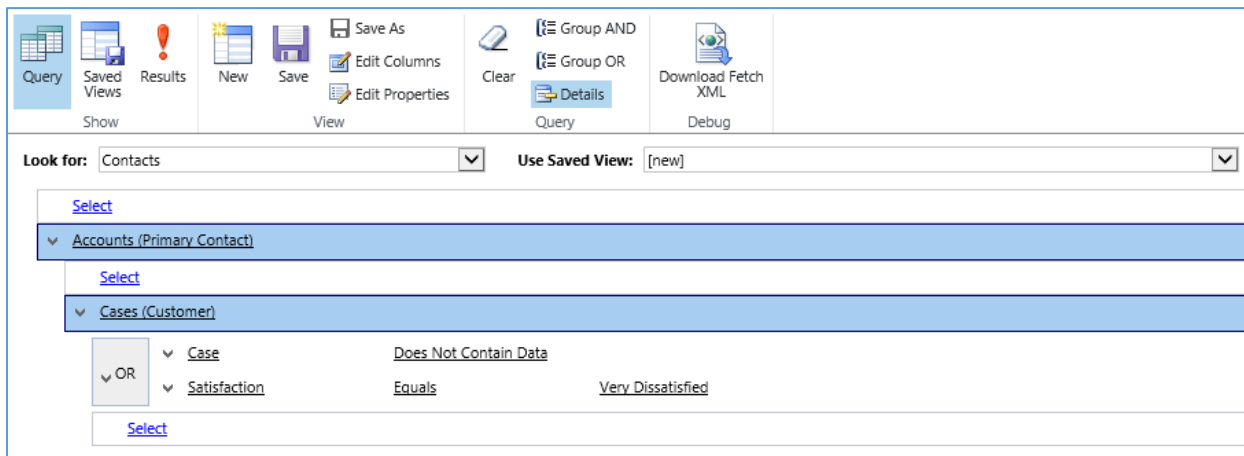
Now let's go a little deeper. Let's say the Account record, in this case, doesn't get you all the information you want because you'd like to know more information about the Primary Contact associated with this Account. We can take this Advanced Find a step further by using nested links to either a look-up value or a child record and build the following "Not In" Advanced Find.



Now, instead of finding the Accounts with no Cases, we are retrieving the Primary Contact of all Accounts with no Cases which is much more useful if you're looking to reach out to your customers. Wait. It gets better.

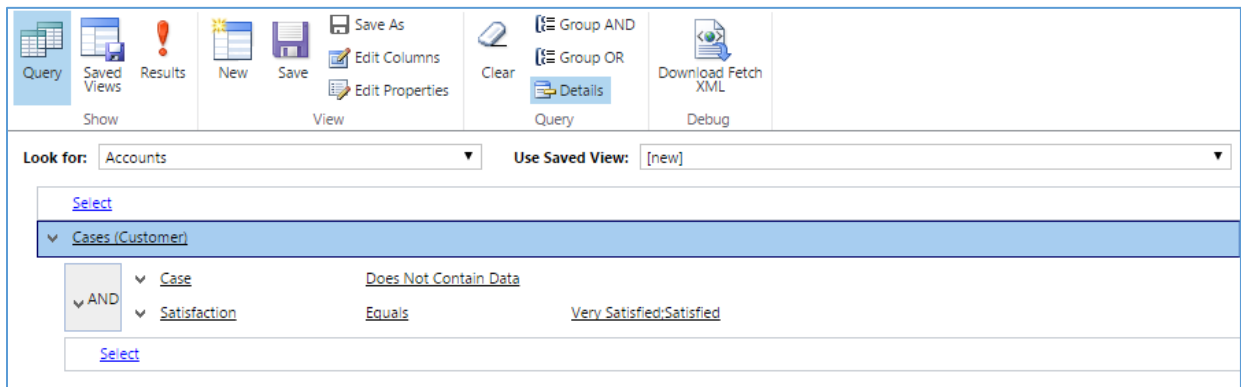
### Filters and Groups

Continuing with this example; let's say you not only wanted to find information about the Primary Contact of Accounts with no Cases, but also Accounts with cases that had a Satisfaction value of "Very Dissatisfied". With CRM Intelligent Query you can include additional filters and group them using an OR operator. So, in the Advanced Find below we are asking CRM to show us the Primary Contact of Accounts who have never submitted a case or have a case with which they were "Very Dissatisfied".

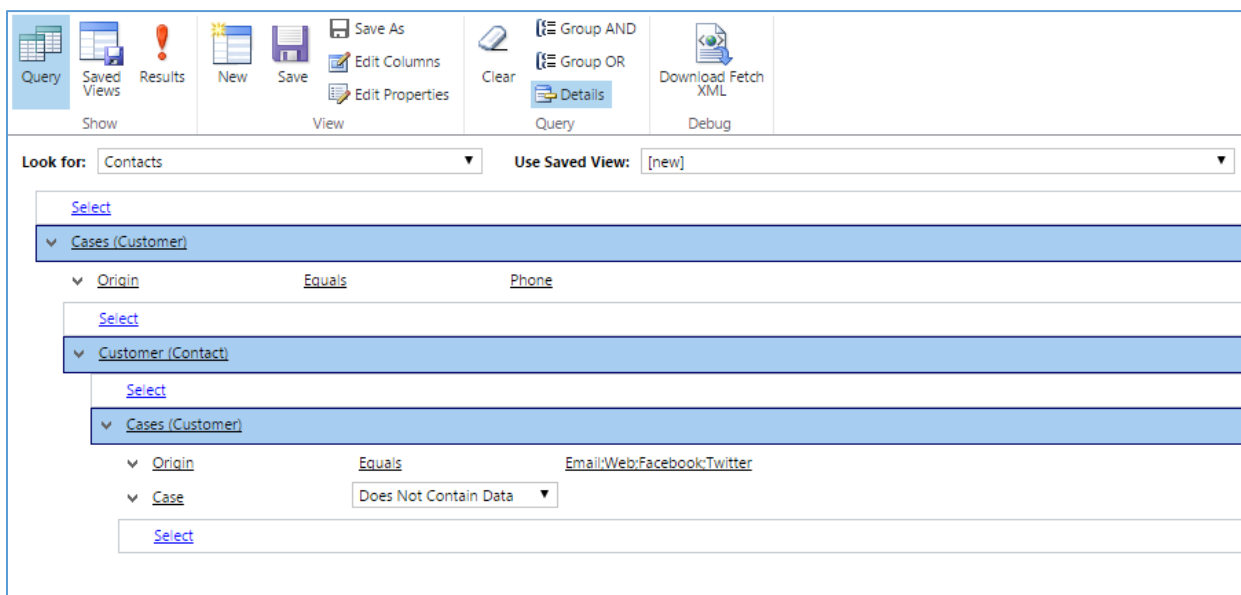


Okay, let's go crazy...

Now that you're familiar with the idea, let's try some different queries that are possibly even more useful. Consider, if you wanted to retrieve all **Accounts that have never had a case submitted with which they were "Satisfied" or "Very Satisfied"**. With CRM Intelligent Query you can build the following Advanced Find.



In this case we are using an 'AND' to group our conditions instead of an 'OR'. Now, you can start doing some really complex things with Advanced Find that simply are not possible out of the box. For example, in this last scenario, your support team is being inundated with calls and very few support requests are coming from your other web-based channels. You want to find all **Contacts that are only submitting cases by phone and have never used web-based methods for submitting cases**. With Intelligent Query you can build the following query using Advanced Find to search for these contacts and make them aware of the other options available to them.



Notice that we are linking to the cases entity twice in the above Advanced Find. That's not a mistake. By linking first to the cases entity and then back to the contact and finally back to the cases again we

are telling CRM that we only want to see contacts that have at least one case. Because without the first link we would be retrieving both contacts with no cases as well as contacts with no cases submitted via electronic methods. Does this seem a little complicated? Well, it is a little bit and it definitely takes some getting used to but once you've got the hang of it these types of queries will become second nature and will open up a lot of potential to your organization. Did I mention you can save these queries as views and it works for Marketing Lists as well?

## Configuring Intelligent Query

Intelligent Query opens up a whole new way of looking at your data in CRM when building ad-hoc Advanced Find searches or Marketing Lists. However, there are some considerations you need to make when using the solution. The way Intelligent Query works is by registering custom code on the "Retrieve Multiple" message in CRM. By default, it will register the code on every entity in CRM. With custom code comes some degradation of CRM's performance because any time a call is made to retrieve entities in CRM the code must be run even if the particular query is not a "not in" or "does not have" query. Because of this you may want to only use Intelligent Query for certain entities in CRM and turn it off for others. Fortunately, Intelligent Query provides the ability to do exactly that. After you installed it you can configure the solution by going to the Configuration page.

**Intelligent Query Configuration**

By default, Intelligent Query (IQ) is registered to run on all entities as a retrieve multiple plugin. This means that all entities, even if they can't be shown in the Advanced Find, will execute the IQ plugin on retrieve. This configuration page allows you to enable the IQ plugin for all entities or select which Advanced Find eligible entities for which you want IQ registered.

Happy finding!

**Enable IQ For:**  
 All Entities  
 Some Entities  
 No Entities

<input type="checkbox"/>	Display Name	Name	Schema Name
<input type="checkbox"/>	Account	account	Account
<input type="checkbox"/>	Activity	activitypointer	ActivityPointer
<input type="checkbox"/>	Address	customeraddress	CustomerAddress
<input type="checkbox"/>	Anonymous Visitor	cdi_anonymousvisitor	cdi_anonymousvisitor
<input type="checkbox"/>	Answer	msdyn_answer	msdyn_answer
<input type="checkbox"/>	Appointment	appointment	Appointment
<input type="checkbox"/>	Article	kbarticle	KbArticle
<input type="checkbox"/>	Article Template	kbarticletemplate	KbArticleTemplate
<input type="checkbox"/>	Azure Deployment	msdyn_azuredeployment	msdyn_azuredeployment
<input type="checkbox"/>	Bookable Resource	bookableresource	BookableResource
<input type="checkbox"/>	Bookable Resource Booking	bookableresourcebooking	BookableResourceBooking
<input type="checkbox"/>	Bookable Resource Booking Header	bookableresourcebookingheader	BookableResourceBookingHeader
<input type="checkbox"/>	Bookable Resource Category	bookableresourcecategory	BookableResourceCategory
<input type="checkbox"/>	Bookable Resource Category Assn	bookableresourcecategoryassn	BookableResourceCategoryAssn
<input type="checkbox"/>	Bookable Resource Characteristic	bookableresourcecharacteristic	BookableResourceCharacteristic
<input type="checkbox"/>	Bookable Resource Group	bookableresourcegroup	BookableResourceGroup
<input type="checkbox"/>	Booking Status	bookingstatus	BookingStatus
<input type="checkbox"/>	Bulk Text Message	cdi_bulktxtmessage	cdi_bulktxtmessage
<input type="checkbox"/>	Campaign	campaign	Campaign
<input type="checkbox"/>	Campaign Activity	campaignactivity	CampaignActivity
<input type="checkbox"/>	Campaign Automation	cdi_automation	cdi_automation
<input type="checkbox"/>	Campaign Response	campaignresponse	CampaignResponse

From here you can use the default All Entities option or specify which entities you want to use the solution with. For example, you may decide that your organization doesn't have enough products to warrant needing the overhead of the Intelligent Query plugin running every time someone searches for products. By deselecting the product entity and clicking Save you can control with which entities you want IQ to work.

Cobalt Intelligent Query takes advantage of an otherwise useless query and turns it into something very useful. It does so by finally bringing "Not In" searches to users in the native CRM Advanced Find tool that they use every day. Like I said, it's free and it just plain works, so it's kind of no-brainer to start using it in your organization to empower your users.