

DEFINE TERMSETS AND USE QRULES TO LOAD THEM INTO YOUR FORMS

PRODUCT: qRules
LAST UPDATED: July 03, 2012

qRules (v4.0 and later) allows you to query your SharePoint managed metadata in your InfoPath forms. In this document you will learn how to query the SharePoint termsets using qRules and display them in your form.

Pre-requisites	1
GetTermSets	3
GetTermsByLabel	12
Using the GetTermSets commands Without GetTermSetIds	17
Cascading dropdowns	20
Support	27

To learn more about SharePoint 2010 Managed metadata, read the following links:

- <http://technet.microsoft.com/en-us/library/ee530393.aspx>
- <http://msdn.microsoft.com/en-us/library/ee559337.aspx>

PRE-REQUISITES

You will first need to add terms to the SharePoint Term Store.

1. On the main SharePoint page (<https://servername/>) select **Site Actions** > **Site Settings**.
2. Under **Site Administration**, click on **Term store management**.



Site Administration

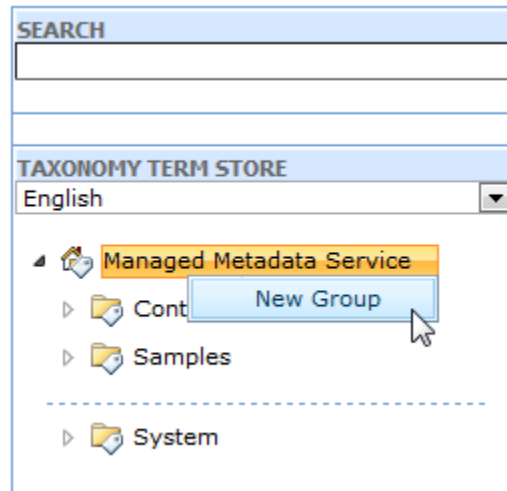
Regional settings
Site libraries and lists
User alerts
RSS
Search and offline availability
Sites and workspaces
Workflow settings
Related Links scope settings
Term store management



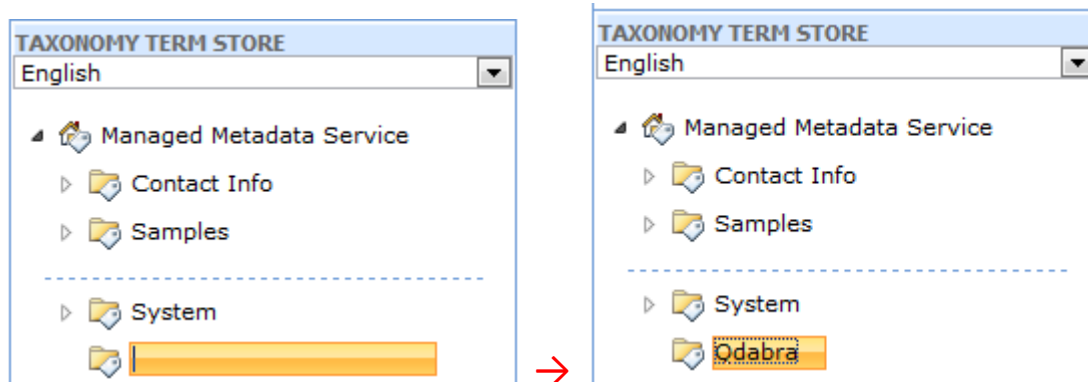
<http://www.qdabra.com>

Copyright © 2006-2012 Autonomy Systems, LLC. All rights reserved.

3. In the left hand taskpane, right click on **Managed Metadata Service** and select **New Group**.



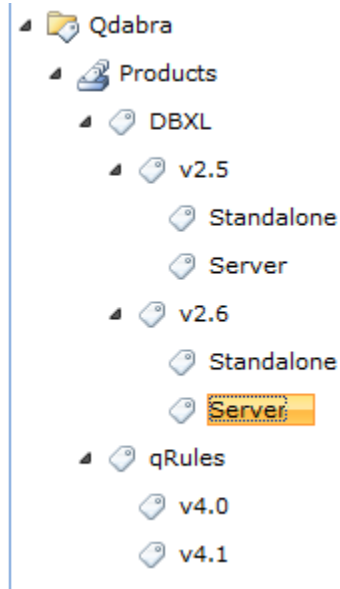
4. Enter a name and tab out.



5. For this group, add a **Description**, **Managers** and **Contributors**, as desired, then click **Save**.
6. On the left hand taskpane, click on **Qdabra** and select **New Term Set**.
7. Enter a name (in this case, **Products**) and hit **Tab** on your keyboard.
8. Click on **Products**, select **Create Term**, enter a term name and Tab using your keyboard.
9. Repeat the step above to add a few terms. Continue adding levels; we'll need these later.

When finished, your list might look like this:





GETTERMSETS

Create a managed Metadata column

In order for the qRules command GetTermSetIds to query the terms added above, we need to add them to our site. This is done by adding a column to a library.

1. Navigate to the **Shared Documents** library.

Note: You could choose to implement this column in another library. We will have details about this in a further step.

2. Go to **Library > Library Settings**.
3. Under **Columns**, click **Create column**.
4. Under **Column name**, enter **ManagedData**.

Name and Type Type a name for this column.	Column name: ManagedData
--	--

5. From the options under **"The type of information in this column is"**, select **Managed Metadata**.



The type of information in this column is:

- ☐ Single line of text
- ☐ Multiple lines of text
- ☐ Choice (menu to choose from)
- ☐ Number (1, 1.0, 100)
- ☐ Currency (\$, ¥, €)
- ☐ Date and Time
- ☐ Lookup (information already on this site)
- ☐ Yes/No (check box)
- ☐ Person or Group
- ☐ Hyperlink or Picture
- ☐ Calculated (calculation based on other columns)
- ☐ External Data
- ☒ Managed Metadata

6. Under **Term Set Settings**, you'll be able to navigate the **Managed Metadata Service** to locate your term set.

Term Set Settings

Enter one or more terms, separated by semicolons, and select Find to filter the options to only include those which contain the desired values.

After finding the term set that contains the list of values to display options for this column, click on a term to select the first level of the hierarchy to show in the column. All levels below the term you select will be seen when users choose a value.

☒ Use a managed term set:

Find term sets that include the following terms.

Managed Metadata Service

▶ Contact Info

▶ Qdabra

▶ **Products**

▶ Samples

This will match what was set up in a previous section.

7. Enter the name of the term set into the field.

Term Set Settings

Enter one or more terms, separated by semicolons, and select Find to filter the options to only include those which contain the desired values.

☒ Use a managed term set:

Find term sets that include the following terms.

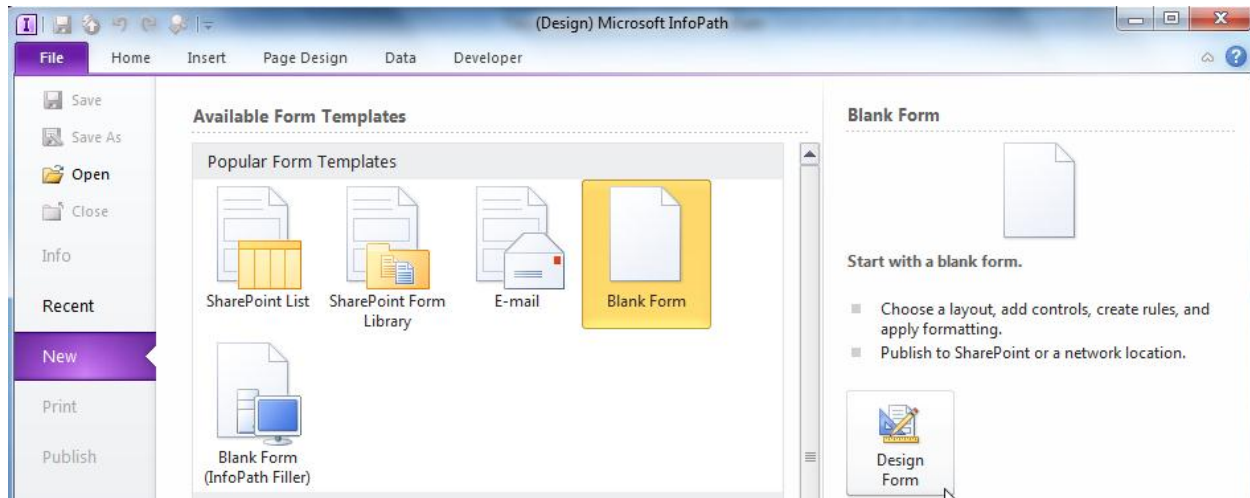
  

8. Click **OK** at the bottom of the page.

Create a form and inject it

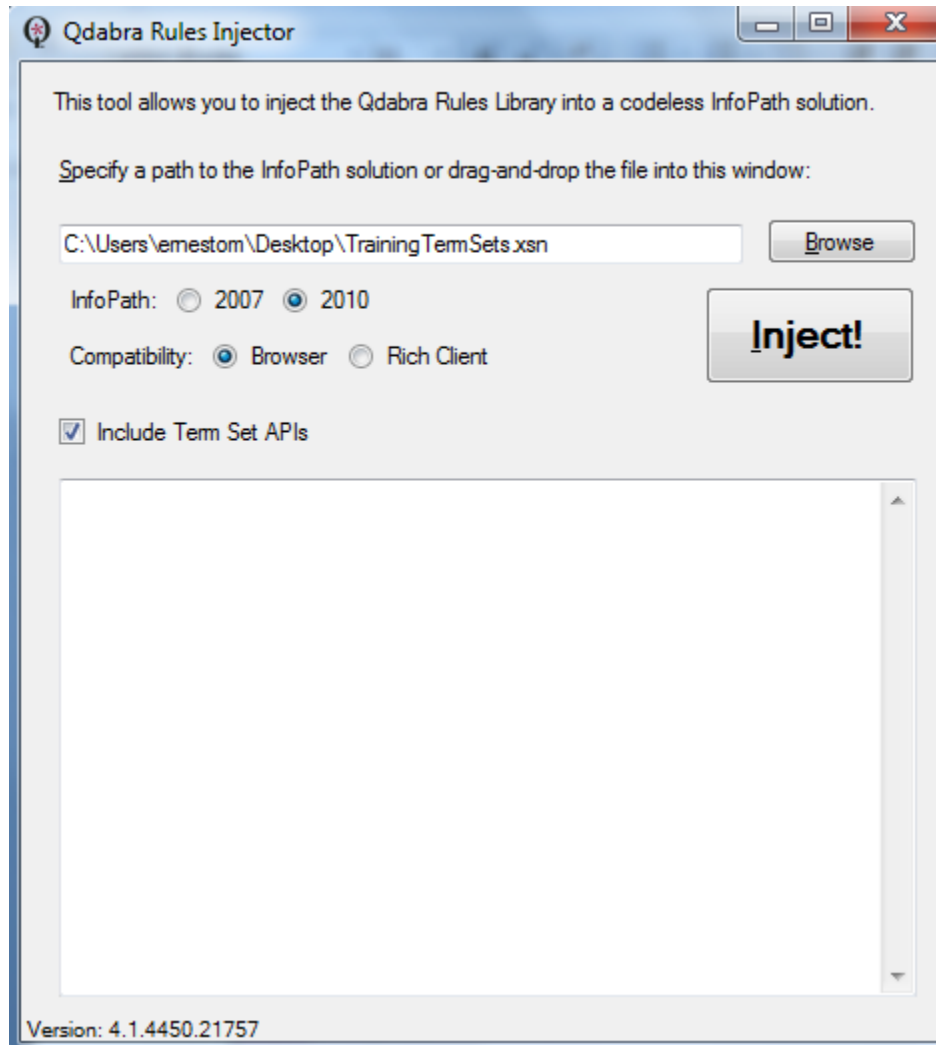
9. Create a new, blank XSN.





10. Select **File > Save** and save the form template to your local drive.
11. Close InfoPath.
12. Launch the Qdabra Rules Injector tool by going to **Start > Programs > Qdabra > Tools > qRules > qRules Injector**.





13. Specify the XSN file you wish to inject with qRules.
14. Check the checkbox for **Include Term Set APIs**.
15. The injector will try to automatically detect the InfoPath version (2007 or 2010). This lab assumes we are working on SharePoint 2010 and InfoPath 2010, so select 2010.
16. Choose the type of form you are injecting qRules into. In this case, it is **Browser** a form.
17. Click the **Inject** button. You will receive a confirmation dialog when the operation has finished.
18. Close the injector.
19. Now right click on the form template you just injected, and select **Design**.

Add the GetTermSets data connection

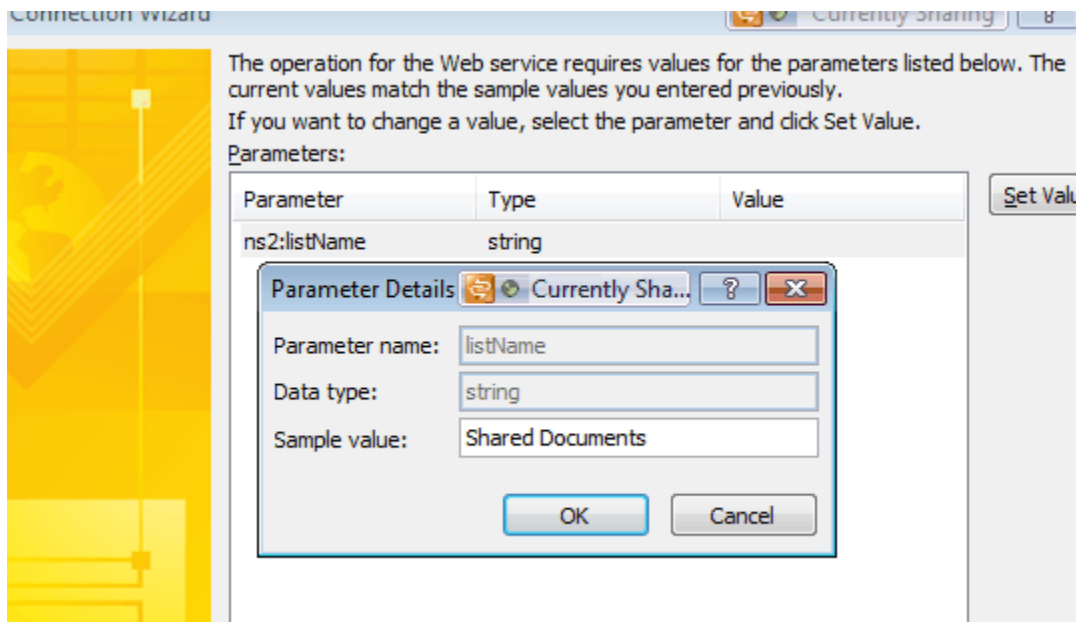
20. Go to **Data > Data Connections** and click **Add**.
21. Select **Create a new data connection to** and **Receive data**. Click **Next**.
22. Select SOAP Web Service and click **Next**.
23. Enter http://<servername>/_vti_bin/taxonomyclientservice.asmx and click **Next**.
24. Select **GetTermSets** and click **Next**.
25. Leave all parameters blank and click **Next** twice.



26. Uncheck the option to ***Automatically retrieve data when the form loads*** and then click **Finish**. Do not change the default data connection name, since the command will look for that.

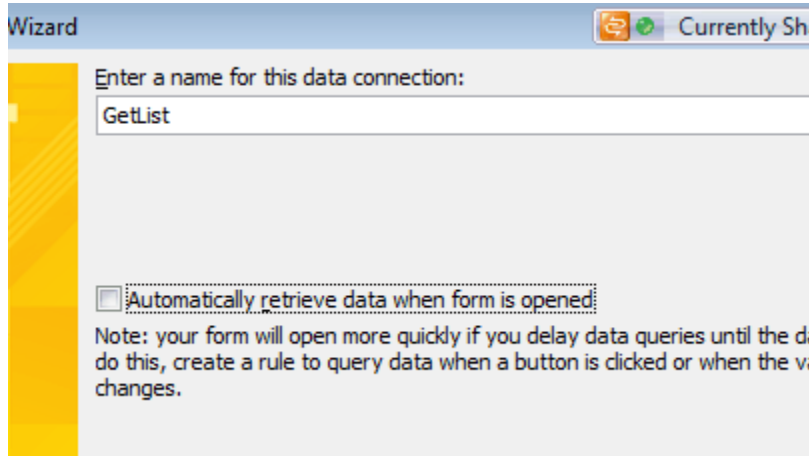
Add the GetList data connection

27. Go to **Data > Data Connections** and click **Add**.
28. Select **Create a new data connection to** and **Receive data**. Click **Next**.
29. Select SOAP Web Service and click **Next**.
30. Enter http://<servername>/_vti_bin/lists.asmx and click **Next**.
31. Select **GetList** and click **Next**.
32. Enter **Shared Documents** and click **Next**. The wizard should prompt you to enter the same value a second time.



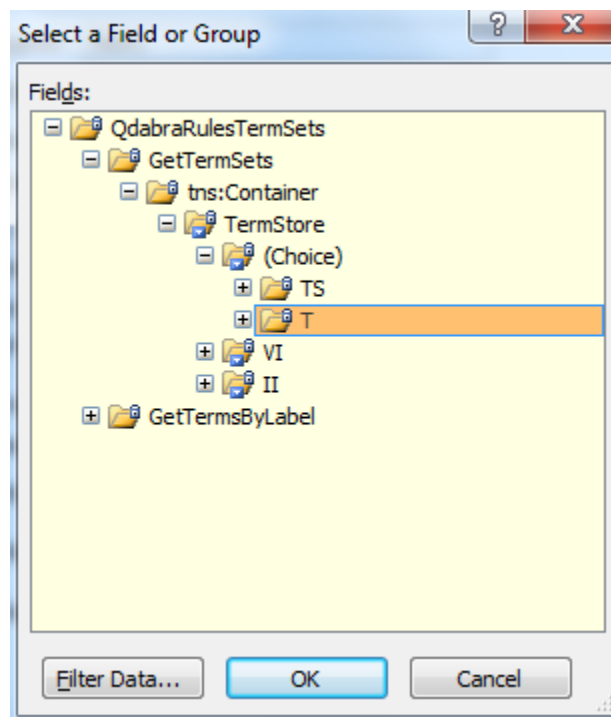
33. Uncheck the option to ***Automatically retrieve data when the form loads*** and then click **Finish**. Do not change the default data connection name, since the command will look for that.





Design the form

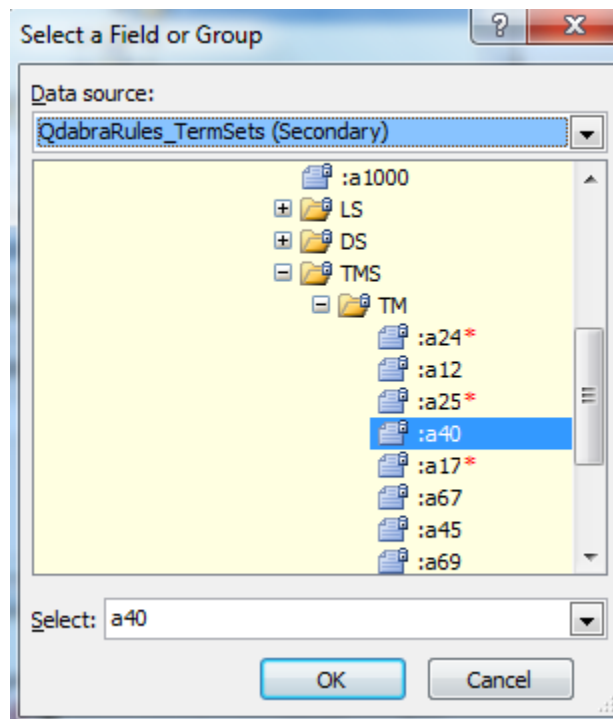
34. Add a dropdown to your canvas.
35. Right click on the dropdown and select **Properties**.
36. Select “**Get choices from an external data source**” and select the **QdabraRules_TermSets** data connection from the dropdown.
37. For Entries, bind to the node:
/ns1:QdabraRulesTermSets/ns1:GetTermSets/tns:Container/tns:TermStore/tns:T



38. Click Filter. You will need to filter this selection to ensure you don't get the entire termStore returned into this node.

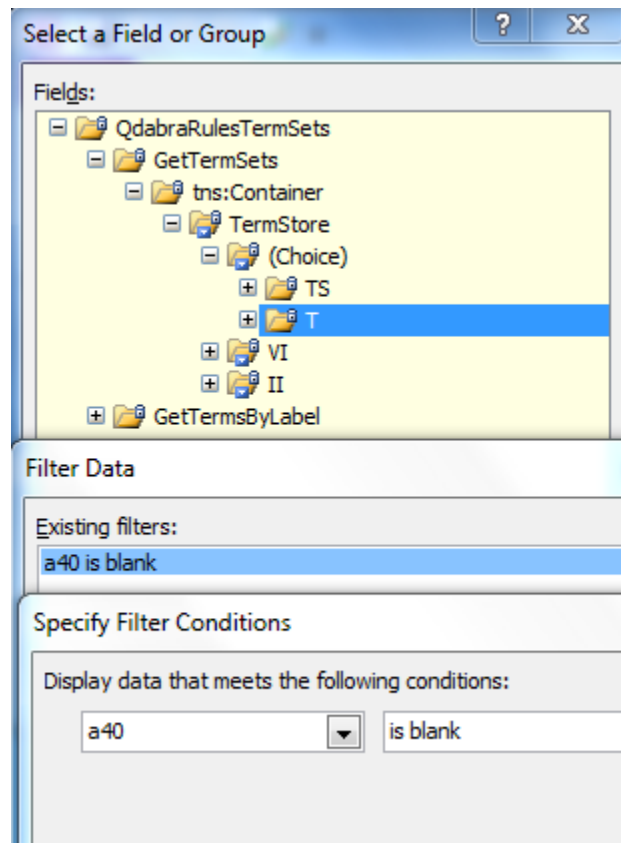


39. Click Add. For the filter, select the a40 node in the first dropdown.

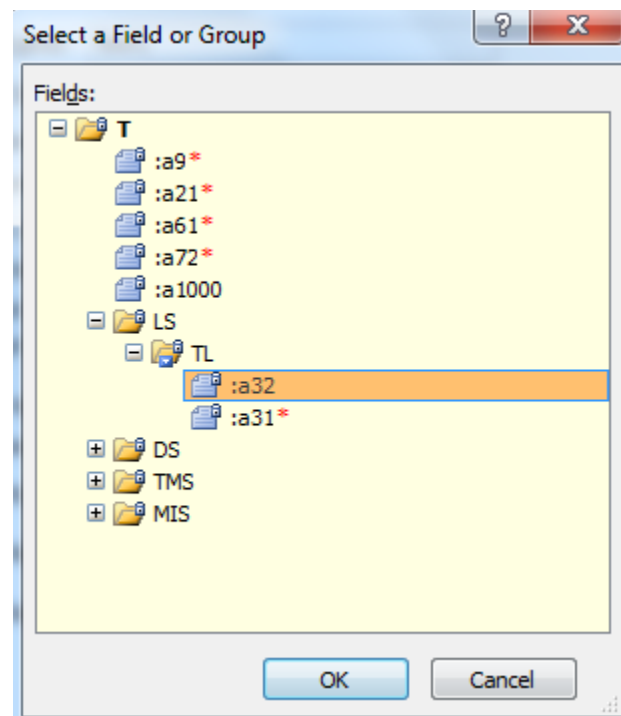


40. In the middle dropdown, select is blank.

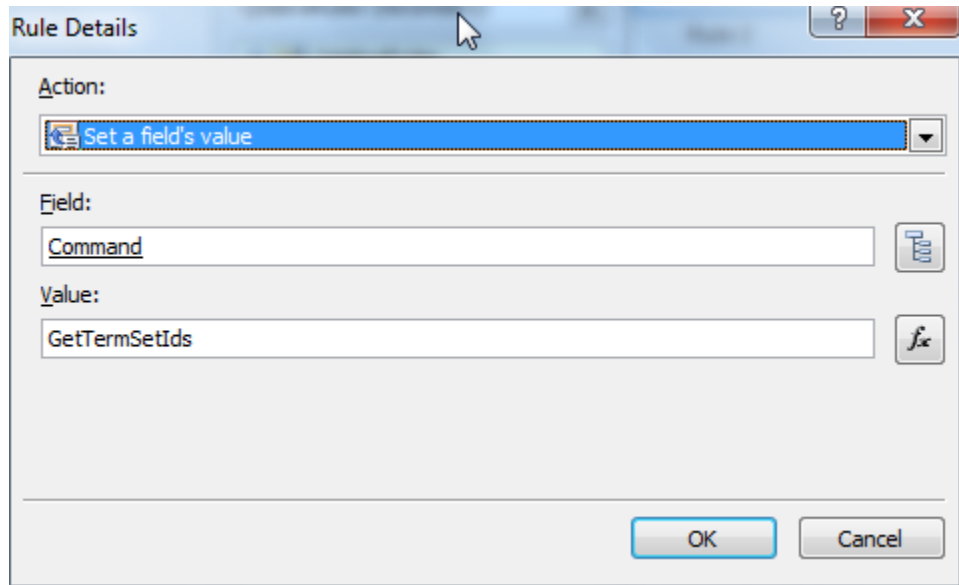




41. Bind the **Value** field to @a9 and the **Display** field to tns:LS/tns:TL/@a32.



42. Select the **finishedLoading** node in the **QdabraRules** secondary data source, then select **Home > Manage Rules**.
43. Click **New > Action**.
44. Select **Add > Set a field's value**.



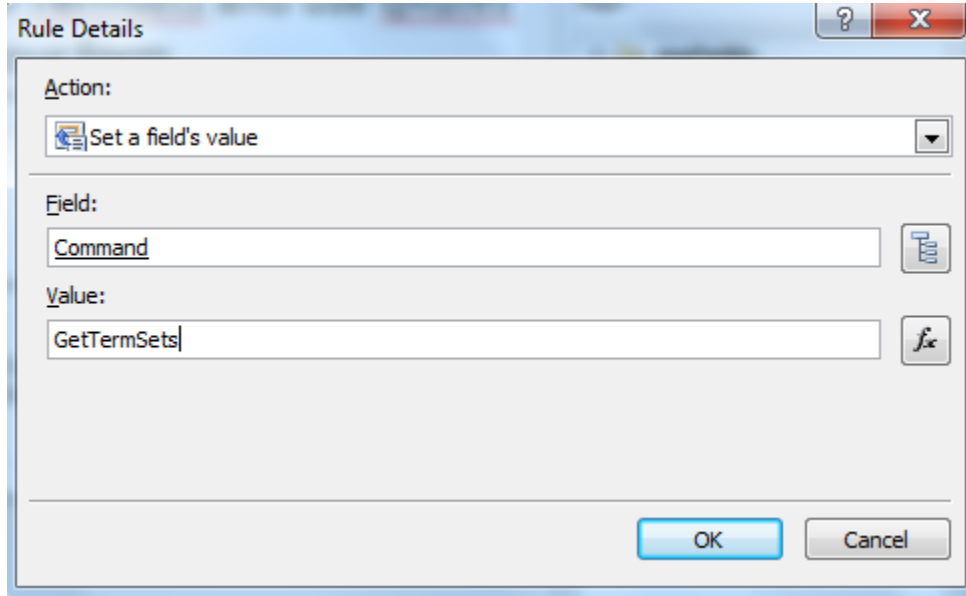
Note: If you create a column in a library other than **Shared Documents**, use the **/list** parameter, like this:

GetTermSetIds /list=Shared Documents

And substitute 'Shared Documents' for the name of your library.

45. For the **Field**, select the **Command** node in the QdabraRules secondary data source.
46. For the **Value**, type in **GetTermSetIds**.
47. Select **Add > Set a field's value**.
48. For the **Field**, select the **Command** node in the QdabraRules secondary data source.
49. For the **Value**, type in **GetTermSets**.





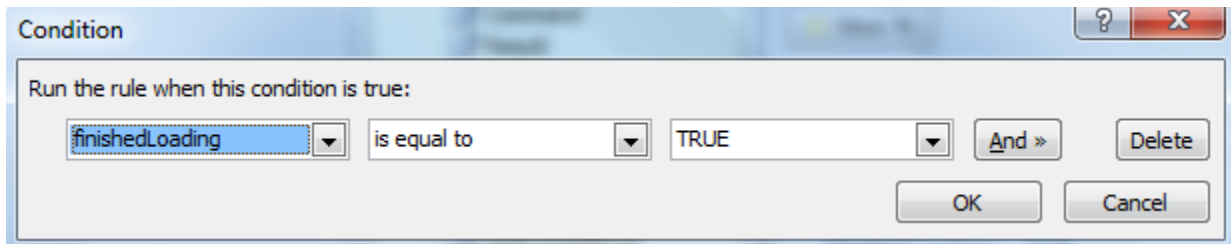
Rule Details

Action:

Field:

Value:

50. Add a condition such that this rule executes only when **finishedLoading** is **true**.



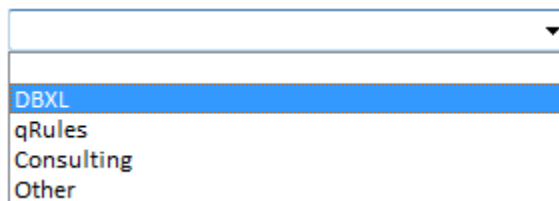
Condition

Run the rule when this condition is true:

is equal to

51. Preview the form.

The terms that were defined on the SharePoint server are now displayed in your form's dropdown.



DBXL
 qRules
 Consulting
 Other

GETTERMSBYLABEL

What if you wanted to use GetTermsByLabel instead? This command does not make use of a Managed Metadata column in a SharePoint form library, since it searches the entire termStore for matching terms.

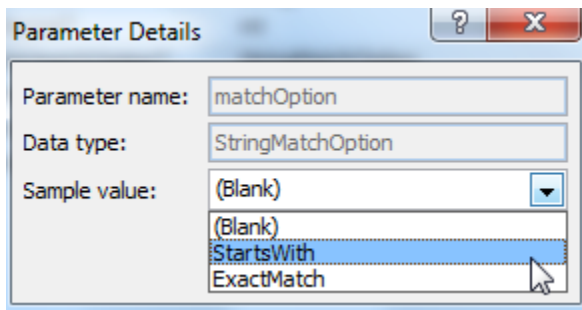
Basic set up



1. Complete the pre-requisites.
2. Complete steps under “Create a form and inject it”.
3. Open the form in Design mode.

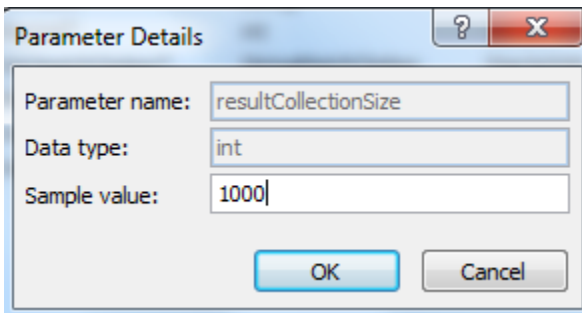
Add the GetTermSets data connection

4. Go to **Data > Data Connections** and click **Add**.
5. Select **Create a new data connection to** and **Receive data**. Click **Next**.
6. Select SOAP Web Service and click **Next**.
7. Enter http://<servername>/_vti_bin/taxonomyclientservice.asmx and click **Next**.
8. Select **GetTermsByLabel** and click **Next**.
9. For **lcid**, enter 1033 (or the code for the language of your choice).
10. For **matchOption**, select your desired search method. We will proceed with StartsWith in this example.



The dialog box titled "Parameter Details" shows the configuration for the matchOption parameter. The "Parameter name" is "matchOption", the "Data type" is "StringMatchOption", and the "Sample value" is set to "StartsWith" from a dropdown menu. The dropdown menu also shows "(Blank)" and "ExactMatch".

11. For **resultCollectionSize**, enter a value in case it is necessary to limit the number of results returned.



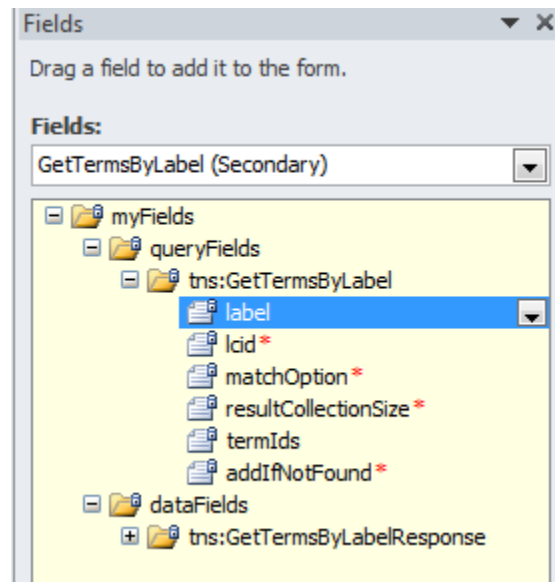
The dialog box titled "Parameter Details" shows the configuration for the resultCollectionSize parameter. The "Parameter name" is "resultCollectionSize", the "Data type" is "int", and the "Sample value" is "1000". The "OK" and "Cancel" buttons are at the bottom.

12. Click **Next** twice.
13. Uncheck the option to **Automatically retrieve data when the form loads** and then click **Finish**. Do not change the default data connection name, since the command will look for that. Click Close.

Design the canvas

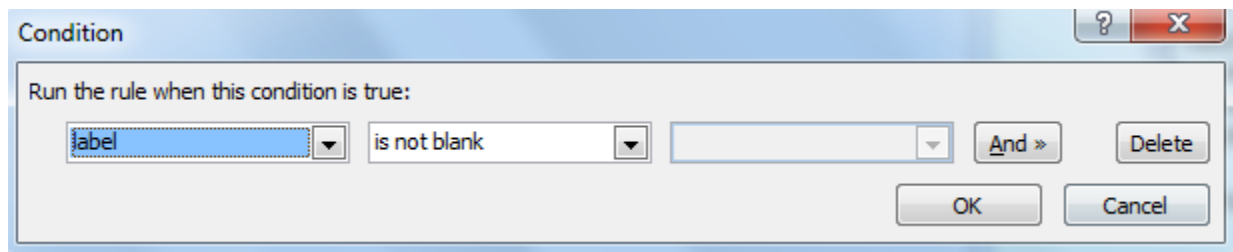
14. Drag and drop the label node, from the GetTermsByLabel secondary data connection, into your canvas. This is where users will enter a search term.





15. Select this textbox, and then select **Home > Manage Rules.**

16. Add the condition: ***label is not blank***

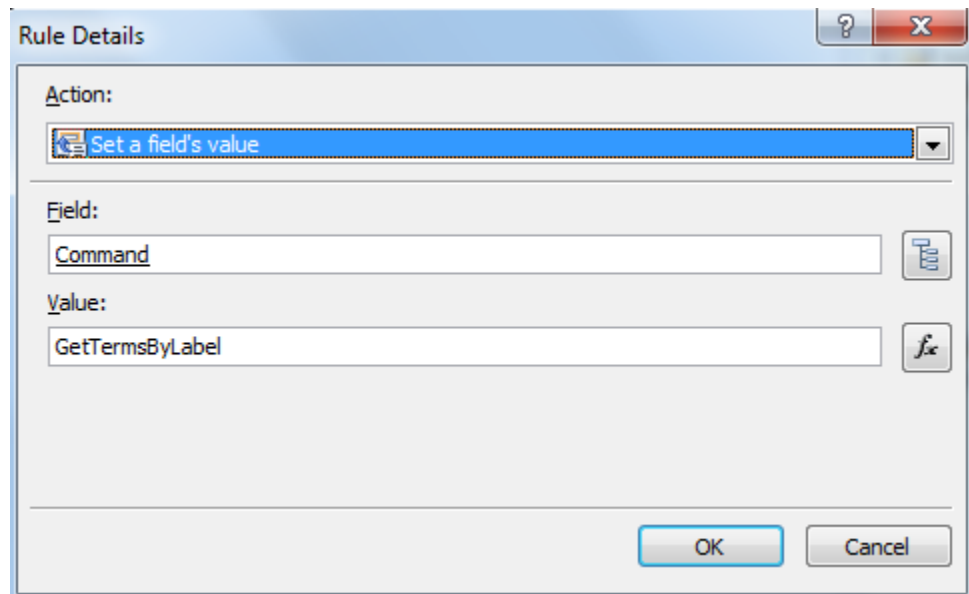


17. Click **Add > Set a field's value.**

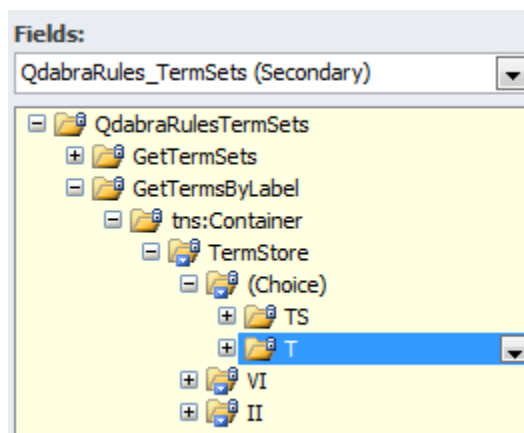
18. For **Field**, select the **Command** node in the QdabraRules data connection.

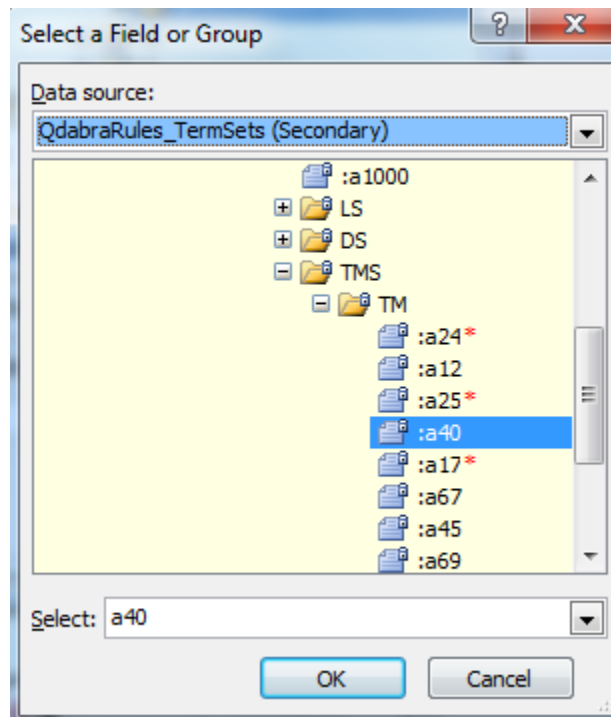
19. For **Value**, type in **GetTermsByLabel.**



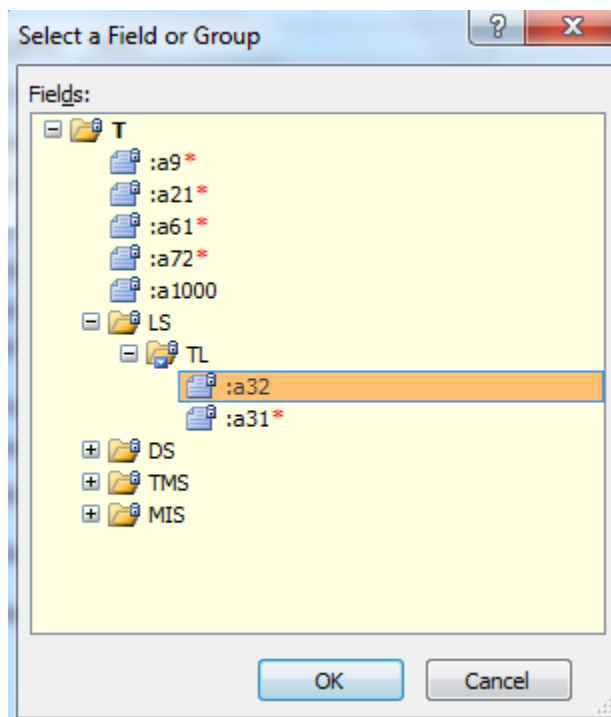


20. Add a dropdown to your canvas.
21. Right click on the dropdown and select **Properties**.
22. Select **"Get choices from an external data source"** and select the **QdabraRules_TermSets** data connection from the dropdown.
23. For Entries, bind to the node:
/ns1:QdabraRulesTermSets/ns1:GetTermsByLabel/tns:Container/tns:TermStore/tns:T





24. Bind the **Value** field to @a9 and the **Display** field to tns:LS/tns:TL/@a32.



25. Preview the form and test this out!



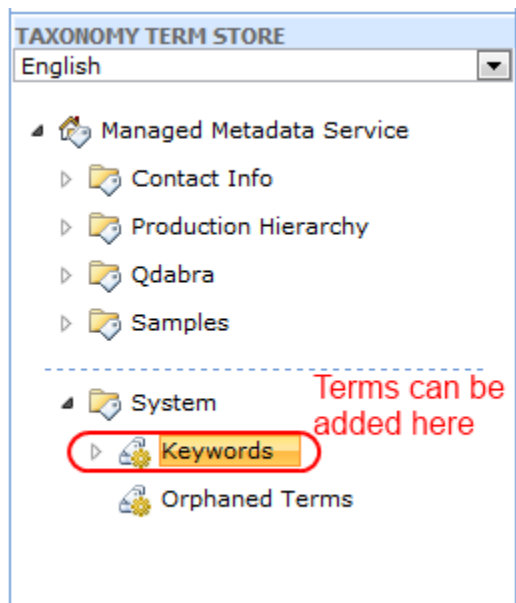
Search term:

Matching results:

Access Control
Account Analysis
Accounting and Finance
Accounting Method Conversions
Accounting Policies
Accounts and Account Codes
Accounts Payable
Accounts Receivable
Accrual Policies

Extra – Add Tags with *GetTermsByLabel*

It is also possible to add terms to the Keywords store:



For step 10 above, set **matchOption** to **ExactMatch**. Then set the **addIfNotFound** parameter in the **GetTermsByLabel** data connection to true. When you try to search for a term, if the exact match does not exist the new term will be added to the system Keywords.

USING THE **GETTERMSSETS** COMMANDS WITHOUT **GETTERMSETIDS**

When using the **GetTermSets** command, added in qRules v4.0, you can choose to not use the **GetTermSetIds** command (added in v4.1). In place of the **GetTermSetIds** command, you will need to manually provide the



parameter values when you create the data connection to the Taxonomy Client web service (http://<servername>/_vti_bin/taxonomyclientservice.asmx?WSDL).

When you create the data connection, using the InfoPath data connection wizard, you'll be asked to select between the two options below and provide values for the necessary parameters. If you encounter issues using these commands, please review the values for the data connection parameters. Any error in the query parameter values will cause the commands to fail.

GetTermSets

- sharedServicesIds:
 - The format for this parameter is `<termStoreId>3dfdd366-4b8b-4a26-b600-a7a39617c315</termStoreId>`. Use the powershell script later in this appendix to obtain the value for your environment's termStoreId.
 - Description*: *The identifiers of the term stores in which the requested term sets reside. The number of sharedServiceIds child nodes MUST equal the number of termSetIds child nodes and the two sets of child nodes MUST be in one-to-one correspondence so that each sharedServiceIds child represents the term store identifier of the corresponding term set in the termSetIds, which is an XML encoded string that MUST conform to the schema of the SspIdsForGetTermSetsClientService complex type (section 2.2.4.13). An empty string with no term set information MUST be returned if this node is absent.*
- termSetIds:
 - The format for this parameter is `<termSetId>52b15e82-6331-4a6e-9f35-6e569d51d64b</termSetId>`. Use the powershell script later in this appendix to obtain the value for your environment's termSetIds.
 - Description*: *The term set identifiers of the term sets to be retrieved, which is an XML encoded string that MUST conform to the schema of the TermSetIdsForGetTermSetsClientService complex type (section 2.2.4.16).*
- lcid:
 - The value of the language in your SharePoint environment. The default value is 1033, for English.
 - For additional information, see this link: <http://msdn.microsoft.com/en-us/goglobal/bb964664>
 - Description*: *The LCID of the requested language*
- clientTimeStamps:
 - Example: `<timeStamp>1900-01-01T00:00:00</timeStamp>`
 - Description*: *The time in ticks used to determine whether or not to return the term set data. This is an XML encoded string that MUST conform to the schema of the ClientTimeStampsForGetTermSetsClientService complex type (section 2.2.4.2). The number of clientTimeStamps child nodes MUST equal the number of termSetIds child nodes and these two sets of child nodes MUST be in one-to-one correspondence. For each term set specified in termSetIds, if the term set has been updated since the time specified by the corresponding time stamp in clientTimeStamps, the protocol server MUST return the data for the term set. Otherwise, the protocol server MUST return an empty node for the term set. An empty string with no term set information MUST be*



returned if this node is absent. A protocol client can request that term set data be returned regardless of last update time by specifying the text string "1900-01-01T00:00:00" instead of the integer number for the time stamp (since unparseable values are treated as 0).

- clientVersions:
 - Example: <version>1</version>
 - Description*: The server version numbers that the term set data stored on the client was retrieved from. The number of clientVersions child nodes MUST equal the number of termSetIds child nodes and these two sets of child nodes MUST be in one-to-one correspondence. For each term set specified in termSetIds, if the term set is already stored on the client it MUST send 1, otherwise it MUST send 0. This is an XML encoded string that MUST conform to the schema of the ClientVersionsForGetTermSetsClientService complex type (section 2.2.4.3). An empty string with no term set information MUST be returned if this node is absent.

*from the [Microsoft Enterprise Managed Metadata Web Service Protocol Specification](#)

Powershell

Since you are not using the **GetTermSetIds** command, you will also need information about your termStore in SharePoint. SharePoint administrators can use the following powershell script (after changing site url):

```
#Connect to Central Admin
$taxonomySite = get-SPSite http://<servername>
#Connect to Term Store in the Managed Metadata Service Application
$taxonomySession = Get-SPTaxonomySession -site $taxonomySite
$taxonomySession
$termStore = $taxonomySession.TermStores["Managed Metadata Service"]
$termStore
write-host "Connection made with term store -"$termStore.Name
write-host "*****"
# Output info about termsets
foreach ($g in $termStore.Groups) { $g; $g.TermSets; write-host
"=====";}
#Dispose of taxonomy site object
$taxonomySite.Dispose()
```

If the termStore does not contain the terms you require, please visit the following URL to add additional terms:

<http://<servername>/Layouts/termstoremanager.aspx>

To learn more about SharePoint 2010 Managed metadata, read the following links:

<http://technet.microsoft.com/en-us/library/ee530393.aspx>

<http://msdn.microsoft.com/en-us/library/ee559337.aspx>



CASCADING DROPDOWNS

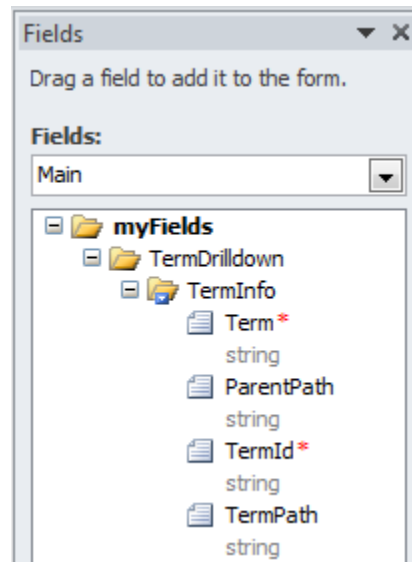
Now that you have seen the basics of the command, we present a more complex scenario, in which cascading dropdowns allow you to drill down the taxonomy. This tutorial will use the **GetTermSetIds** command.

Basic set up

26. Complete the pre-requisites.
27. Complete steps under "Create a form and inject it".
28. Complete steps under "Add the GetTermSets data connection".
29. Complete steps under "Add the GetList data connection".

Design your form schema

30. Add the following schema to your form.

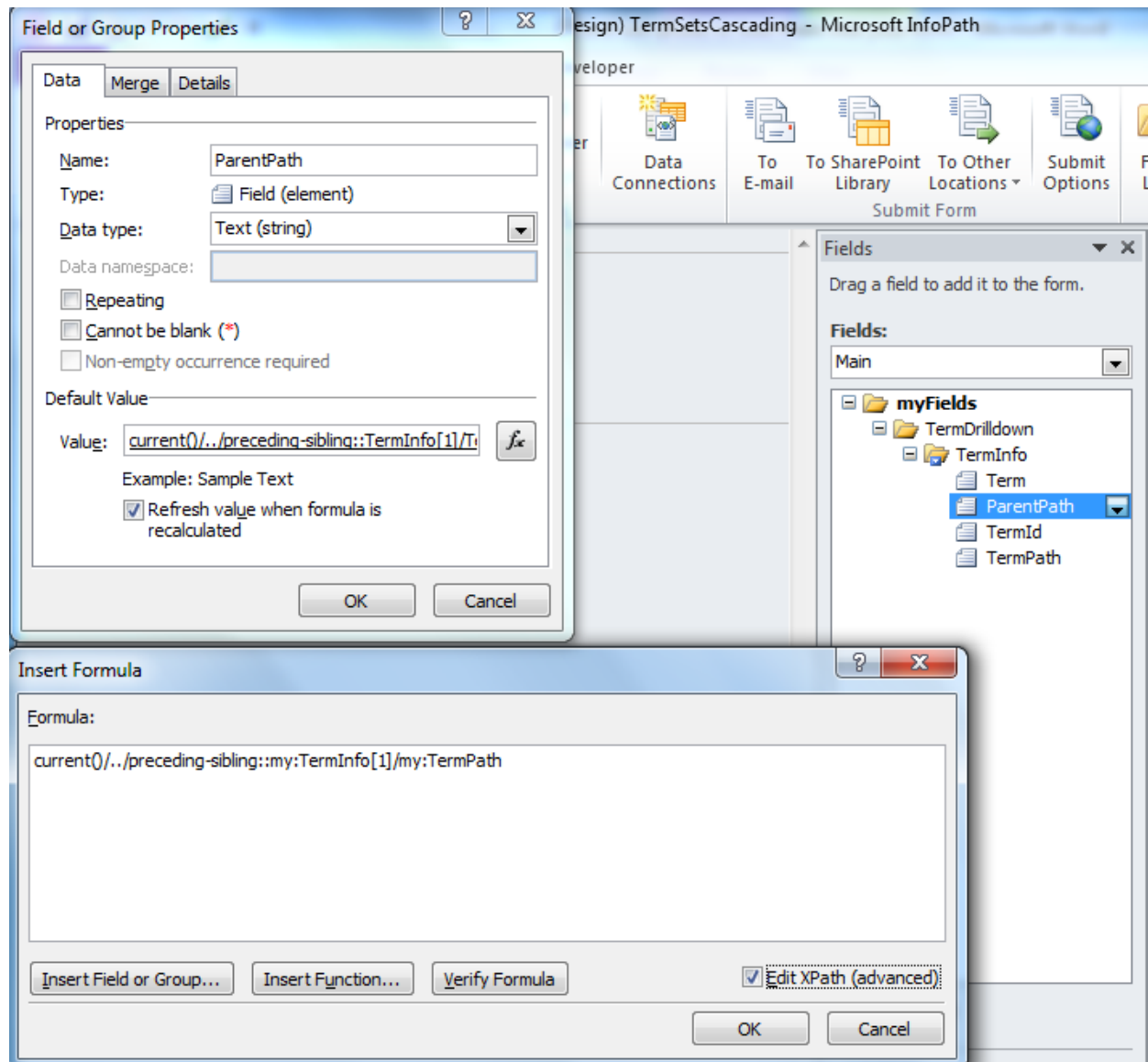


31. Right click on ParentPath and select **Properties**.
32. Under **Default Value**, click **fx**.
33. Paste the following:

```
current()/../preceding-sibling::my:TermInfo[1]/my:TermPath
```

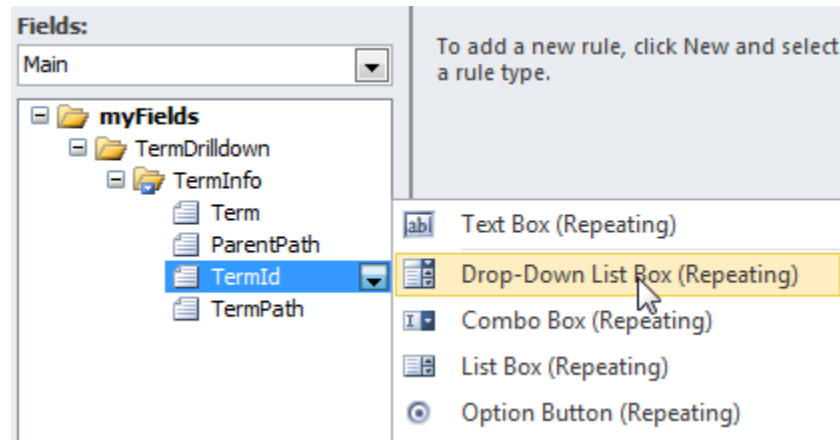
34. Click OK twice. The screenshot below shows you the setup for this Default value.



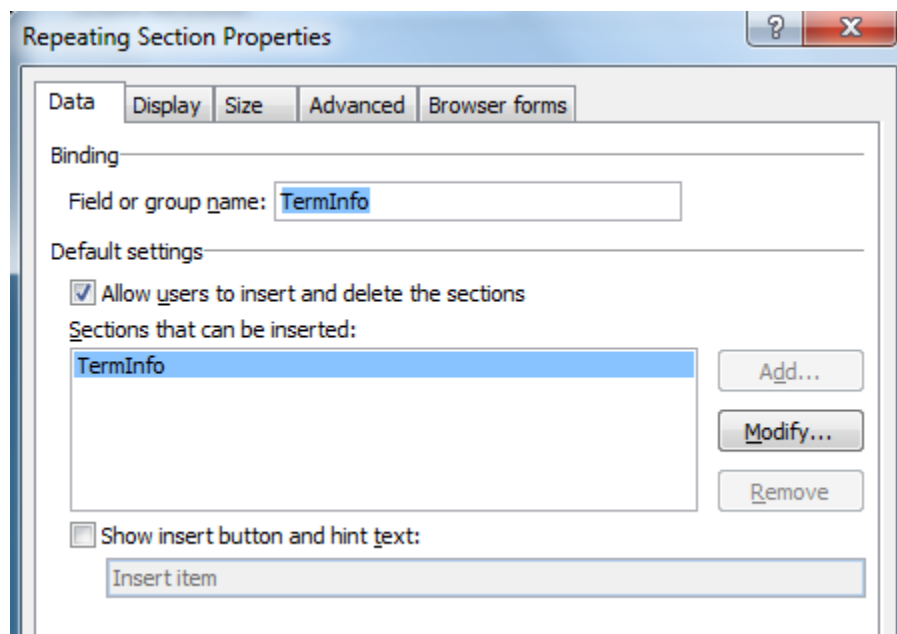


35. Right click on **TermId** and select **Drop-Down List Box (Repeating)**.





36. Right click on the repeating table and select **Properties**.
37. Under **Default settings**, uncheck "Show insert button and hint text", and then click OK.



38. Right click on the new dropdown in your canvas, and select **Properties**.
39. Under **List box choices**, select **Get choices from an external data source**.
40. In the **Data source** dropdown, select the **QdabraRules_TermSets** data connection.
41. For **Entries**, select:

```
/ns3:QdabraRulesTermSets/ns3:GetTermSets/tns:Container/tns:TermStore/tns:T[tns:TMS/tns:TM/@a40 = current()/my:ParentPath]
```

To accomplish this, you will need to select the node:

```
/ns3:QdabraRulesTermSets/ns3:GetTermSets/tns:Container/tns:TermStore/tns:T
```



Then click **Filter Data**, and then click **Add**.

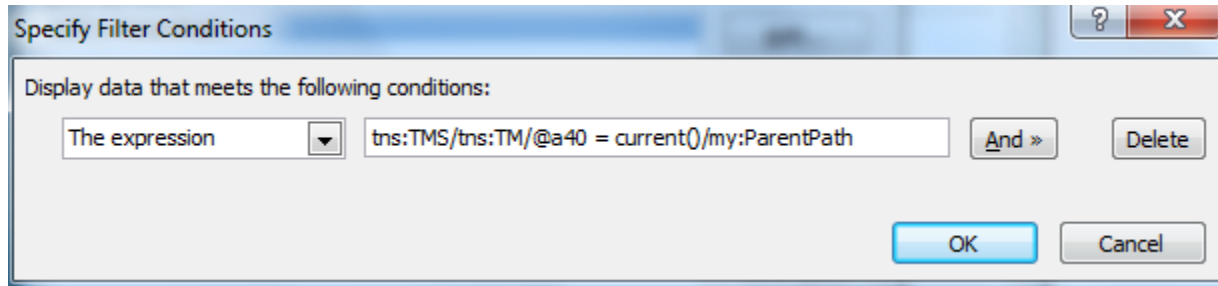
When specifying the filter, there will be three dropdowns.

42. In the first dropdown, select the field **tns:TMS/tns:TM/@a40**

43. In the third dropdown, select **Use a formula** and enter **current()/my:ParentPath**

As an alternative, you can select “The expression” in the first dropdown, and enter:

tns:TMS/tns:TM/@a40 = current()/my:ParentPath

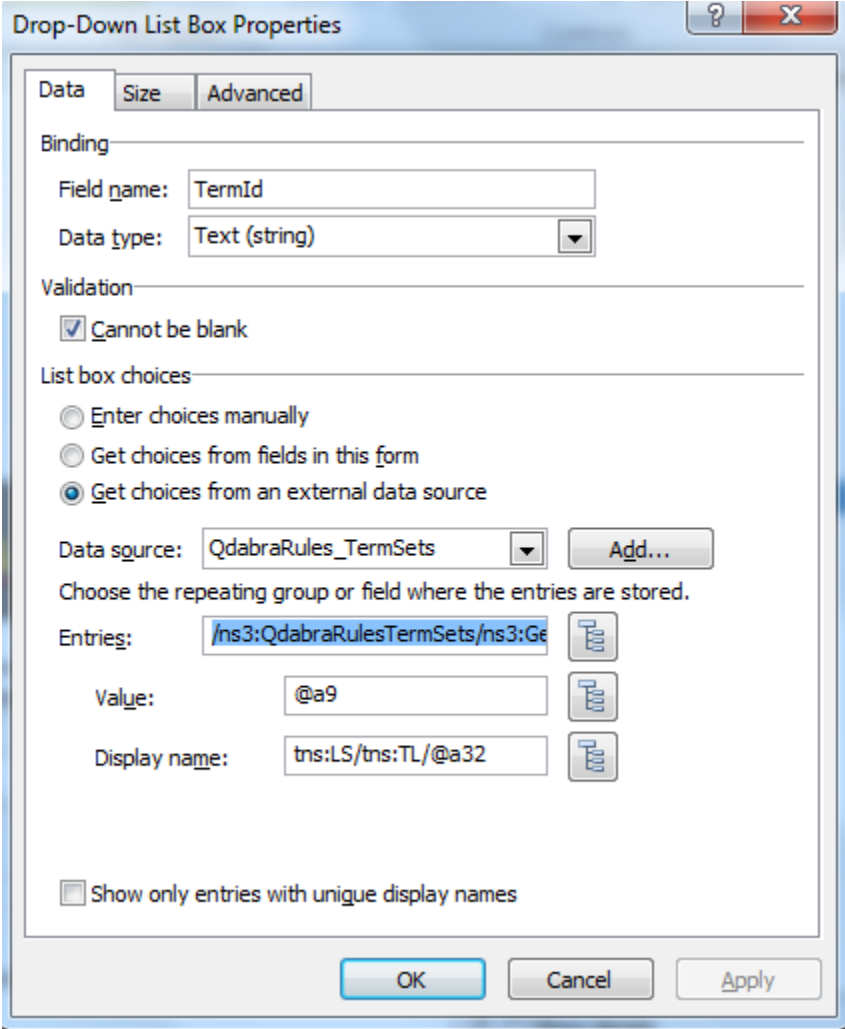


44. Click OK to return to the Drop Down List Box properties.

45. For **Value**, select @a9

46. For **Display Name**, select tns:LS/tns:TL/@a32





Add Form Logic on load

47. Select the **finishedLoading** node in the **QdabraRules** secondary data source, then select **Home > Manage Rules**.
48. Click **New > Action**.
49. Select **Add > Set a field's value**.
50. For the **Field**, select the **Command** node in the QdabraRules secondary data source.
51. For the **Value**, type in **GetTermSetIds**.
52. Select **Add > Set a field's value**.
53. For the **Field**, select the **Command** node in the QdabraRules secondary data source.
54. For the **Value**, type in **GetTermSets**.
55. Add a condition such that this rule executes only when **finishedLoading** is **true**.

Add rules to the TermInfo table

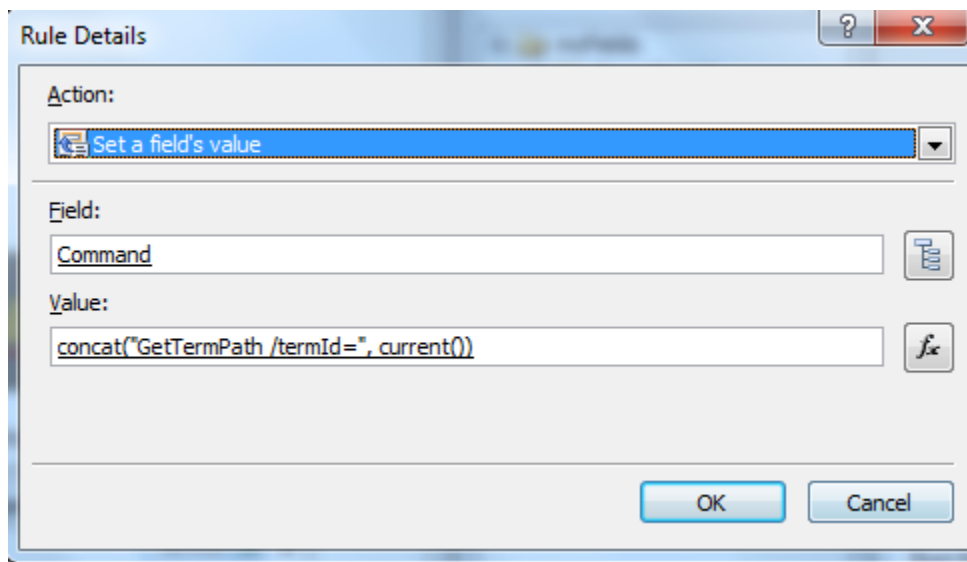
56. Select the **TermId** node on the main data source, and then select **Home > Manage Rules**.



57. Select **New > Action**.
58. Call this rule **UpdateSiblingValues**.
59. Click **Add > Set a field's value**.
60. For the **Field**, select the **Term** node in the main data source.
61. For the **Value**, click **fx** and enter

```
xdXDocument:GetDOM("QdabraRules_TermSets")/ns1:QdabraRulesTermSets/ns1:GetTermSets/tns:Container/tns:TermStore/tns:T/tns:LS/tns:TL/@a32[../@a9 = current()]
```

62. Click **Add > Set a field's value**.
63. For the **Field**, select the **Command** node from the QdabraRules data connection.
64. For **Value**, click **fx**, and enter `concat("GetTermPath /termId=", current())`



65. Click **Add > Set a field's value**.
66. For the **Field**, select the **TermPath** node.
67. For **Value**, select the **Result** node from the QdabraRules data connection.
68. Select **New > Action**.
69. Call this rule **AddTermInfoRow**.
70. Click **Add > Set a field's value**.
71. For the **Field**, select the **Command** node from the QdabraRules data connection.
72. For **Value**, enter `Insert /parent=/my:myFields/my:TermDrilldown /child=my:TermInfo`
73. Add a condition for this second rule.



Condition

Run the rule when this condition is true:

TermPath	is not blank		and	Delete
a25	is equal to	TermId	And »	Delete

OK Cancel

74. TermPath and TermId are in the main data source. The **a25** node can be found here:

Select a Field or Group

Data source: QdabraRules_TermSets (Secondary)

- QdabraRulesTermSets
 - GetTermSets
 - tns:Container
 - TermStore
 - (Choice)
 - TS
 - T
 - :a9 *
 - :a21 *
 - :a61 *
 - :a72 *
 - :a1000
 - LS
 - DS
 - TMS
 - TM
 - :a24 *
 - :a12
 - :a25 ***
 - :a40
 - :a17 *
 - :a67
 - :a45
 - :a69
- MIS
 - VI
 - II
- GetTermsByLabel

Select: a25

OK Cancel



Test

75. Preview the form.
76. In the first dropdown, make a selection.
77. The second dropdown will show. Make a selection.
78. If you have selected a taxonomy with more levels, you may continue to get more cascading dropdowns. Keep going until no more dropdowns show – this means you have reached the final level of the taxonomy.

A screenshot of a cascading dropdown menu. The first dropdown is open, showing three options: 'DBXL', 'v2.5', and 'Standalone'. Each option has a small downward arrow on the right side of the text.

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 [Qdabra Product support forums](http://InfoPathDev.com) to request help from the community.

Finally, the qRules product page on Qdabra.com will contain an updated list of all available documentation.

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>

