

ACTIVE DIRECTORY WEB SERVICE USER GUIDE

LAST UPDATED: January 4, 2013

Introduction.....	1
Web methods	4
Active Directory query fails on Windows XP	27
Support	28

INTRODUCTION

The Active Directory Web Service is available in two of Qdabra's products:

- Qdabra's Database Accelerator (DBXL)
- Qdabra's Active Directory Standalone tool

For additional information on both these products visit the Products page on [Qdabra.com](http://www.qdabra.com).

LDAP

Using the Active Directory web service requires an LDAP connection string. If you are having trouble determining your connection string, please refer to [this document](#).

VERIFY INSTALLATION

You can quickly verify that the installation is working as expected by following these steps from the machine where the product is installed:

1. Open Internet Explorer.
2. Open the web service URL: <http://<servername>/<QdabraPath>/AdUserInfo.aspx>.
3. Click on **GetMyInfo**.
4. Click **Invoke**.

A new Internet Explorer window will open, showing the current user's Active Directory properties.



<http://www.qdabra.com>

Copyright © 2006-2010 Autonomy Systems, LLC. All rights reserved.

ACCESS PATH

This document aims to explain the web service, though we must note that the URL to access the web service will vary depending on the product you are using.

- If you are using DBXL, the access path to the web service will be <http://<servername>/QdabraWebService/ADUserInfo.aspx>.
- If you are using the standalone tool, the path will be <http://<servername>/QdabraAD/AdUserInfo.aspx>

You can navigate to this page in Internet Explorer to view the list of available methods.

AdUserInfo

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [FindUsersByAlias](#)
Attempts to find any users with an alias matching the specified search string.
- [FindUsersByName](#)
Attempts to find any users with a name matching the specified search string.
- [GetEmployeesForManager](#)
Returns a list of employees that have a manager with the specified alias.
- [GetGroupsForUser](#)
Returns a list of the groups to which the current user belongs.
- [GetManagerAlias](#)
Returns the alias for a user's manager if one is set.
- [GetMembersOfGroup](#)
Returns the alias, display name and email address of group members.
- [GetMyDomains](#)
Returns a list of the DBXL domains to which the current user belongs.
- [GetMyInfo](#)
Gives the Active Directory properties of the currently Logged In user.
- [GetUserInfo](#)
Gives the Active Directory properties of the specified user name.
- [IsUserMemberOfGroup](#)
Returns true if the user is part of the specified group, false otherwise.

For the purposes of the steps provided in this document, we'll refer to this path like this:

<http://<servername>/<QdabraPath>/ADUserInfo.aspx>

LIMIT IN NUMBER OF RESULTS



<http://www.qdabra.com>

Copyright © 2006-2010 Autonomy Systems, LLC. All rights reserved.

The Active Directory Web Service is limited, by default, to return only 200 items. This can be controlled in web.config; locate and edit the following line to increase the maximum number of results returned.

<adService path="LDAP://DC=AutonomySystems,DC=local" MaxResultItems="200">



WEB METHODS

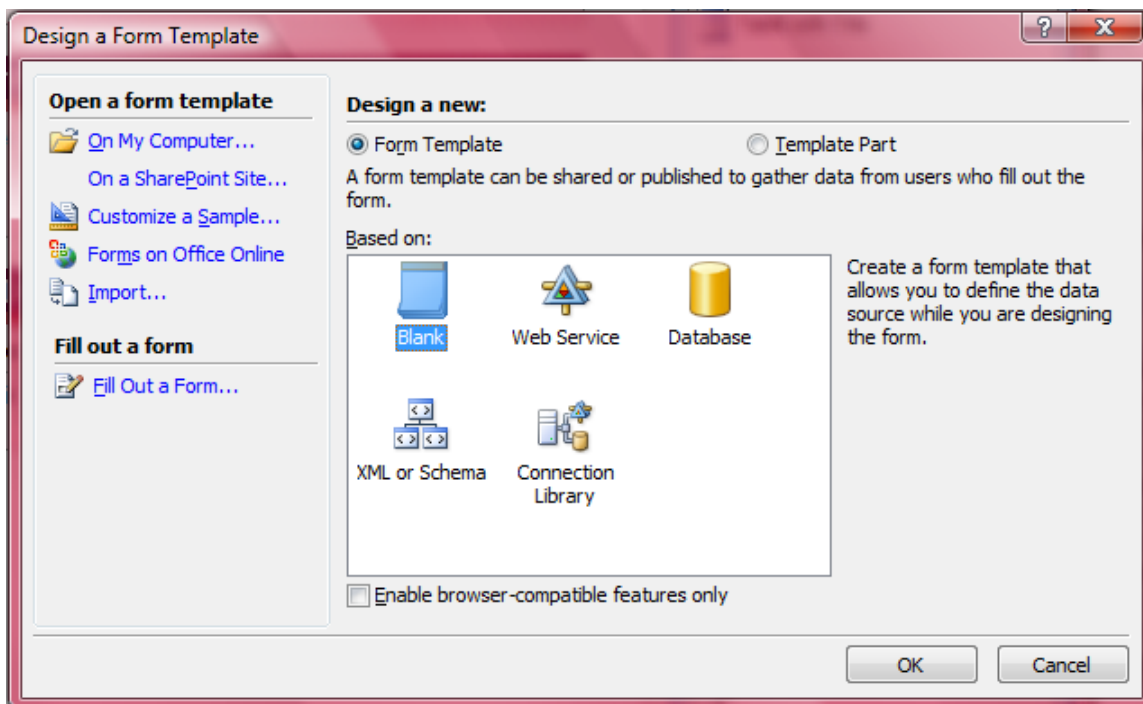
The Active Directory Web Service supports the following methods:

- **GetMyInfo** - gives the Active Directory properties of the currently Logged In user
- **GetUserInfo** - gives the Active Directory properties of the specified user name
- **GetManagerAlias** - returns the alias for a user's manager if one is set
- **GetEmployeesForManager** - returns a list of employees that have a manager with the specified alias
- **FindUsersByName** - attempts to find any users with a name matching the specified search string
- **FindUsersByAlias** - attempts to find any users with an alias matching the specified search string
- **IsUserMemberOfGroup** - returns true if the user is part of the specified group, false otherwise
- **GetGroupsForUser** - returns a list of the groups to which the user belongs. If no alias is provided, the method uses the currently logged in user.
- **GetMembersOfGroup** - returns the alias, and display name of group members.
- **GetAllUsers** - Returns the alias and display name of domain users

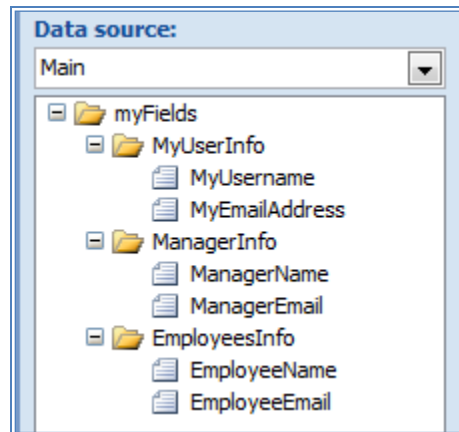
This section will show you how to set up a sample form which queries these operations.

DESIGN THE INFOPATH FORM TEMPLATE

1. Open InfoPath and design a new blank form.



2. Add the following groups and fields in your Main data source:



3. Add three tables with titles, and place each group's fields into each of the tables:

4. Add secondary data connections:
 - a. Go to **Tools** → **Data Connections** → **Add....**
 - b. Create a new connection to **receive data** from a **Web service**.



- ☒ Create a new connection to:
☐ Submit data
☒ Receive data

From where do you want to receive your data?

- ☐ XML document
☐ Database (Microsoft SQL Server or Microsoft Office Access only)
☒ Web service
☐ SharePoint library or list

- c. Enter the web service URL: <http://<servername>/<QdabraPath>/ADUserInfo.asmx>
- d. From the list of operations, select **GetGroupsForUser**.

Select an operation:

GetGroupsForUser
 GetMyDomains
 GetMembersOfGroup
 GetManagerAlias
 FindUsersByAlias
 FindUsersByName
 GetEmployeesForManager
 IsUserMemberOfGroup
 GetUserInfo
 GetMyInfo

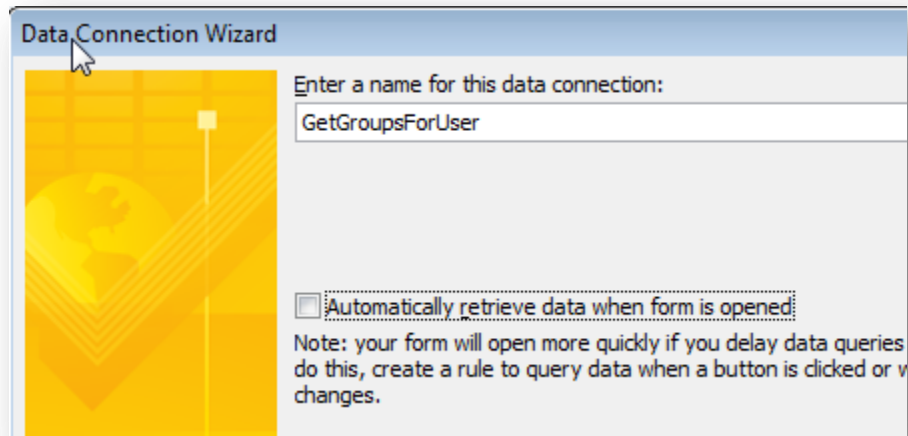
- e. Enter the default values:

Parameters:

Parameter	Type	Value
tns:alias	string	arunaa
tns:recursive*	boolean	false



- f. Uncheck the box that says **Automatically retrieve data when the form is opened**, and then click **Finish**.

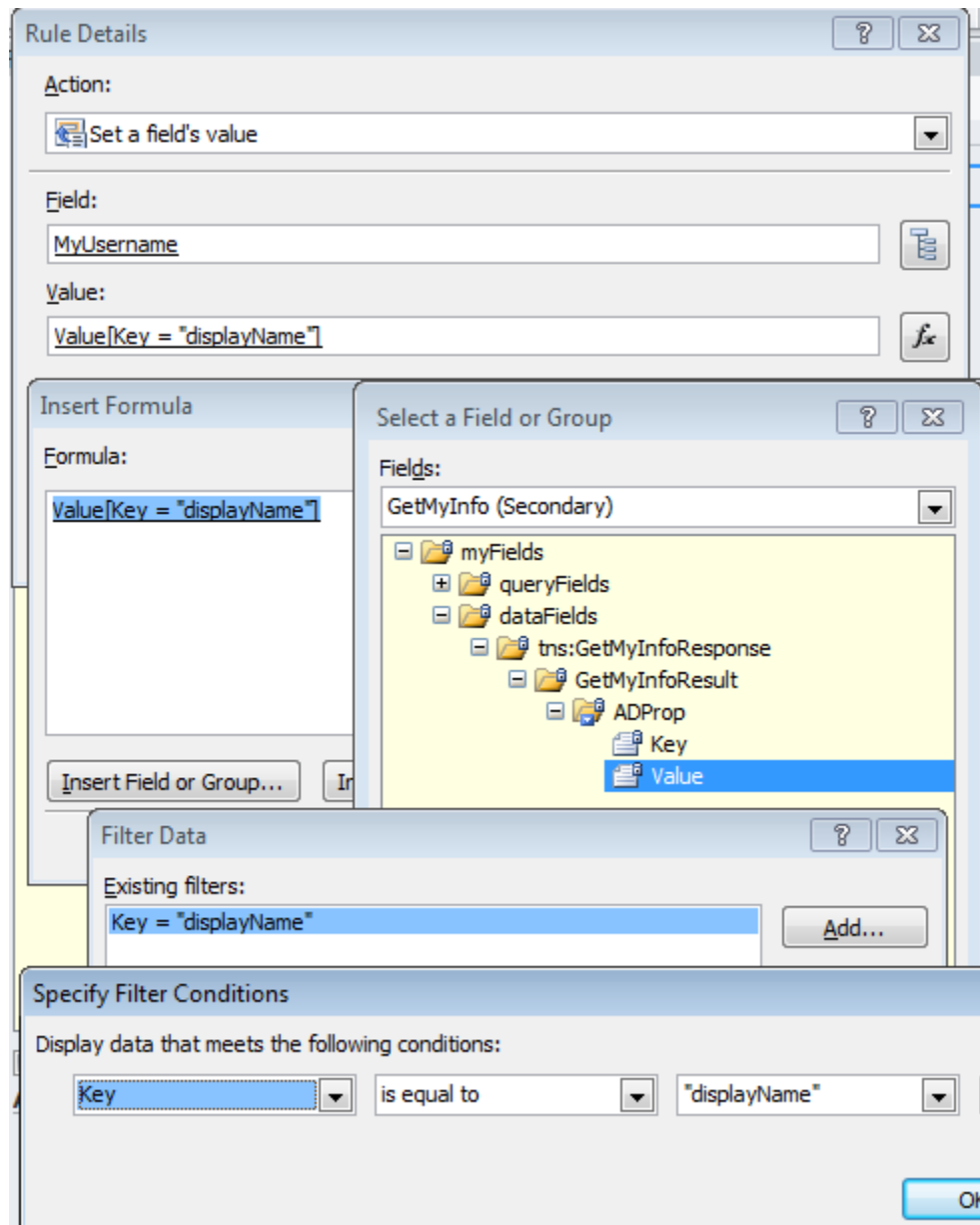


5. Repeat **Step 4** to add data connections for the other eight operations.
6. When done, exit the data connection wizard.

QUERY GETMYINFO

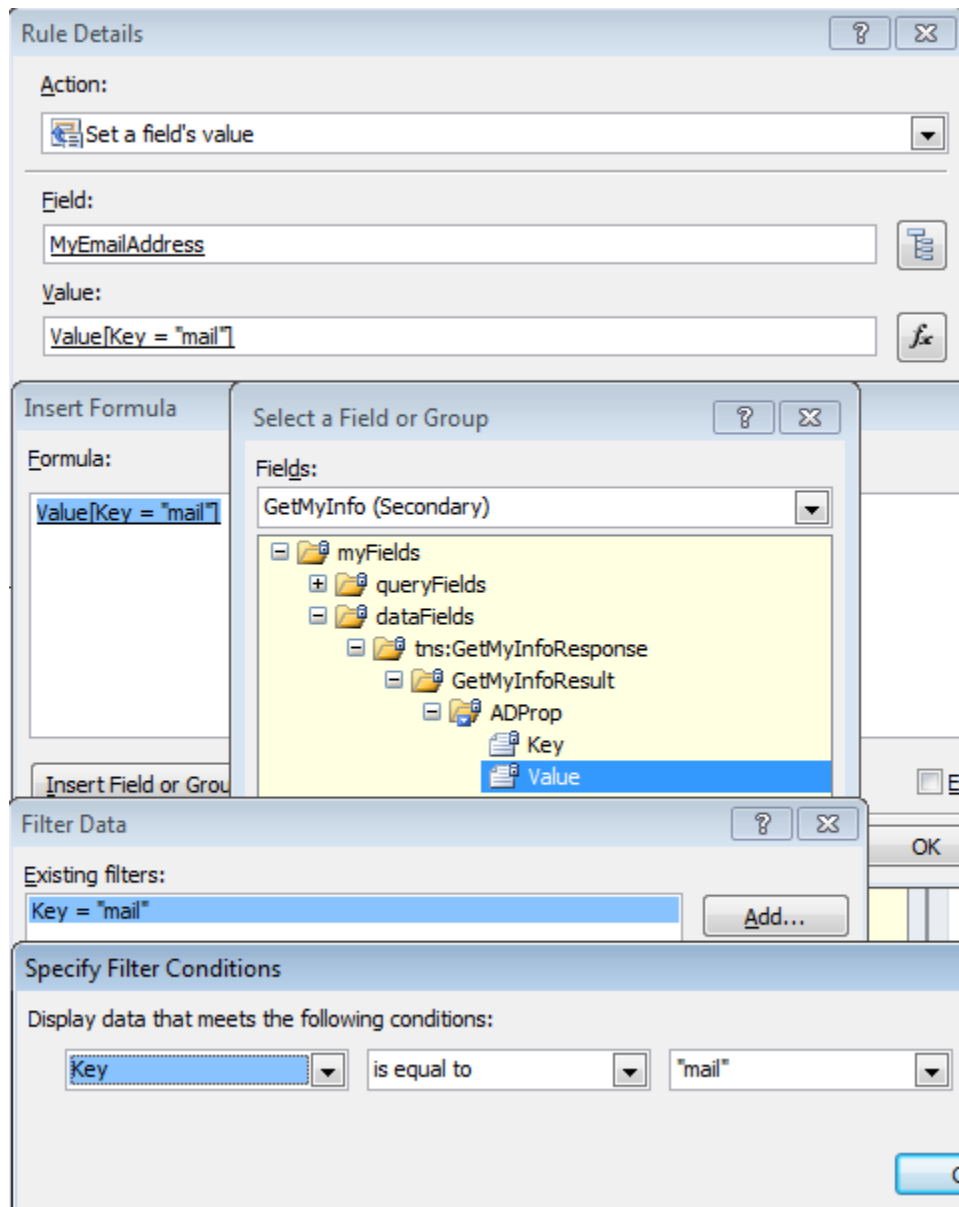
1. Add an on load rule that queries GetMyInfo.
2. Add a new action that sets the value of the MyUsername field to the displayName obtained from the GetMyInfo query. This is accomplished by adding a filter so that it only displays the **Value** when **Key** is equal to the text **displayName**.





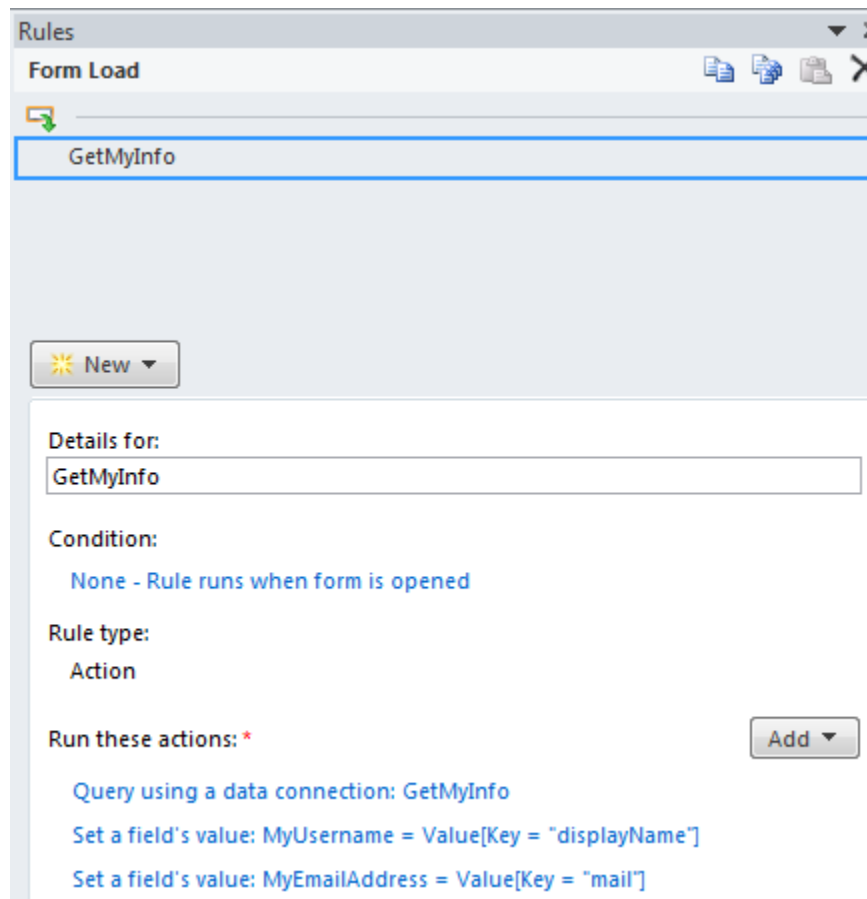
3. Add a new action that tests the value of **MyEmailAddress** field, as in step 2 above, except that **Value** should be filtered such that **Key** is equal to the text **mail**.





Your on load rules may look like this once you have added the three actions above:





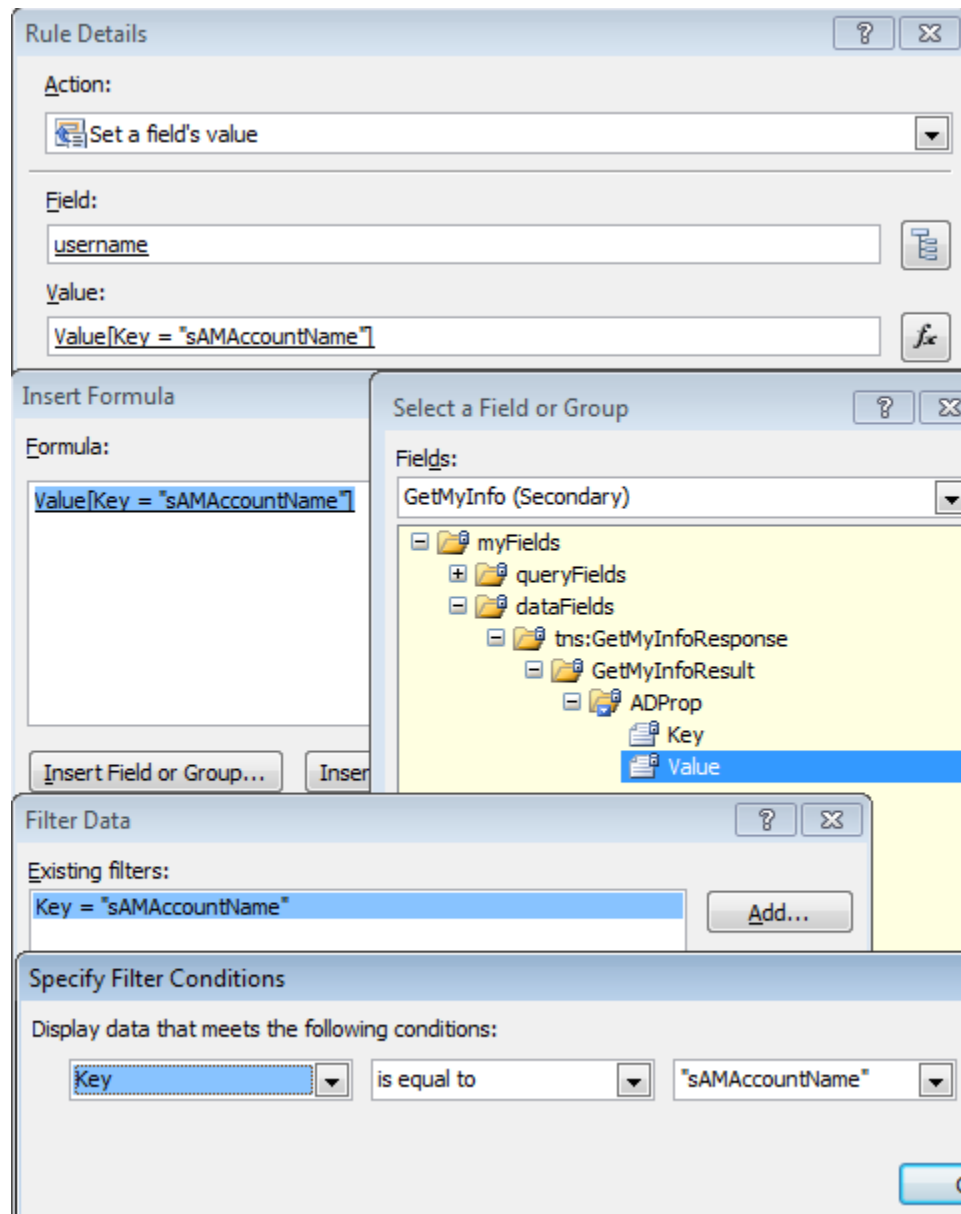
We are only using the displayName and mail keys in the steps above. To view a list of all the Active Directory keys, drag the whole **ADProp** node
 (/dfs:myFields/dfs:dataFields/tns:GetMyInfoResponse/tns:GetMyInfoResult/tns:ADProp) as a repeating table onto your form, and then preview.

4. Preview your form. Your Active Directory displayName and email address should be displayed in the **My Info** table.

QUERY GETMANAGERALIAS

5. Return to the on load rules and add a new action that sets the value of the GetManagerAlias username field (/dfs:myFields/dfs:queryFields/tns:GetManagerAlias/tns:username) to another value obtained from the GetMyInfo query. This time, instead of filtering for displayName or email, filter for the "sAMAccountName" Key.





6. Add a new action that queries the GetManagerAlias data connection.

Though we have queried for the manager's alias, we need an additional data connection query to obtain additional information about the manager.

QUERY GETUSERINFO



7. Add a new action that sets the value of the GetUserInfo username field (/dfs:myFields/dfs:queryFields/tns:GetUserInfo/tns:username) to the result of the GetManagerAlias connection (/dfs:myFields/dfs:dataFields/tns:GetManagerAliasResponse/tns:GetManagerAliasResult).
8. Add a new action that queries the GetUserInfo data connection.

We are using the manager's alias (from the GetManagerAlias data connection result) to call GetUserInfo.

9. Add an action that sets the value of the ManagerName node. The value is obtained from the GetUserInfo data connection, by filtering the ADProp table in the same way done in steps 2 and 3. To obtain the name, use the "Key" = "name".
10. Add an action that sets the value of the ManagerEmail node. Use the "mail" key as in step 3.

Your on load rules will look like this:

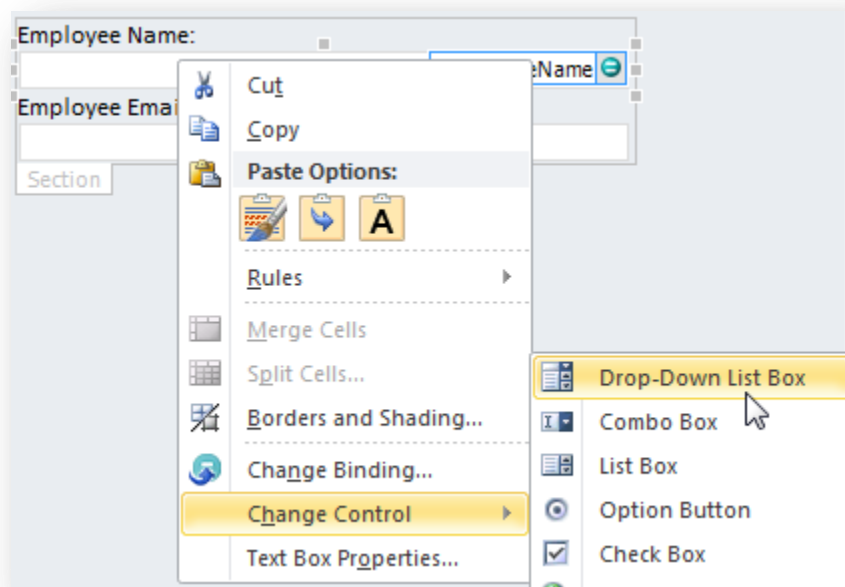
```
Query using a data connection: GetMyInfo
Set a field's value: MyUsername = Value[Key = "displayName"]
Set a field's value: MyEmailAddress = Value[Key = "mail"]
Set a field's value: username = Value[Key = "sAMAccountName"]
Query using a data connection: GetManagerAlias
Set a field's value: username = GetManagerAliasResult
Query using a data connection: GetUserInfo
Set a field's value: ManagerName = Value[Key = "name"]
Set a field's value: ManagerEmail = Value[Key = "mail"]
```

Preview the form to check your progress. So far, you'll see your own name and email, as well as your manager's name and email populated into the form fields.

QUERY GETEMPLOYEESFORMANAGER

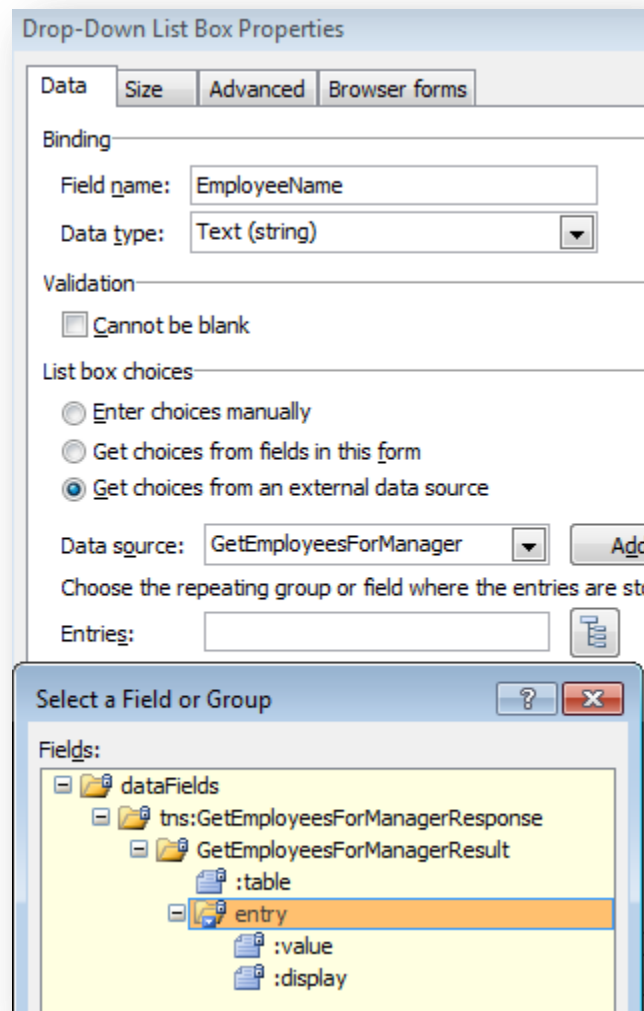
11. Add a new action that sets the value of the managerAlias node in the GetEmployeesForManager secondary data connection (/dfs:myFields/dfs:queryFields/tns:GetEmployeesForManager/tns:managerAlias) to the result obtained by the GetManagerAlias data connection (/dfs:myFields/dfs:dataFields/tns:GetManagerAliasResponse/tns:GetManagerAliasResult).
12. Add a new action that queries the GetEmployeesForManager data connection.
13. Change the EmployeeName textbox into a drop-down list box.





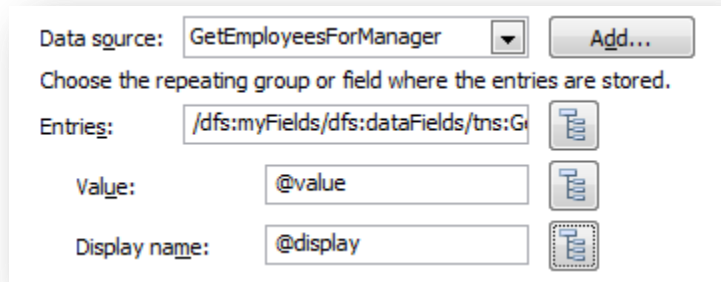
14. Open the dropdown properties for **EmployeeName**.
15. In the **Data** tab, under **List box entries**, select **Look up values from an external data source**, and choose **GetEmployeesForManager** from the Data source drop-down.
16. Click on the **Select XPath** button  for **Entries**, and expand the **dataFields** node to get to the **entry** repeating node.





17. Click on the **Select XPath** button for **Display name**, and select the display node. Your **EmployeeName** dropdown should be similar to this:



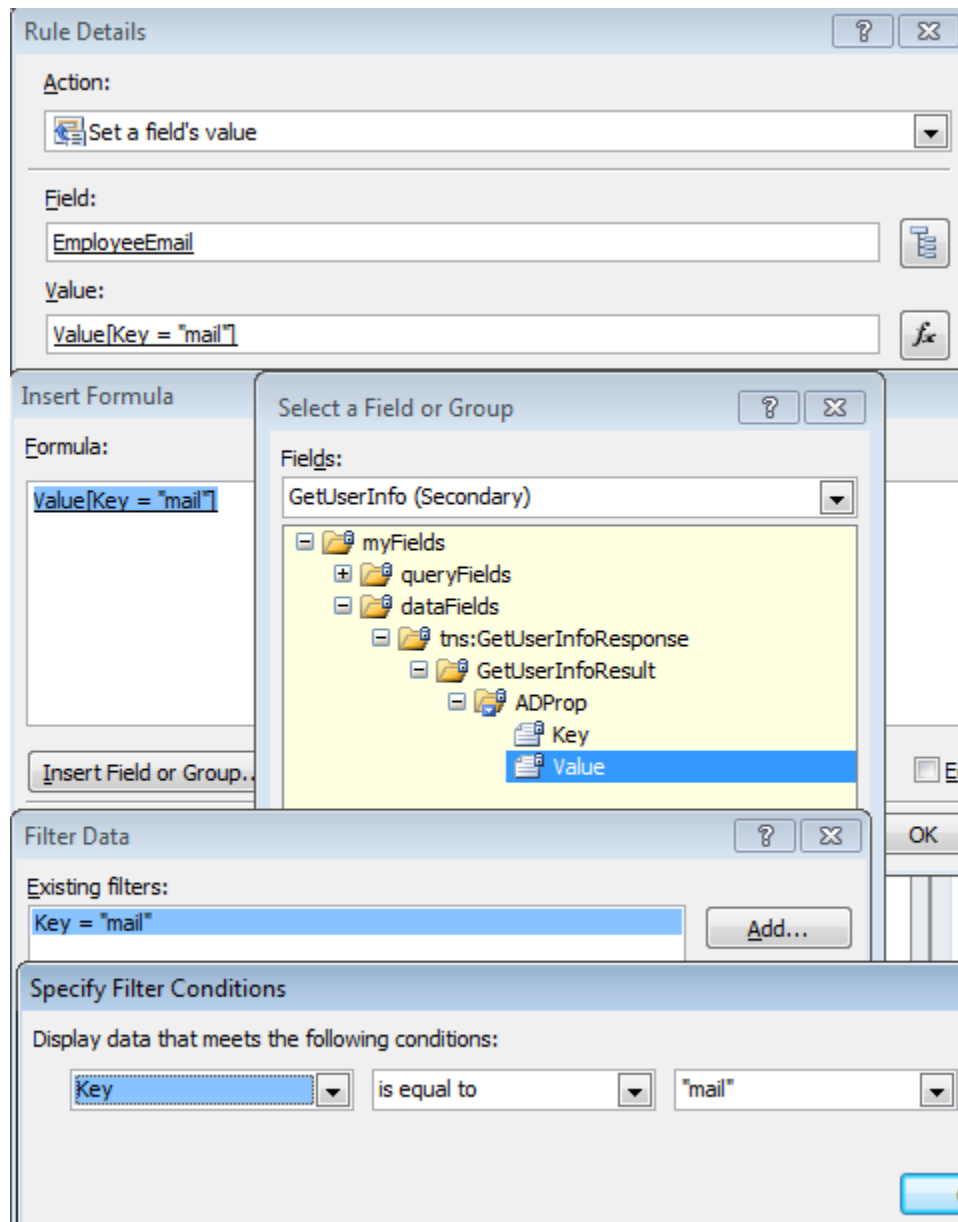


18. Add a rule to the EmployeeName dropdown.
19. Add a condition such that the rule only executes when the EmployeeName dropdown is not blank.

We will re-use the GetUserInfo method to retrieve the employee's email address.

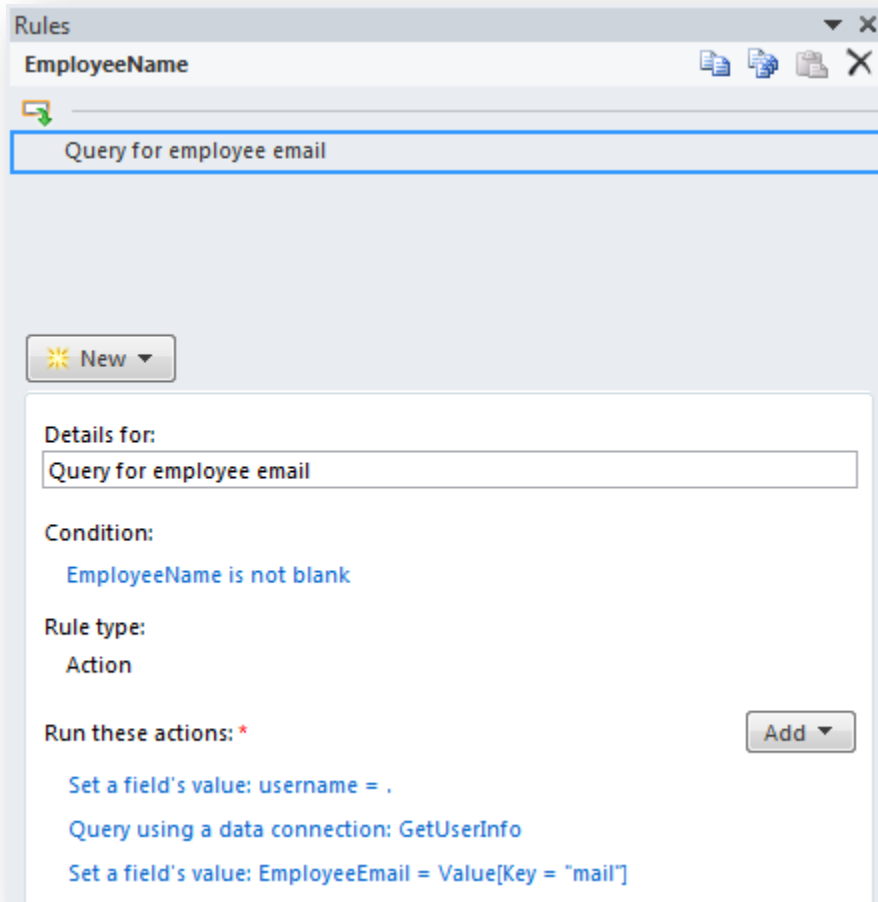
20. Create an action that sets the username field in the GetUserInfo secondary data connection (/dfs:myFields/dfs:queryFields/tns:GetUserInfo/tns:username) to the EmployeeName dropdown.
21. Add an action to call the GetUserInfo data connection.
22. Add an action that sets the value of the EmployeeEmail field by filtering the result of the GetUserInfo web service result, as we have done before.





The rules on the dropdown should look like this:





Rules

EmployeeName

Query for employee email

New

Details for:
Query for employee email

Condition:
EmployeeName is not blank

Rule type:
Action

Run these actions: *

Add

Set a field's value: username = .

Query using a data connection: GetUserInfo

Set a field's value: EmployeeEmail = Value[Key = "mail"]

Preview the form, and then select an Employee from the dropdown. All the fields in the form will be populated with the users' information.



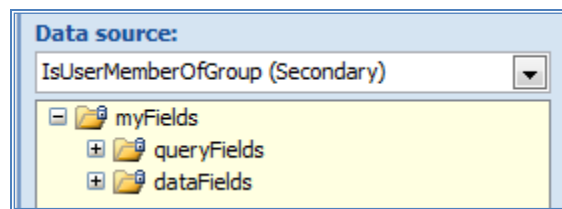
My Username:	Ernesto Machado (Test)
My Email Address:	Ernesto.Machado@qdabra.com
Manager Name:	Patrick Halstead
Manager Email:	Patrick.Halstead@Qdabra.com
Employee Name:	Tomoko Jones ▼
Employee Email:	Tomoko.Jones@Qdabra.com

QUERY ISUSERMEMBEROFGROUP

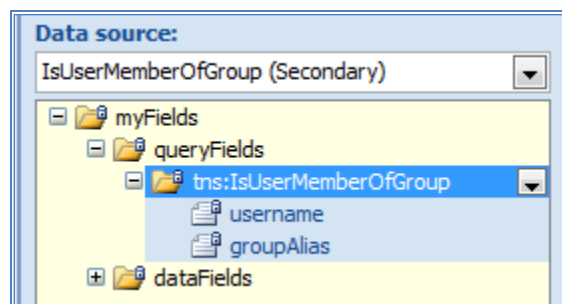
The **IsUserMemberOfGroup** method returns true if the user is part of the specified group, false otherwise.

To test this method in our sample form, follow these steps:

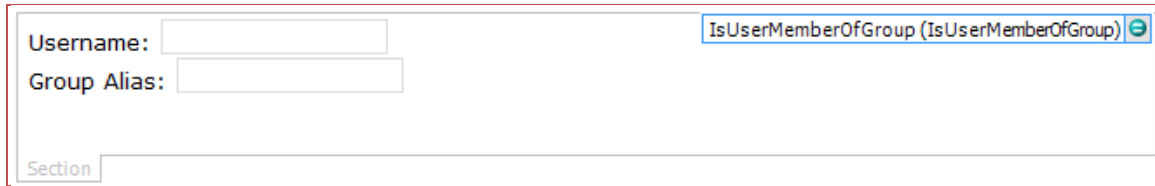
1. On the **Data source** task pane, select the **IsUserMemberOfGroup** secondary data source.



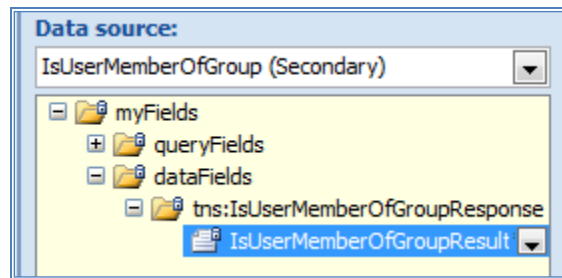
2. Expand the **queryFields** node to select **tns:IsUserMemberOfGroup**.



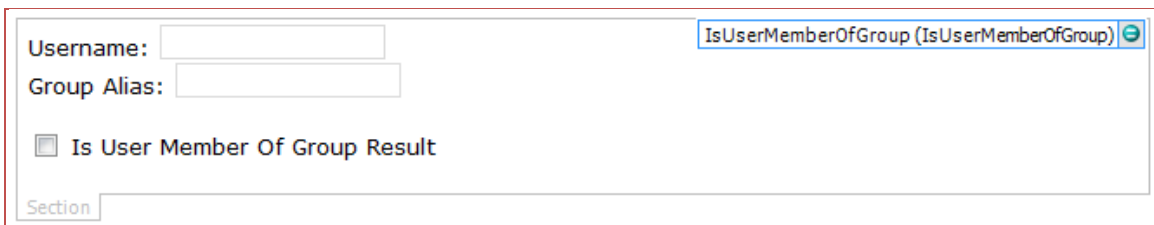
3. Drag this node as a **Section with controls** onto your form's view.



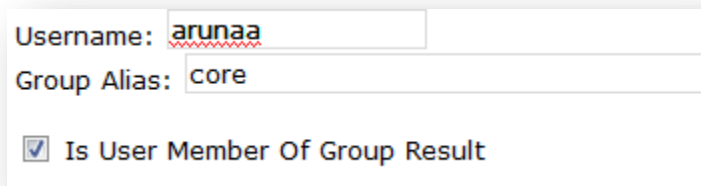
4. Expand the **dataFields** node to get to the **IsUserMemberOfGroupResult** node.



5. Drag this node inside the **IsUserMemberOfGroup** section. By default, it uses a check box control to indicate if the query returns true or false.



6. Add a button with a rule that queries using the **IsUserMemberOfGroup** data connection.
7. Preview your form. Specify any value in the **Username** and **Group Alias** parameters. Clicking the button will result to either true (box checked) or false (box unchecked).



QUERY FINDUSERSBYNAME

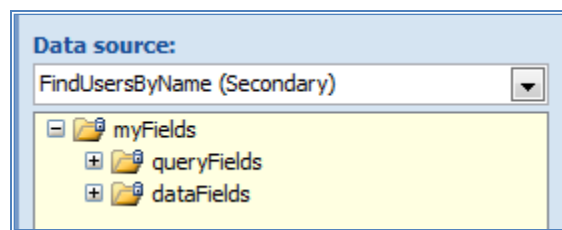


The **FindUsersByName** method attempts to find any users with a **name** matching the specified search string. When searching for a user by name, you can select from the following search methods:

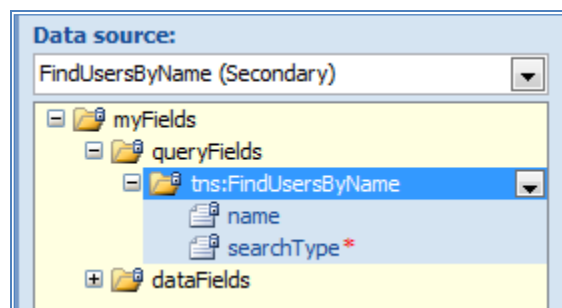
- *StartsWith*
- *EndsWith*
- *Contains*
- *Exact*

To test this method in our sample form, follow these steps:

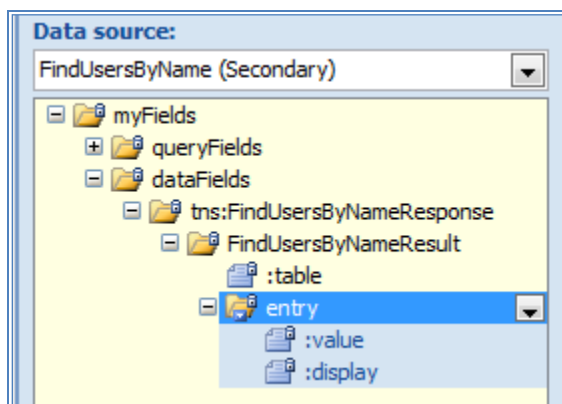
1. On the **Data source** task pane, select the **FindUsersByName** secondary data source.



2. Expand the **queryFields** node to select **tns:FindUsersByName**.



3. Drag this node as a **Section with controls** onto your form's view.
4. Expand the **dataFields** node to get to the **entry** repeating node.



5. Drag this node as a **Repeating table** inside the **FindUsersByName** section.



Name:

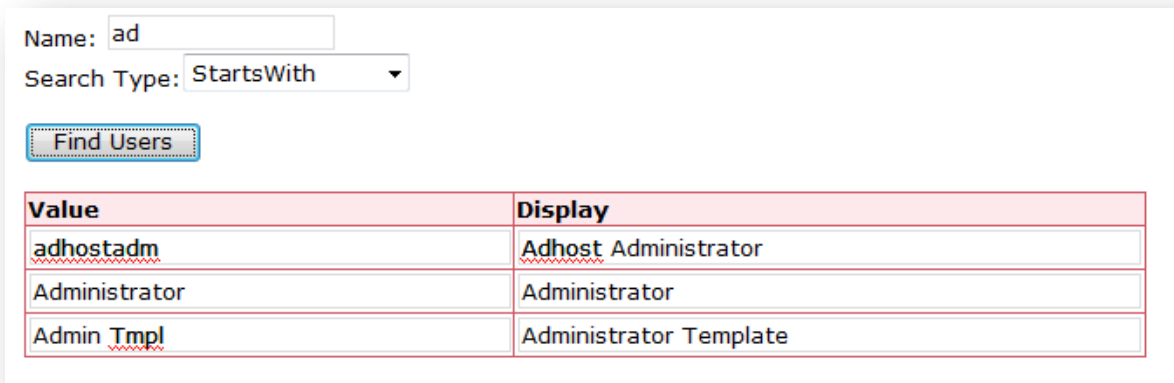
Search Type: Select...

Value	Display

☒ Repeating Table

Section:

6. Add a button below the **searchType** drop-down, and double-click to see its properties.
7. Label the button as **Find Users**, then add a rule that queries using the **FindUsersByName** data connection.
8. Preview your form once again. Enter a value in the **Name** textbox and select any **Search Type**, and then click **Find Users**. The repeating table will populate with a list of names that match the specified parameters. The **Value** column shows the user's alias, while the **Display** column shows the user's display name.



Name:

Search Type: StartsWith

Value	Display
adhostadm	Adhost Administrator
Administrator	Administrator
Admin Tmpl	Administrator Template

QUERY FINDUSERSBYALIAS

FindUsersByAlias does the same thing as **FindUsersByName**, except that it performs the search through the user's alias. Repeat all of the steps above, using the **FindUsersByAlias**' nodes and parameters.



Alias:

Search Type:

Value	Display
arunaa	Aruna Akella
cc-arunaa	Aruna Akella

QUERY THE GETGROUPSFORUSER METHOD

The **GetGroupsForUser** method returns a list of the groups to which the current user belongs. We will start by adding a data connection into a new blank InfoPath Filler form that queries this method.

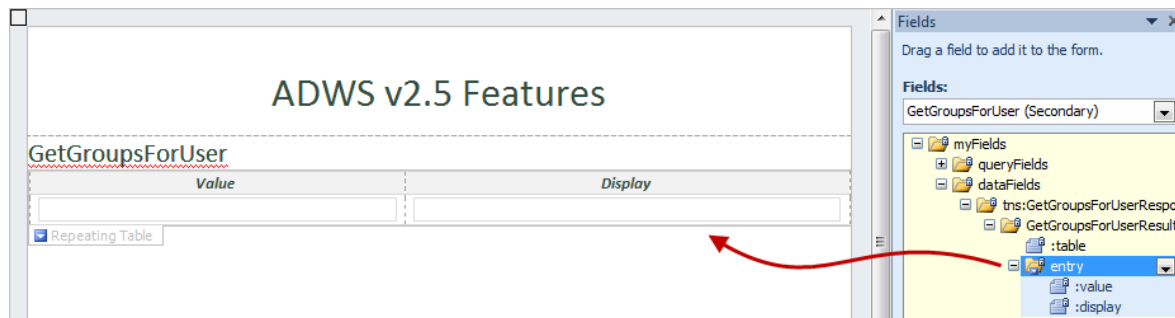
1. Design a new InfoPath Filler form.
2. Go to *Data > Data Connections > Add*.
3. Create a new connection to receive data from a SOAP Web service.
4. Enter the URL to the Qdabra Active Directory web service which by default is <http://<servername>/QdabraWebService/AdUserInfo.aspx>, replacing <servername> with the name of the machine where DBXL is installed.

Enter the location of the SOAP Web service definition:

5. Select **GetGroupsForUser** from the list of operations.
6. Click **Next** three times, leaving the defaults including allowing data to automatically retrieve on form open.
7. Click **Finish**, and then click **Close**.

To see the results of your query, drag the **entry** node as a repeating table onto your form, like shown:



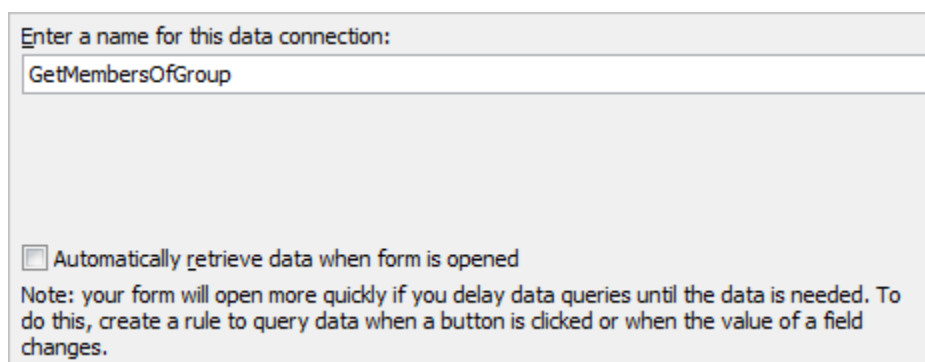


Preview your form and verify that the table auto-populates with the list of groups where you are a member of.

QUERY THE GETMEMBERSOFGROUP METHOD

The **GetMembersOfGroup** method returns the alias, display name and email address of group members. We will add another data connection that calls this method.

8. Go to *Data > Data Connections > Add*.
9. Create a new connection to receive data from a SOAP Web service.
10. Enter the same Active Directory web service URL that you specified earlier (default is <http://<servername>/QdabraWebService/AdUserInfo.asmx>, replacing <servername> with the name of the machine where DBXL is installed).
11. Select **GetMembersOfGroup** from the list of operations.
12. Click **Next** three times, this time deselecting the box 'Automatically retrieve data when form is opened' because we will need to specify the group before we actually call the web service.



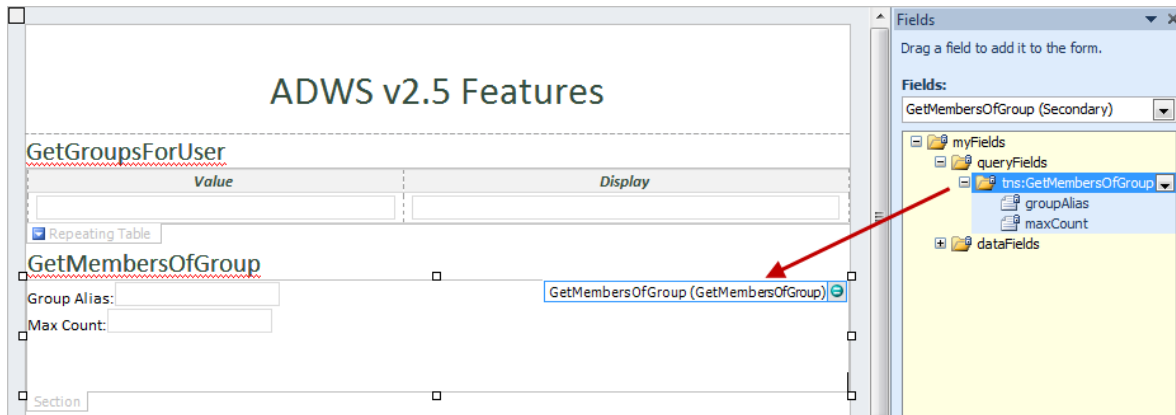
13. Click **Finish**, and then click **Close**.

We will display the query fields on our form so that we can specify which group to query and (optional) set the maximum number of members we wish to return.

14. From the *Fields* taskpane, switch to the **GetMembersOfGroup** data source.

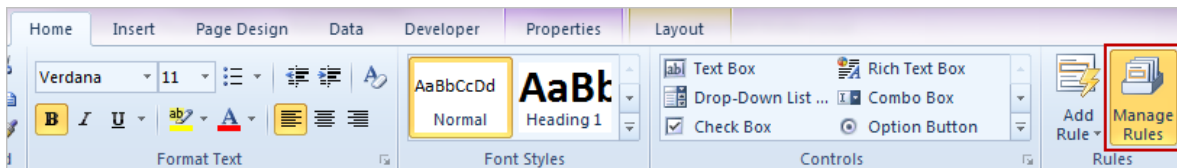


15. Drag the **tns:GetMembersOfGroup** node onto your form, like shown:

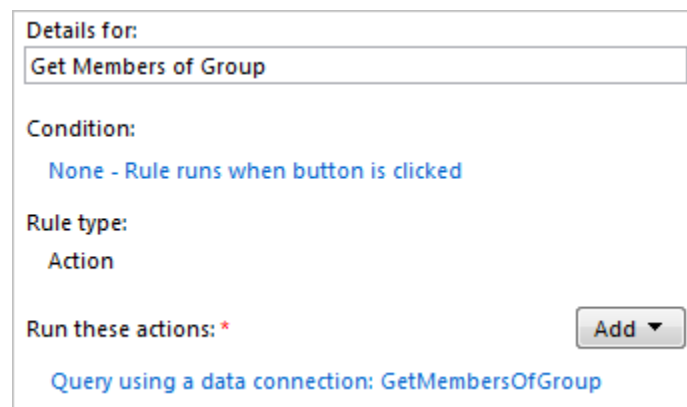


Next, we will add a button with a rule that queries the data connection.

16. From the *Controls* menu, select the **Button** control – you may place this inside the section you just dragged and then label it as 'Get Members of Group'.
17. From the *Home* tab > *Rules* section, click **Manage Rules** to display the *Rules* taskpane.



18. Add a new rule by clicking **New > Action > Add > Query for data** and then selecting the *GetMembersOfGroup* data connection.
19. Give your rule a descriptive name.



To see the results of your query, drag the corresponding **entry** node as a repeating table onto your form.



You may test the web service in Preview mode:

20. Select one group alias from the returned list of groups in the **GetGroupsForUser** data connection – enter this in the *Group Alias* text box. You may also specify the *Max Count* value which is the maximum number of members you wish to display.
21. Click on the 'Get Members of Group' button and verify that the table populates with the members of the group you specified.

GetMembersOfGroup

Group Alias:

Max Count:

Value	Display
<u>jo-anne</u>	<u>Jo-Anne Alfafara</u>
<u>MBalsamo</u>	<u>Mel Balsamo</u>
<u>cynthia</u>	<u>Cynthia Eng-Dinsel</u>
<u>ErnestoM</u>	<u>Ernesto Machado (Test)</u>

QUERY THE GETALLUSERS METHOD

The **GetAllUsers** method returns all users in the domain. This method has three parameters (all of which can be left blank), which allow you to narrow down the results:

- **OU:** You can enter an OU (organizational unit) such as OU=SBSUsers,OU=Users,OU=MyBusiness to filter results.
- **Filter:** Allows you to create filters to obtain a list of active directory users. Here are some examples of filters:
 - First name beginning with 'J' **givenName=J***
 - Users with an email address **mail=***
 - Users with a phone number starting with (425) or (206)
 [(telephoneNumber=\28425\29*)(telephoneNumber=\28206\29*)]
 - Users with an email and department **(mail=*)(department=*)**
 - Users without an email account in AD **!(mail=*)**

Characters that must be escaped in the filter:

*	\2A
(	\28
)	\29
\	\5C



NULL	\00
------	-----

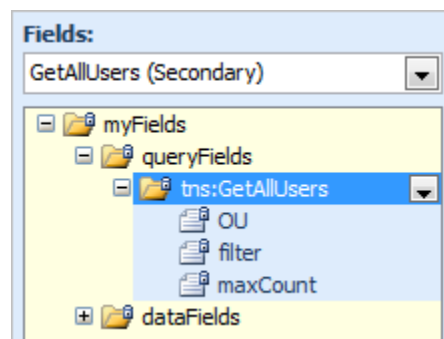
- **Max Count:** This parameter will limit the number of results returned by the method. This maximum is upper-limited by the limit in web.config explained earlier in this document.

For this method, we will repeat the same technique we used earlier. First, we will add a data connection that calls this method.

22. Go to *Data > Data Connections > Add*.
23. Create a new connection to receive data from a SOAP Web service.
24. Enter the same Active Directory web service URL that you specified earlier (default is <http://<servername>/QdabraWebService/AdUserInfo.asmx>, replacing <servername> with the name of the machine where DBXL is installed).
25. Select **GetAllUsers** from the list of operations.
26. Click **Next** three times and deselect the box 'Automatically retrieve data when form is opened' so that we can specify the query filter before we actually call the web service.
27. Click **Finish**, and then click **Close**.

Next, we will display the query fields on our form so that we can try out a few filters for the query and (optional) set the *OU* and the *Max Count* fields.

28. From the *Fields* taskpane, switch to the **GetAllUsers** data source.
29. Drag the **tns: GetAllUsers** node onto your form.



Heare again, we will add a button with a rule that queries the data connection.

30. Select the **Button** control from the *Controls* menu and label it as 'Get All Users'.
31. In the *Rules* taskpane, add a new rule by clicking **New > Action > Add > Query for data** and then selecting the *GetAllUsers* data connection.
32. Give your rule a descriptive name.

Let's display the result of our query by dragging the corresponding **entry** node as a repeating table onto your form.



Test the web service in Preview mode:

33. You may start by clicking on the 'Get All Users' button right away; verify that the table populates with all the domain members.
34. You can then narrow down your results by using any of the examples above for your *Filter* query, like shown:

GetAllUsers

OU:

Filter:

Max Count:

Value	Display
<u>jo-anne</u>	Jo-Anne <u>Alfajara</u>
<u>jimcan</u>	Jim Cantwell
<u>jimcan-admin</u>	Jim Cantwell (admin)
Janice	Janice Kutzera
<u>irishe</u>	James <u>Rishe</u>
<u>jwalts</u>	Jennifer Walts

ACTIVE DIRECTORY QUERY FAILS ON WINDOWS XP

If you install DBXL on Windows XP and you are receiving an error when querying the Active Directory Web service (ADUserInfo.asmx), you will have to manually configure your machine's ASPNET account. To verify that this is the problem, navigate to <http://<server name>/QdabraWebService/ADUserInfo.asmx>, click **GetMyInfo** and **Invoke**. If you don't get XML you will need to follow the following configuration steps.

1. Navigate to <http://support.microsoft.com/kb/329290/>, download and install the AspNet_setreg.exe package.
2. Open a command window by clicking on **Start**, selecting **Run**, and then typing **cmd**.
3. Run `aspnet_setreg.exe -k:SOFTWARE\Qdabra\DBXL\identity -u:"yourdomainname\username" -p:"password"` with the correct values for yourdomainname\username and password.
4. Click **Start**, then **Run** and enter **regedt32**.
5. Expand **HKEY_LOCAL_MACHINE\Software\Qdabra\DBXL\identity\ASPNET_SETREG**. In the left pane, right click on the **ASPNET_SETREG** key, and select **Permissions**.
6. Add permissions to allow **MACHINE\ASPNET** to read the key (where MACHINE is the name of your machine).
7. Click **Start**, then **Run** and type in **InetMgr**.



8. Expand **Web Sites > Default Web Site**.
9. Right click on **QdabraWebService** and choose **Open**.
10. In the window that opens, right click **web.config** and choose **Open With**, then select **Notepad**.
11. Search for `<identity impersonate="false" />` and replace it with the following:

```
<identity impersonate="true"
userName="registry:HKEY_LOCAL_MACHINE\SOFTWARE\Qdabra\DBXL\identity\ASPNET_SETREG,userName"
password="registry:HKEY_LOCAL_MACHINE\SOFTWARE\Qdabra\DBXL\identity\ASPNET_SETREG,password" />
```

Note that there must be no spaces in the value for username and password.

12. Find the key called `ActiveDirectoryPath` and change its value from the default (`LDAP://DC=domain`) to the value that corresponds to your server. If you do not know the correct value, ask your administrator or [use the LDAP Test Tool](#).
13. Save and close `web.config`.

SUPPORT

If you have questions about the information in this document, please contact Qdabra Software for assistance.

Licensed customers can contact us via Support@Qdabra.com.

You can also use the [InfoPathDev.com](http://www.infopathdev.com) [Qdabra Product support forums](#) to request help from the community.

CONTACT INFORMATION

Qdabra Software
218 Main Street, Suite 731, Kirkland, WA 98033

Phone:	877.544.2389
Email:	Support@Qdabra.com
Website:	http://www.qdabra.com
Community:	http://www.infopathdev.com



<http://www.qdabra.com>

Copyright © 2006-2010 Autonomy Systems, LLC. All rights reserved.