

CREATE A SIMPLE WORKFLOW USING XML-TO-XML

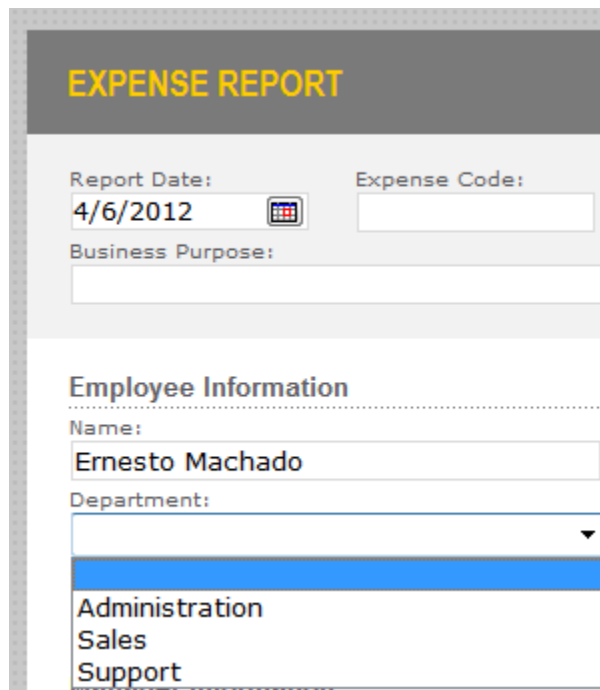
In this exercise, we will modify the Expense Report to show how simple workflows can be done in the form as opposed to SharePoint.

ADD THE WORKFLOW XML DATA CONNECTION

We are going to add the configuration XML as a resource to our form template.

1. Save the *Workflow.xml* in your local drive.
2. Open in the text editor of your form, such as Notepad.

Recall that in a previous lab we configured the Department dropdown:

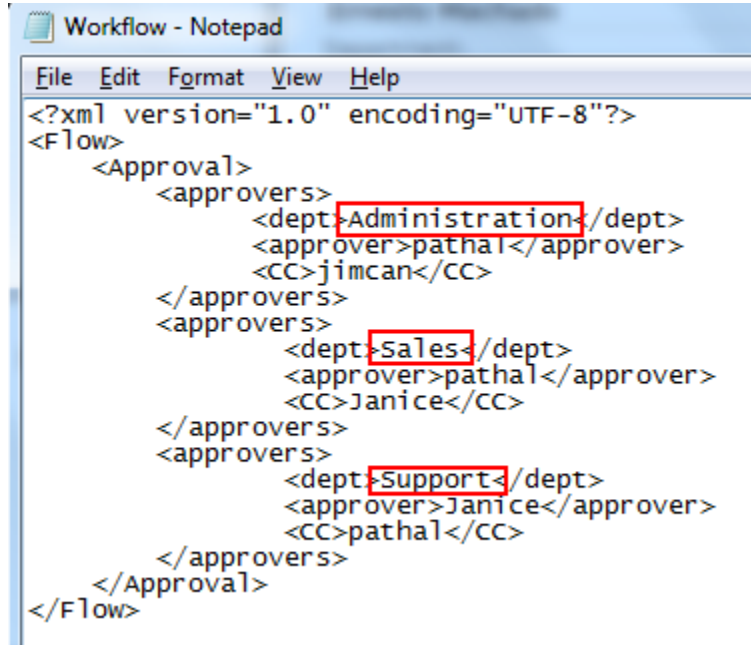


The image shows a screenshot of an "EXPENSE REPORT" form. The form has a title bar "EXPENSE REPORT" in yellow text on a dark grey background. Below the title bar, there are three input fields: "Report Date:" with the value "4/6/2012" and a calendar icon, "Expense Code:" with an empty text box, and "Business Purpose:" with an empty text box. Below these fields, there is a section titled "Employee Information" with a dotted line separator. Under this section, there are two fields: "Name:" with the value "Ernesto Machado" and "Department:" with a dropdown menu. The dropdown menu is open, showing three options: "Administration", "Sales", and "Support".

We need the Workflow xml to have three workflows as well, one for each of these dropdown options.

Qdabra DBXL Training: Module 12

3. Modify the Workflow XML such that we have a workflow for each department.



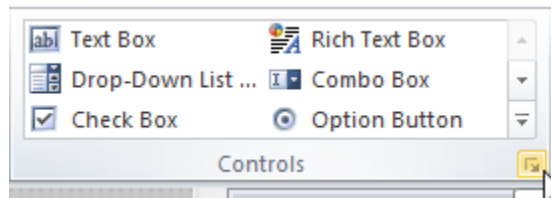
```
<?xml version="1.0" encoding="UTF-8"?>
<Flow>
  <Approval>
    <approvers>
      <dept>Administration</dept>
      <approver>pathal</approver>
      <CC>jimcan</CC>
    </approvers>
    <approvers>
      <dept>Sales</dept>
      <approver>pathal</approver>
      <CC>Janice</CC>
    </approvers>
    <approvers>
      <dept>Support</dept>
      <approver>Janice</approver>
      <CC>pathal</CC>
    </approvers>
  </Approval>
</Flow>
```

4. Modify the approver and CC nodes as well, to add the users you wish to include as part of this workflow.
5. When done, save the Workflow.xml.
6. Right click on the Expense Report form and select Design.
7. In the InfoPath 2010 ribbon, select **Data** and then click on **Data Connections**.
8. Click **Add**.
9. Select the radio button to **Create a New Connection to** and **Receive Data**. Click **Next**.
10. Select **XML document**. Click **Next**.
11. Browse to the location where you saved Workflow.xml. Click **Open**. Click **Next**.
12. Select the option **Include the data as a resource file in the form template or template part**. Click **Next**.
13. Uncheck **Automatically retrieve data when form is opened**. Click **Finish** and then **Close**.

ADD THE QDFLOW XTP TO THE FORM TO ENABLE XML-TO-XML

Qdabra DBXL Training: Module 12

14. Save the *ExpenseReport_QdFlow.xtp* template part in your local drive.
15. Back in the Expense Report form, click the **Home** tab.
16. Expand the **Controls** section to open the taskpane.

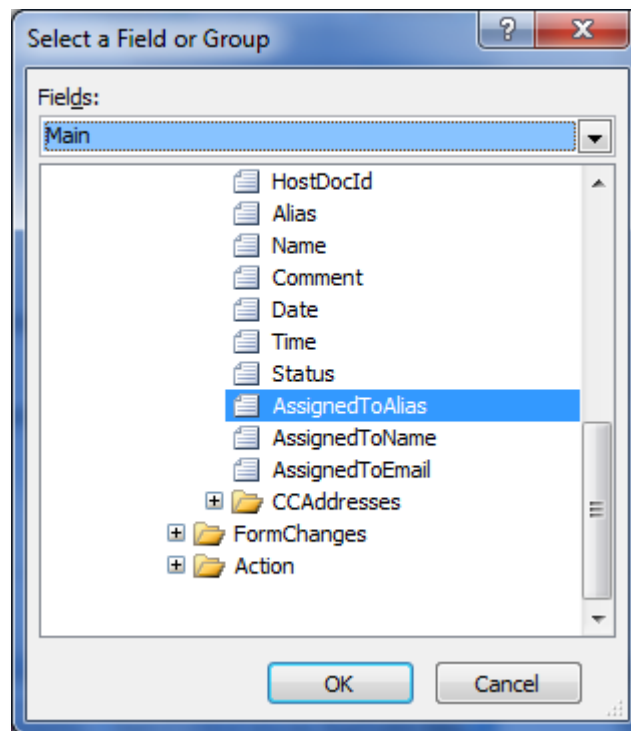


17. Click **Add or Remove Custom Controls** found at the bottom of the task pane.
18. Click **Add**.
19. Select **Template Part**. Click **Next**.
20. Browse to the location where you saved *ExpenseReport_QdFlow.xtp*. Double-click to open and then click **Finish**. Click **Close**, click **OK**. You will now see the template part named **QdFlow** under Custom section.
21. Put your cursor in your canvas in the location where you wish to add the XTP.
22. In the Control's task pane, click the QdFlow template part. The template part schema is now added in your Main data connection.
23. Save the changes in the *ExpenseReport.xsn*.

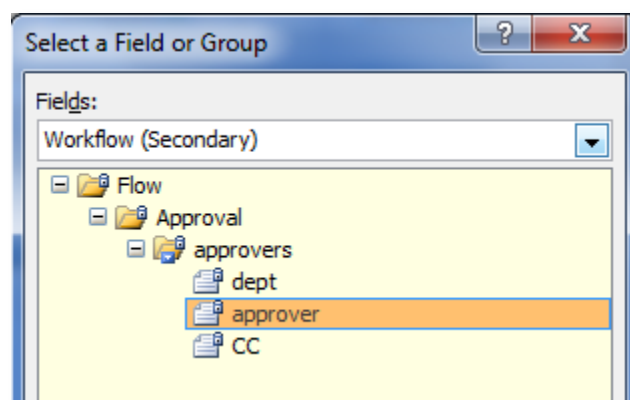
MODIFY THE FORM TEMPLATE TO CONFIGURE FORM WORKFLOW

24. Click **Manage Rules** for the **Department** field. Add the following rules:
 - a. Click **New > Action**.
 - b. Enter the rule name: **Set Approver and CC**.
 - c. Next to **Run these actions**, click **Add > Set a field's value**.
 - d. Click the **Modify** button  to set the value for **Field**.
 - e. Double click on **AssignedToAlias** node under **qdxtp_Flow**
(/my:expenseReport/my:qdxtp_Flow/my:Content/my:myFields/my:AssignedToAlias).

Qdabra DBXL Training: Module 12

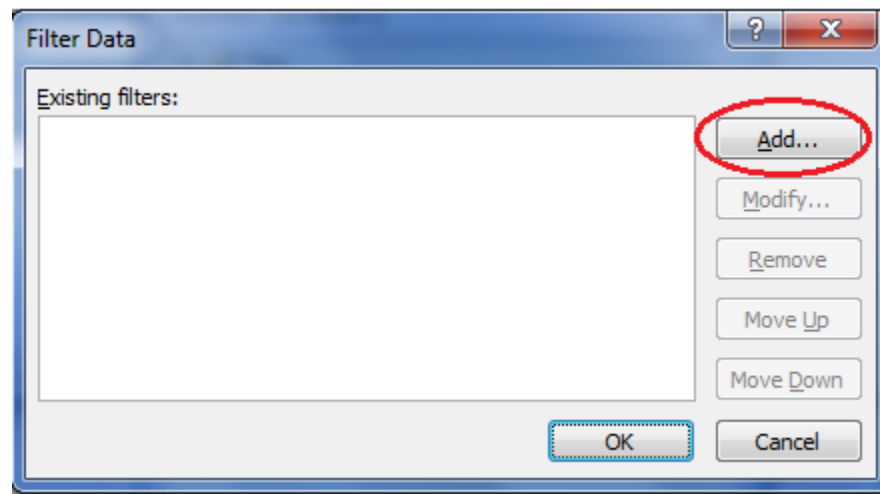


- f. Click the **fx** button  next to the **Value** field.
- g. Click **Insert Field or Group**. Click on the data source dropdown and choose the **Workflow** secondary data connection. Click the **approver** field.

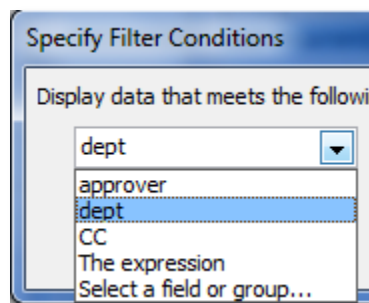


- h. Click **Filter Data**. Click **Add**.

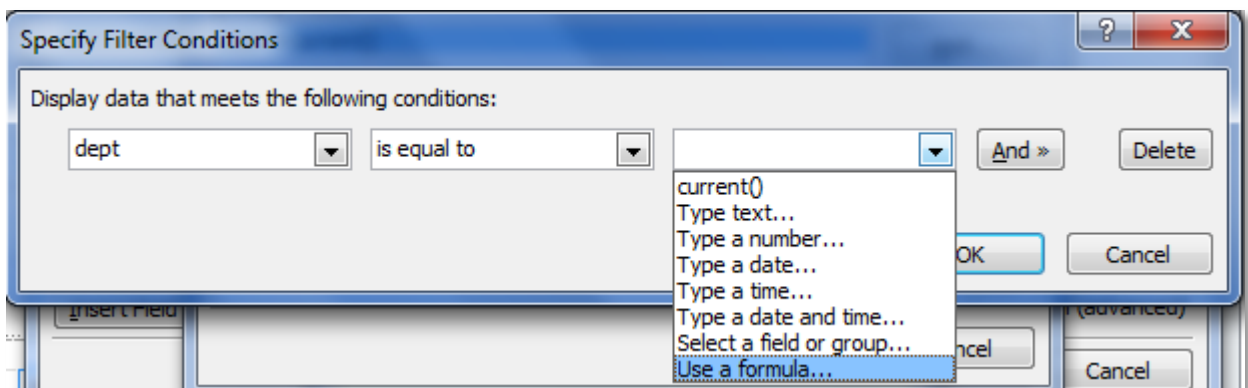
Qdabra DBXL Training: Module 12

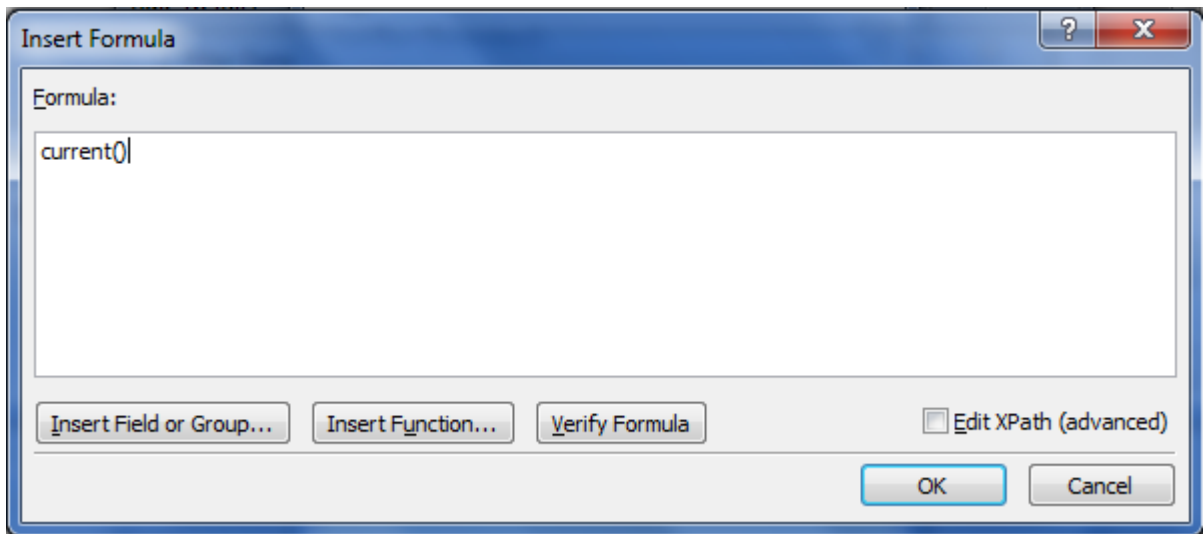


- i. In the first dropdown box, select the **dept** field.

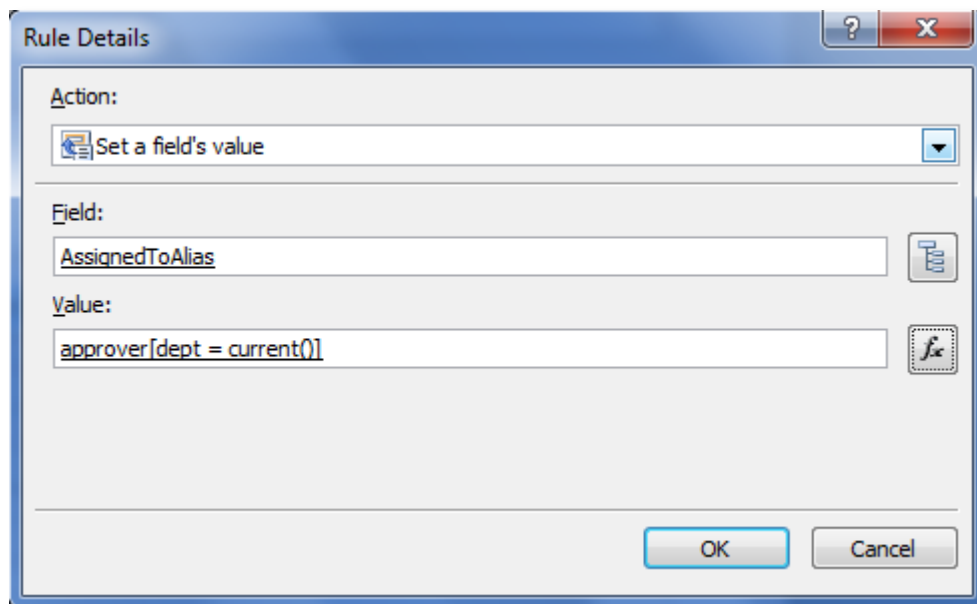


- j. In the second dropdown, select **is equal to**.
- k. In the third dropdown, select **Use a Formula** and then enter **current()**. Click **OK** four times.





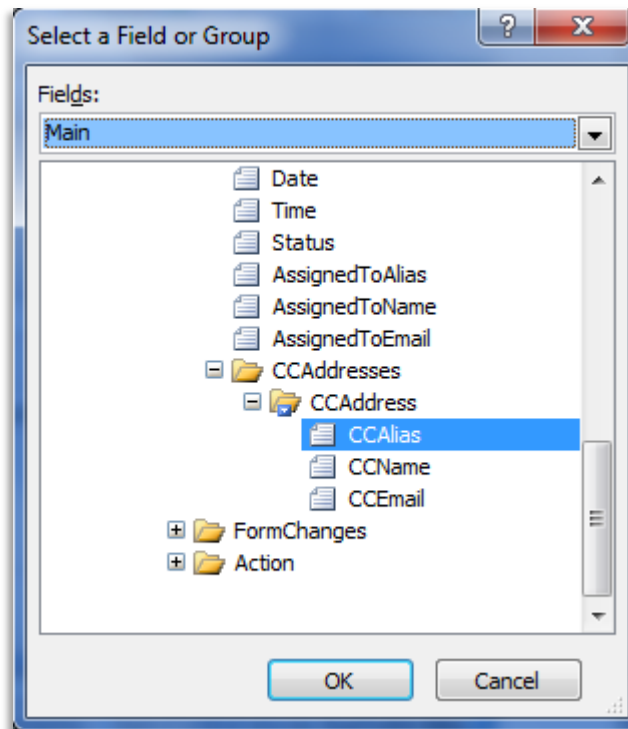
- l. Your rule should look like this:



- m. Next to **Run these actions**, click **Add > Set a field's value**.
- n. Click the **Modify** button  to set the value for **Field**.

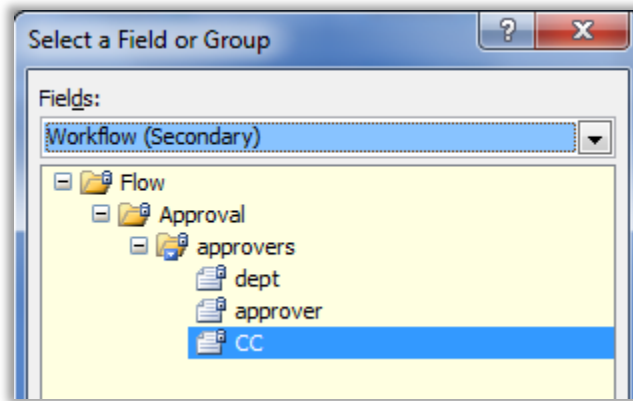
Qdabra DBXL Training: Module 12

- o. Double click on **CCAlias** node under **qdxtp_Flow**
(/my:expenseReport/my:qdxtp_Flow/my:Content/my:myFields/my:CCAddresses/my:CCAd
dress/my:CCAlias).

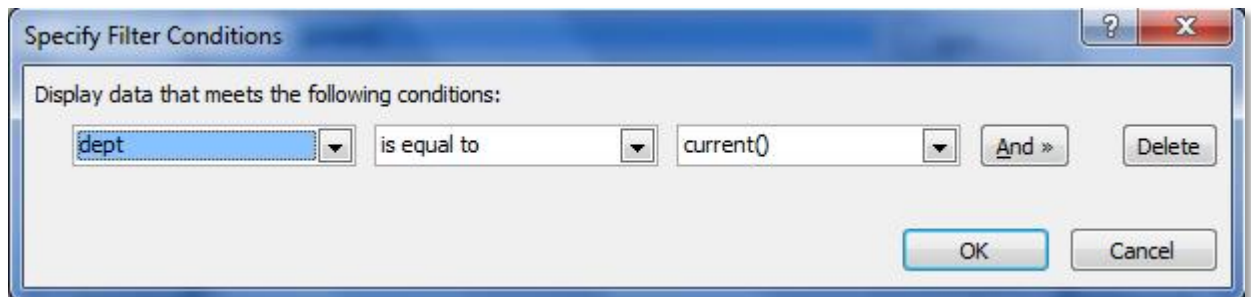


- p. Click the **fx** button  next to the **Value** field.
- q. Click **Insert Field or Group**. Click on the data source dropdown and choose the **Workflow** secondary data connection. Double click the **CC** field.

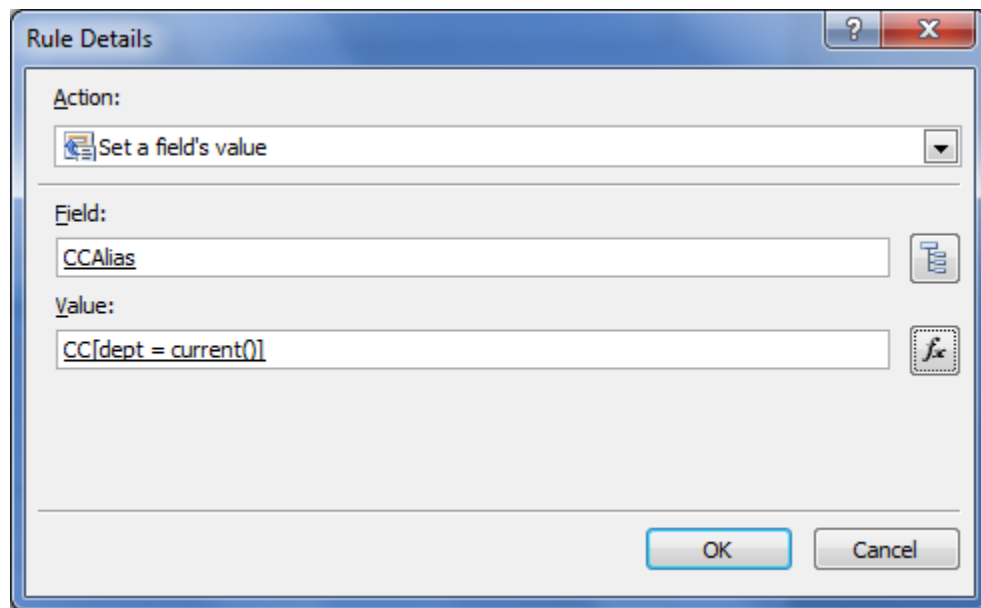
Qdabra DBXL Training: Module 12



- r. Click **Filter Data**. Click **Add**. In the first dropdown box, select the **dept** field.
- s. In the second dropdown, select **is equal to**.
- t. In the third dropdown, select **Use a Formula** and then enter **current()**. Click **OK** four times.

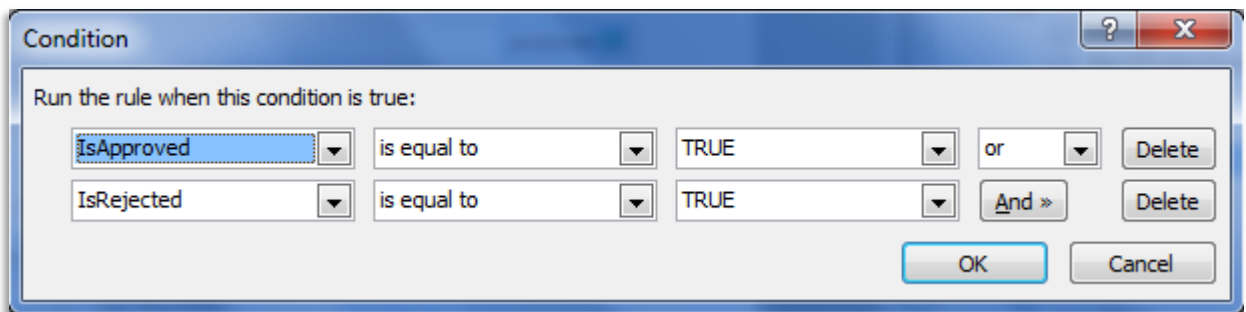


- u. Your rule should look like this:

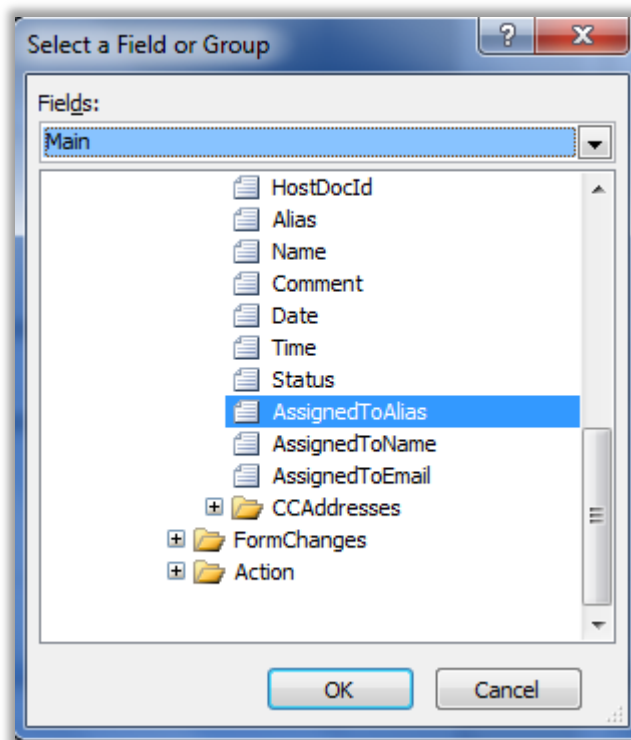


25. Under **Data** tab, click **Form Load**. We are going to add another set of rules to query the XML configuration file on load and set the Assigned To and CC fields based on the selected department.
 - a. Click **New > Action**.
 - b. Enter a name for this rule, such as "Query Assignment OnLoad".
 - c. Add the following conditions:
 - i. **IsApproved**
 (/my:expenseReport/my:qdxtp_Flow/my:Content/my:Action/my:IsApproved) =
 TRUE

or
 - ii. **IsRejected**
 (/my:expenseReport/my:qdxtp_Flow/my:Content/my:Action/my:IsRejected) =
 TRUE



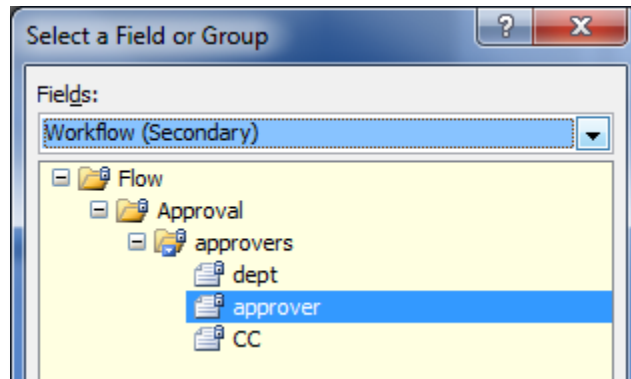
- d. Next to **Run these actions**, click **Add > Set a field's value**.
- e. Click the **Modify** button  to set the value for **Field**.
- f. Double click on **AssignedToAlias** node under **qdxtp_Flow**
(/my:expenseReport/my:qdxtp_Flow/my:Content/my:myFields/my:AssignedToAlias).



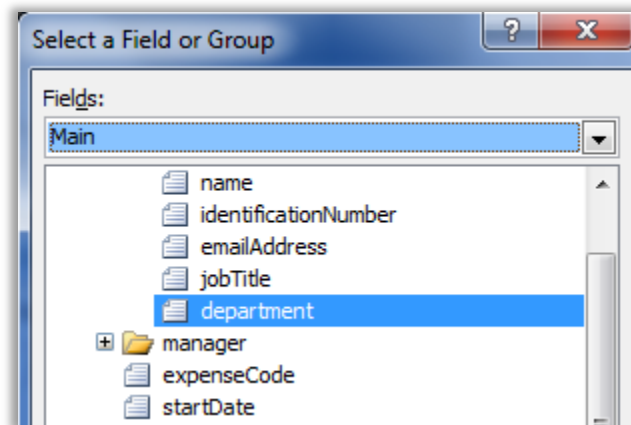
- g. Click the **fx** button  next to the **Value** field.

Qdabra DBXL Training: Module 12

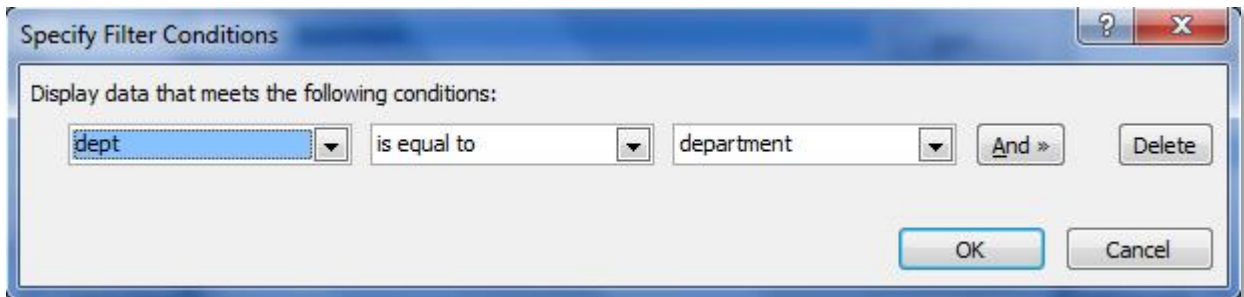
- h. Click **Insert Field or Group**. Click on the data source dropdown and choose the **Workflow** secondary data connection. Click the **approver** field.



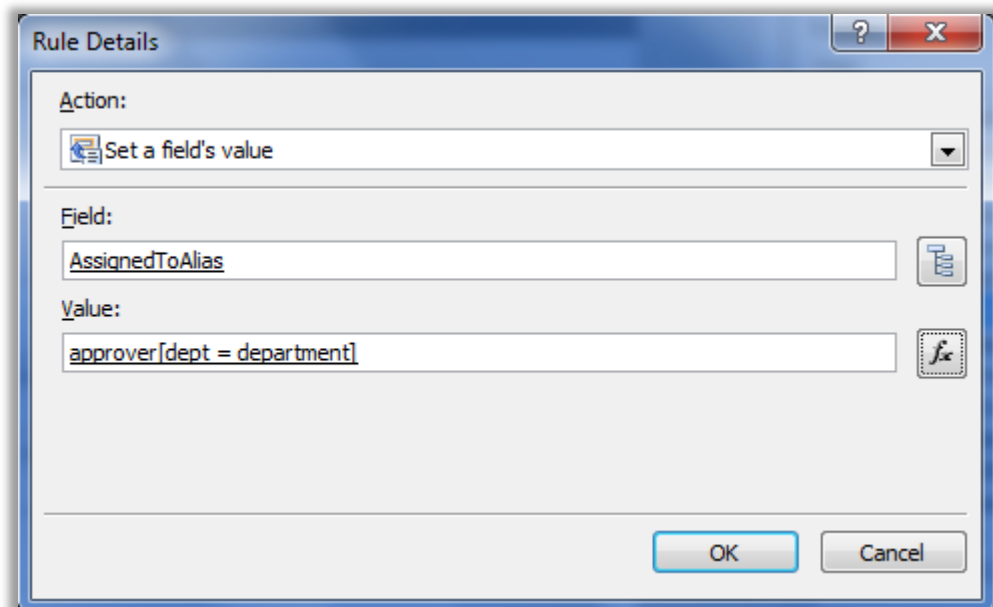
- i. Click **Filter Data**. Click **Add**. In the first dropdown box, select the **dept** field.
- j. In the second dropdown, select **is equal to**.
- k. In the third dropdown, click **Select a Field or Group**. Double-click on the department field in the main data source (/my:expenseReport/my:employee/my:department).



- l. Click **OK**. You'll now have:

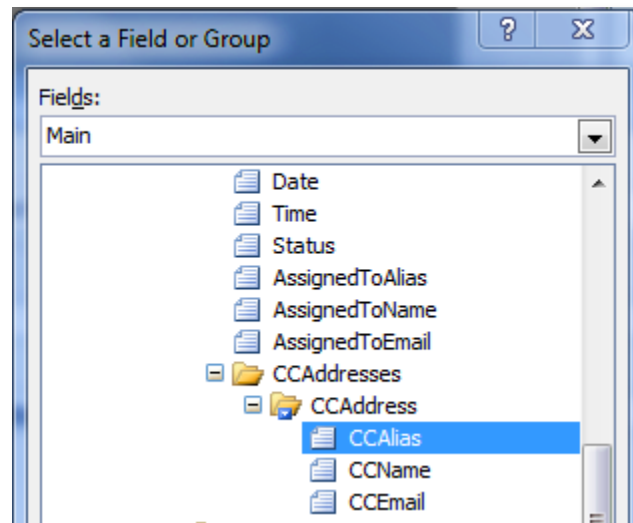


- m. Click **OK** four times. Your rule should look like this:

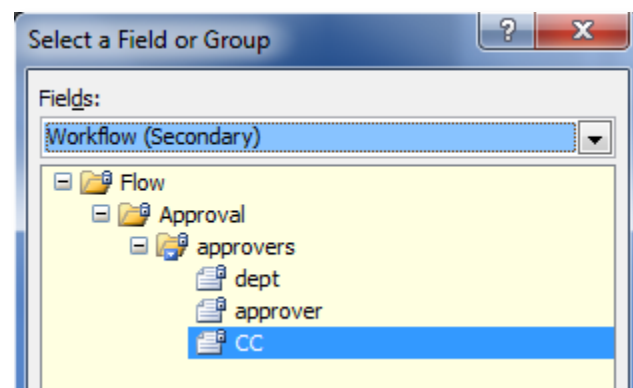


- n. Next to **Run these actions**, click **Add > Set a field's value**.
- o. Click the **Modify** button  to set the value for **Field**.
- p. Double click on **CCAlias** node under **qdxtp_Flow**
(/my:expenseReport/my:qdxtp_Flow/my:Content/my:myFields/my:CCAddresses/my:CCAd
dress/my:CCAlias).

Qdabra DBXL Training: Module 12

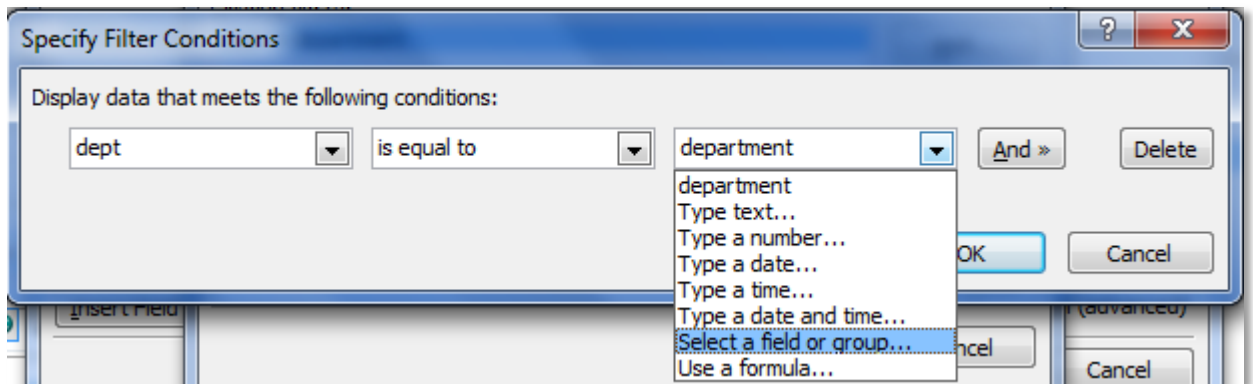


- q. Click the **fx** button  next to the **Value** field.
- r. Click **Insert Field or Group**. Click on the data source dropdown and choose the **Workflow** secondary data connection. Click the **CC** field.

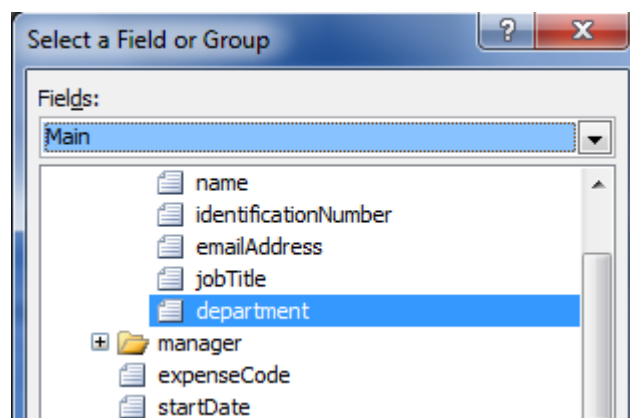


- s. Click **Filter Data**. Click **Add**. In the first dropdown box, select the **dept** field.
- t. In the second dropdown, select **is equal to**.
- u. In the third dropdown, click **Select a Field or Group**.

Qdabra DBXL Training: Module 12

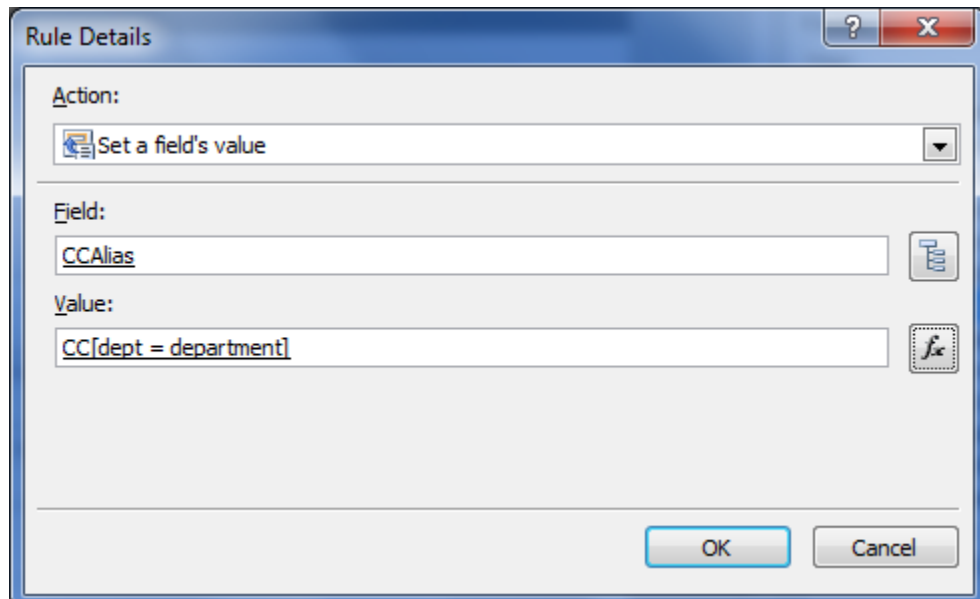


- v. Double-click on the department field in the main data source (/my:expenseReport/my:employee/my:department).



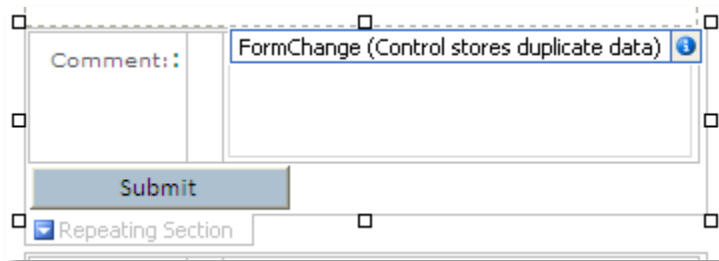
- w. Click **OK** five times.

Your rule should look like this:



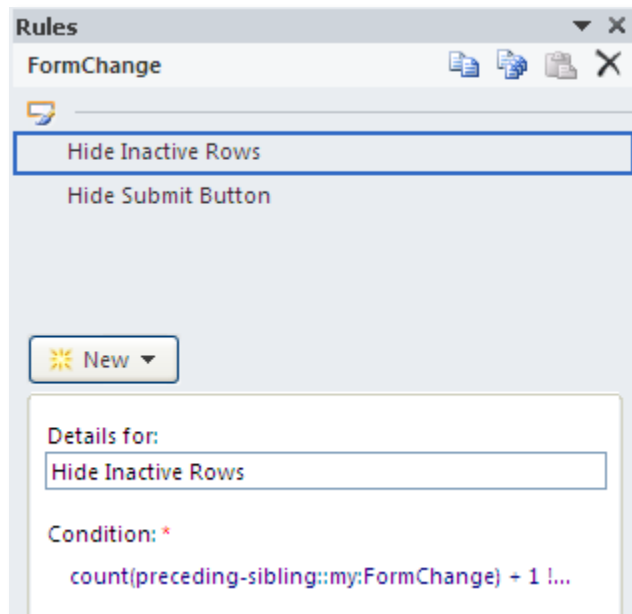
MODIFY THE TEMPLATE PART SECTION

26. Select the Repeating Section in the template part that contains the Submit button.



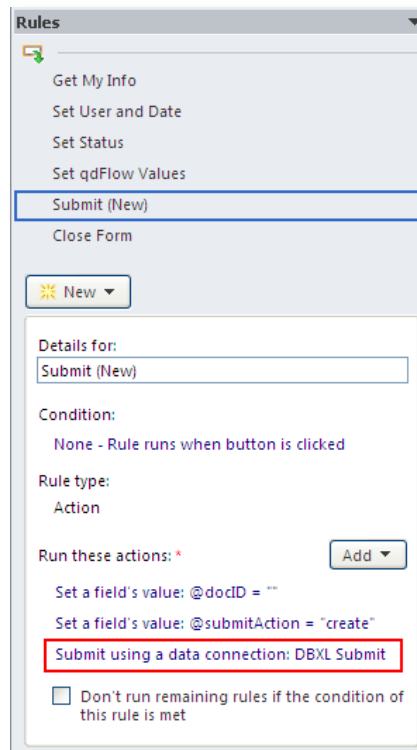
27. In the Rules task pane, modify the conditional formatting named **Hide Inactive Rows** replacing the formula with this: **count(preceding-sibling::my:FormChange) + 1 != ../my:ActiveRow**.

Qdabra DBXL Training: Module 12



28. Repeat for the Repeating Section that contains Reject and Approve buttons.
29. Select the Submit button.
30. Modify the rule named **Submit (New)** adding one last action that submits data using the DBXL web service data connection.

Qdabra DBXL Training: Module 12



31. Select the Approve button and modify the rule named Submit (Approve). Add one last action that submits data using the DBXL web service data connection.
32. Select the Reject button and modify the rule named Submit (Reject). Add one last action that submits data using the DBXL web service data connection.
33. Delete the red guidance text in the XTP. Those instructions are the same as detailed here.

Note: Recall that you already have a submit button in the form. Ideally, you will copy the rules from the existing button into this one, and delete the extra button. Be careful when doing so to make sure you don't end up with duplicate actions or with the actions out of order.

34. Save your changes to the form template.
35. Attach this latest form template to your DBXL document type.

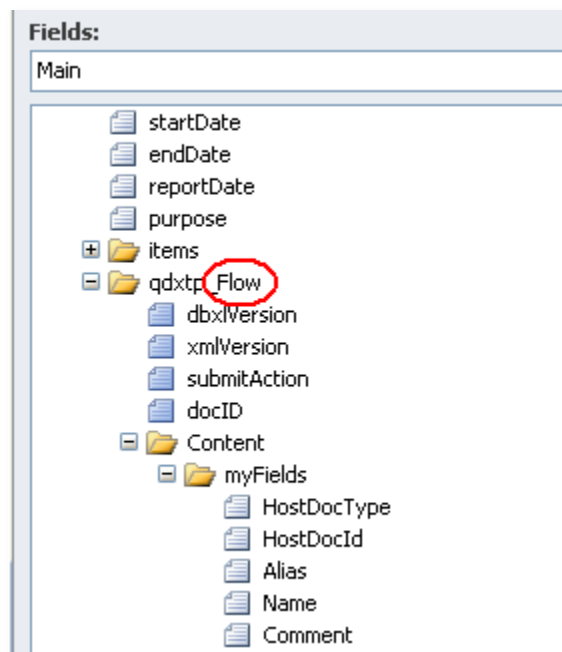
CREATE THE XML-TO-XML MAPPING

XML-to-XML Mapping is a feature in DBXL v2.6 which allows you to map an XML fragment of a form to a document type. We have previously added an XTP that is configured for XML-to-XML. We want this XML to

Qdabra DBXL Training: Module 12

submit to a system document type called EmailFlow. In order to do that, we have to create a QdSubDocumentMapping.

36. In DAT, open a new QdSubDocumentMapping form. This can be found under system document types.
37. Fill it out:
 - a. Name – the name should be the XTP's root node, that is the entire string following *qdxtp*. In this case, we're using **Flow**.



- b. Target Document Type – select QdEmailFlow from the dropdown.
- c. Host Field Mappings
 - i. Click **Insert Item**
 - ii. Enter **/my:myFields/my:HostDocId** which is the full XPath to the HostDocId field in the QdEmailFlow system document type.
 - iii. Select **DocId** in the *Token* dropdown.

Your mapping should look similar to this:

Sub Document Mapping

Mapping Name

Please specify a name for your mapping. This mapping will be applied any time a node with the name **qdxtp_mappingName** is located in a submitted document.

Flow

Target Document Type

Please select the document type to which the mapped documents should be submitted.

QdEmailFlow

Namespace

Below are the namespace URI and prefix of the document type to which mapped documents will be submitted. These should be populated automatically when you select a Target Document Type

Target namespace URI

http://schemas.microsoft.com/office/infopath/2003/myXSD/2006-12-05T15:51:37

Prefix

my

Host Field Mappings

Please enter the XML of any locations in the mapped documents to which you will map metadata values from the host document. For example, to have the host document's DocId injected into the **/my:myFields/my:HostDocId** field of the child documents, enter **/my:myFields/my:HostDocId** for the Field Path, and select DocId for the token. The Field Path is case sensitive.

Field Path	Token
/my:myFields/my:HostDocId	DocId

☒ Insert item

38. Click Save – form will submit to the QdSubDocumentMapping document type and will close.

VERIFY THE SAMPLE FORM SOLUTION

39. Go back to the SharePoint form library for this form.
40. Click **Add Document**.
41. Fill out the form. Select a Department - you will notice that the **Assigned To** and **CC** fields in the Approve Process section at the bottom are automatically populated.
42. Add a comment. When done, click the **Submit** button.

Qdabra DBXL Training: Module 12

The screenshot shows a web form titled "APPROVE PROCESS". It has two main sections. The top section is for assigning the task, with "Assigned To:" showing "Ernesto Machado" and "Jo-Anne Alfafara" with red 'x' icons, and a "CC:" field with an "Add CC" button. The bottom section is for adding a comment, with "Comment::" and a text area containing "Submitted. Please review.". Below the comment field is a blue "Submit" button.

43. Click **OK** in the confirmation dialog. The form will automatically close. If configured, email notifications will be sent out to the specified approver and CC email addresses.
44. In the Documents view, re-open the submitted form. As the manager, you can now Approve or Reject the form. There is no need to type the alias of the person who submitted the form. Add a comment and try clicking **Reject**.

This screenshot shows the same "APPROVE PROCESS" form, but now it has two buttons at the bottom: "Reject" and "Approve". The "Comment::" text area now contains "This is rejected.". The "Reject" button is highlighted with a red rectangular border.

45. The **Assigned To** field will auto-fill with the alias of the user who originally submitted the form. The person who owns the expense report should get a notification email. Click **OK** on the confirmation dialog. The form will automatically close.
46. As the form submitter, re-open the rejected Expense Report form and make changes. Add a comment, and then click **Submit**. Email notifications will be sent to the appropriate approver and CC email addresses.

APPROVE PROCESS

Assigned To:

✖ Ernesto Machado

✖ Jo-Anne Alfafara

CC:

☑ Add CC

Comment:

Expenses modified. Please review the changes.

Submit

47. As the manager/approver of the form, re-open the submitted expense report. Add a comment and click **Approve**.

APPROVE PROCESS

Assigned To:

✖ Ernesto Machado

✖ Jo-Anne Alfafara

CC:

☑ Add CC

Comment:

Looks good. Approved.

Reject

Approve

48. Again, the **Assigned To** field will auto-fill with the alias of the user who originally submitted the form. Thus, the submitter will be notified that the expense report has been approved. Click **OK** in the confirmation dialog box.
49. In addition, the XTP comes with a **Form Change History** section which shows you the previous changes made in the form.

Form Change History:

Date / Time	User Alias	Status	Comments
12/7/2011 2:31:55 PM	jo-anne	Approved	Looks good. Approved.
12/7/2011 2:29:28 PM	jo-anne	Submitted	Expenses modified. Please review the changes.
12/7/2011 2:24:02 PM	jo-anne	Rejected	This is rejected.
12/7/2011 2:19:55 PM	jo-anne	Submitted	Submitted. Please review.

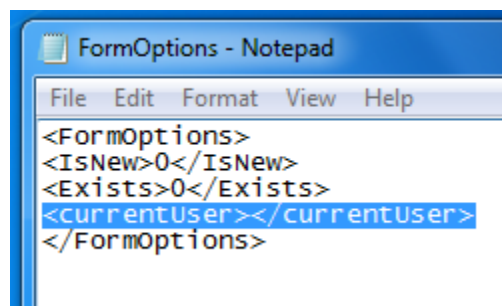
BONUS

You might notice that the buttons (Submit, Reject, Approve) show for all users. We can add conditional formatting to fix this.

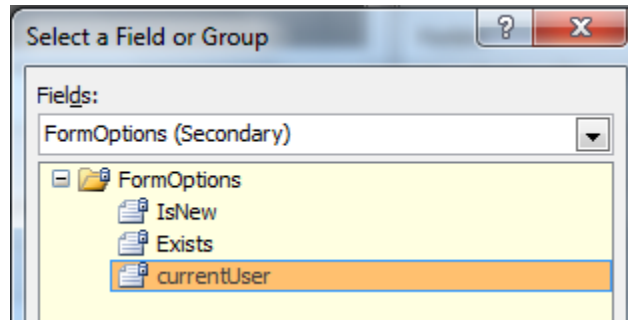
From a previous lab you'll remember that we are storing the submitter's alias in the following xPath:
/my:expenseReport/my:employee/my:employeeAlias

In the InfoPath form, we can hide the Approve and Reject button for this user. But first we need to detect the currently logged in user.

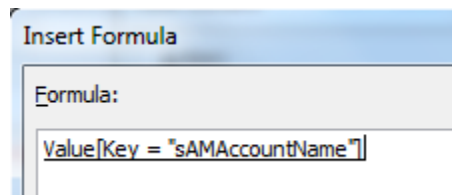
50. Go to **Data > Resource Files**.
51. Select **FormOptions** and click **Export**. Save the file as FormOptions.xml.
52. Open the file in Notepad.
53. Add a node, as shown below: <currentUser></currentUser>



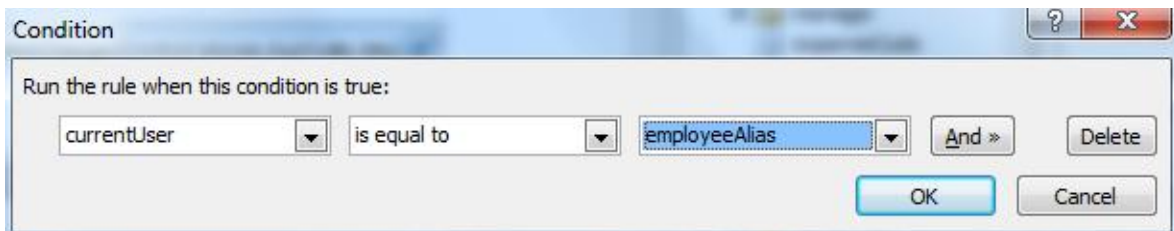
54. Save the file and close.
55. Back in InfoPath Designer, go to **Data > Data Connections**.
56. Select the **FormOptions** data connection, click **Modify**. Click **Browse** and select the FormOptions.xml file that you just edited and saved.
57. Click **Next**, **Finish** and **Close**.
58. Go to **Data > Form Load**.
59. Click **New > Action** and call this rule "Determine current user".
60. Move the rule up so that it executes first on load.
61. Click **Add > Query for data**.
62. Select **GetMyInfo** and click **OK**.
63. Select **Add > Set a field's value**.
64. For **Field**, select the **currentUser** node we added earlier.



65. For **Value**, select the user's **sAMAccountName** (alias) from the **GetMyInfo** data connection. If you don't remember how to do this, please refer to a previous lab.



66. Select the **Reject** button.
67. Select **New > Formatting**.
68. For the condition, we want to check whether the **currentUser** is equal to the **employeeAlias**.



69. Check the checkbox to **Hide this control**.

The steps above have ensured that, when the employee opens the form, s/he will NEVER see the Reject button.

70. Repeat the same steps (66-69) for the **Approve** button.

Once you have determined the currentUser, on load, you can hide or disable sections, buttons and controls, based on whether the currentUser is the employee or the manager or a member of the workflow. You can extend the example above to implement additional formatting, as desired. For example: you could hide the Submit button if the /my:expenseReport/my:qdxtp_Flow/my:Content/my:Action/my:FormStatus node is equal to "Approved" and the currentUser is the original submitter (employee). This means that the employee cannot resubmit the request after it has been approved.