



Send email to a user selected via People Picker using Active Directory

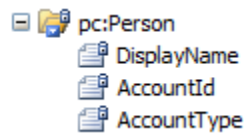
The People Picker allows us to select users from the company directory:

But the People Picker control in InfoPath 2010 has two important limitations. First, it does not query for the user's email:

This is what's stored in the main data source:

Display Name	Account Id	Account Type
Ernesto Machado (Test)	AUTONOMYSYSTEMS\ErnestoM	User

Second, its schema is locked.



This means that we cannot add an additional node (to store the email), nor rules.

This is problematic for sending emails: a data connection which submits the form via email requires an email address for the recipient:



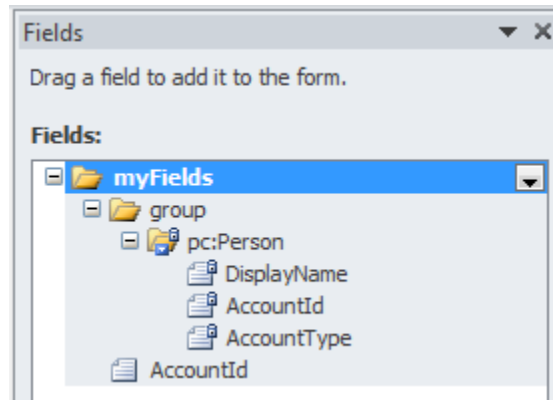
<http://www.qdabra.com>

Last updated on 8/3/2012 11:15 AM

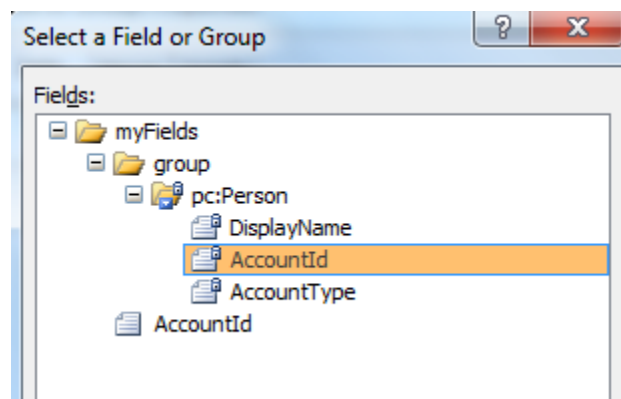
Copyright © 2006-2011 Autonomy Systems, LLC. All rights reserved.

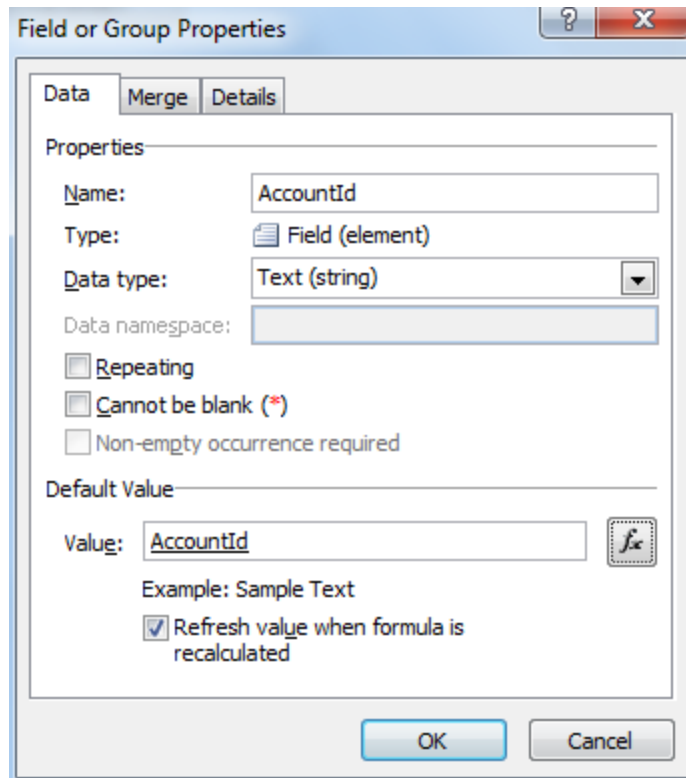
This document will explain a workaround to this scenario.

1. First, add a field to store the user's alias. People Picker calls this the AccountId, but you may name it something else if desired. You'll end up with that additional field, OUTSIDE of the People Picker group:



2. Right click on the new field and select Properties.
3. For its Default Value, select the AccountId field located within the pc:Person group.





The 'Field or Group Properties' dialog box is shown with the 'Data' tab selected. The 'Properties' section contains the following fields and options:

- Name:** AccountId
- Type:** Field (element)
- Data type:** Text (string)
- Data namespace:** (empty)
- ☐ Repeating
- ☐ Cannot be blank (*)
- ☐ Non-empty occurrence required

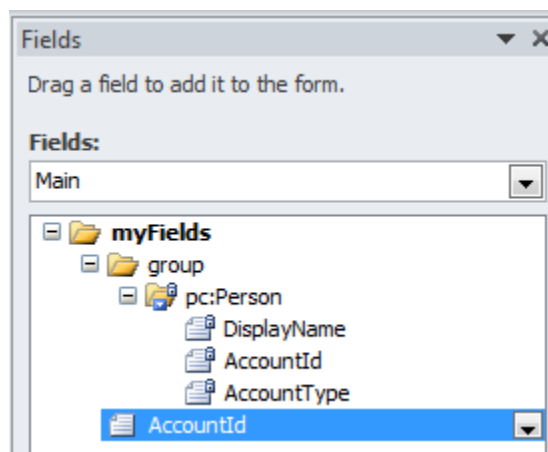
The 'Default Value' section contains:

- Value:** AccountId
- Example:** Sample Text
- ☒ Refresh value when formula is recalculated

Buttons at the bottom: OK, Cancel.

Now that the user's alias is stored outside the People Picker, we can create rules to obtain the email address.

4. Create a data connection to <http://<servername>/qdabrawebservice/aduserinfo.asmx>. To do this go to Data > Data Connections > Add > Receive Data > Next > SOAP Web Service > Next.
5. Enter the URL to the webservice and click Next.
6. Select GetUserInfo and click Next three times.
7. In the final screen, uncheck the option to retrieve data on load. Click Finish.
8. Click Close in the Data Connections dialog.
9. Select the node we added to the schema (to store the user's alias) and select Home > Manage Rules.



The 'Fields' dialog box is shown with the 'Main' dropdown menu. The 'Fields:' section contains a tree view of the schema:

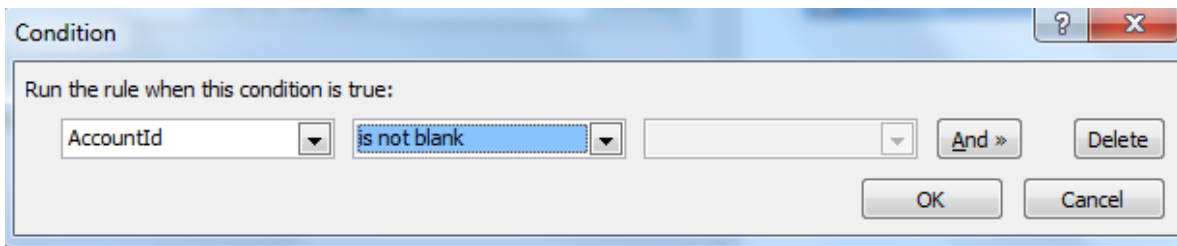
- myFields
 - group
 - pc:Person
 - DisplayName
 - AccountId
 - AccountType

The 'AccountId' field is selected in the list at the bottom.

10. Click New > Action.

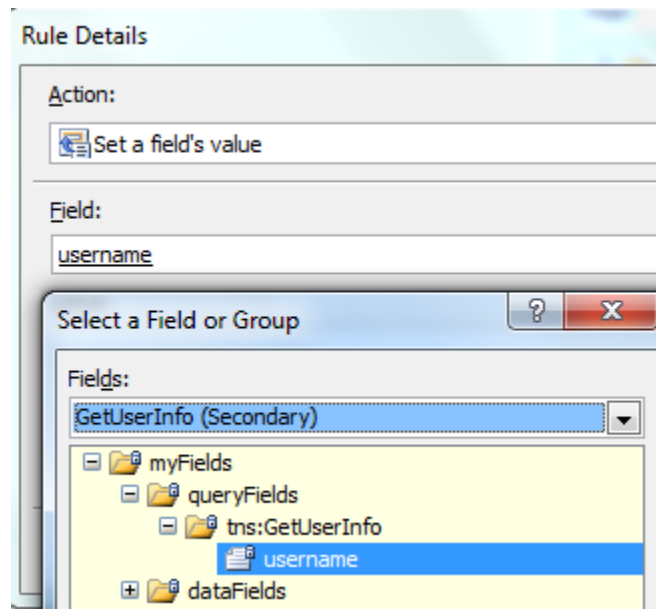


11. The condition for this rule will be AccountId is not blank:



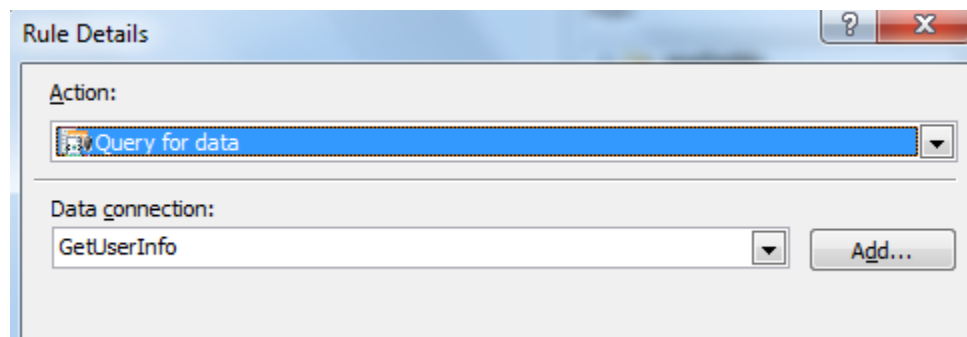
12. Click Add > Set a field's value.

13. For the Field, select the username node in the secondary data connection:



14. For Value, select the AccountId field itself.

15. Click Add > Query for data to add an action that queries the data connection we created earlier:



If you display all these fields in your canvas, you will see the full sequence of events, as follows:

- The user makes a selection in People Picker



AUTONOMYSYSTEMS\ErnestoM  

- People Picker returns data for this user:

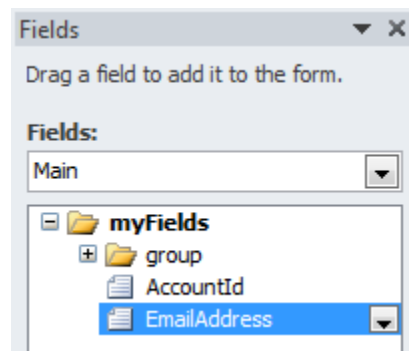
Display Name	Account Id	Account Type
AUTONOMYSYSTEMS\ErnestoM	AUTONOMYSYSTEMS\ernestom	User

- This sets the AccountId field:

Account Id: AUTONOMYSYSTEMS\ernestom

- Which then kicks off the query to retrieve the AD properties for that (including email).

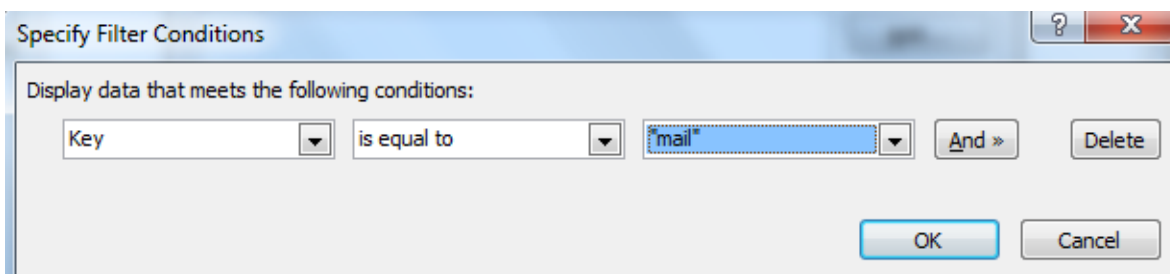
16. Add a new field to your schema to store the email address.



17. Add one more rule to the AccountId field by clicking Add > Set a field's value.

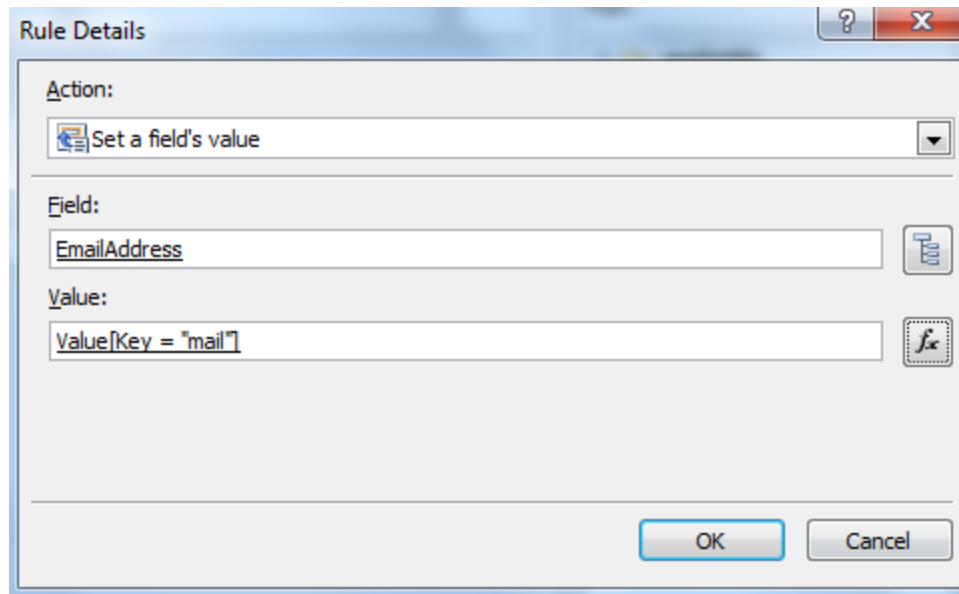
18. For Field, select the EmailAddress node created above.

19. For Value, select the Value node in the GetUserInfo data connection, but click Filter and specify the following filter:



20. The final rule looks like this.





Rule Details

Action:
Set a field's value

Field:
EmailAddress

Value:
Value[Key = "mail"]

OK Cancel

Now the email address will be stored in this node, so that you may use it as desired.

