

CONTROL SAVE AND SUBMIT IN YOUR FORM

PRODUCT: qRules v2.3
LAST UPDATED: September 23, 2010

SCENARIO

As a forms developer or administrator, you wish to have control over the Save and Submit actions carried out by your users.

VALUE PROPOSITION

qRules allows you to dynamically control Save and Submit in your InfoPath forms. Using the command ChangeSubmitUrl you can modify the Submit location of your SharePoint submit data connection. In the tutorial below, the submit location will change according to a selection made in a dropdown.

Using SetSave you can enable or disable InfoPath's built in Save and Save As actions. In the tutorial below, we will disable and enable Save depending on the user's role.

REQUIREMENTS

- **InfoPath:** qRules requires InfoPath 2007 or 2010.
- **SharePoint:**
 - SetSave does not require SharePoint.
 - For the tutorial included in this document, ChangeSubmitUrl does require SharePoint 2007 or 2010.
- **qRules:**
 - ChangeSubmitURL was first introduced in qRules v2.0.
 - Set Save is a new feature in qRules v2.3
 - SetSave is not compatible with browser-enabled forms.



CHANGESUBMITURL

Introduction

The command ChangeSubmitUrl modifies the submit URL in data connection that was created using InfoPath's Data Connection Wizard. For most users this means a SharePoint submit data connection. The command ChangeSubmitUrl, therefore, would allow the InfoPath form to submit to a different SharePoint Url.

Technical details: The command changes the FolderUrl property of a FileSubmitConnection. MSDN tells us that: *"The FileSubmitConnection class represents a data connection created in InfoPath using the Data Connection Wizard that specifies all the information necessary for submitting data to a Microsoft Windows SharePoint Services document library or other server that supports DAV connections."* The command could potentially be used without SharePoint, but this option has not been tested or documented.

The command follows the logic below:

- If the command provides a /url param, the command uses that first.
- If there is no /url param, the command checks to see if a source was obtained on open.
 - If no source was obtained, the command tries to get the href from the processing instructions. If all that fails, it returns an error.

Tutorial

The steps below will guide you through the implementation of the ChangeSubmitUrl command. We will design a simple for that submits to two different SharePoint Urls based on the selection made in a dropdown control. After completing this tutorial, you may try to extrapolate to adapt the command to your scenario.

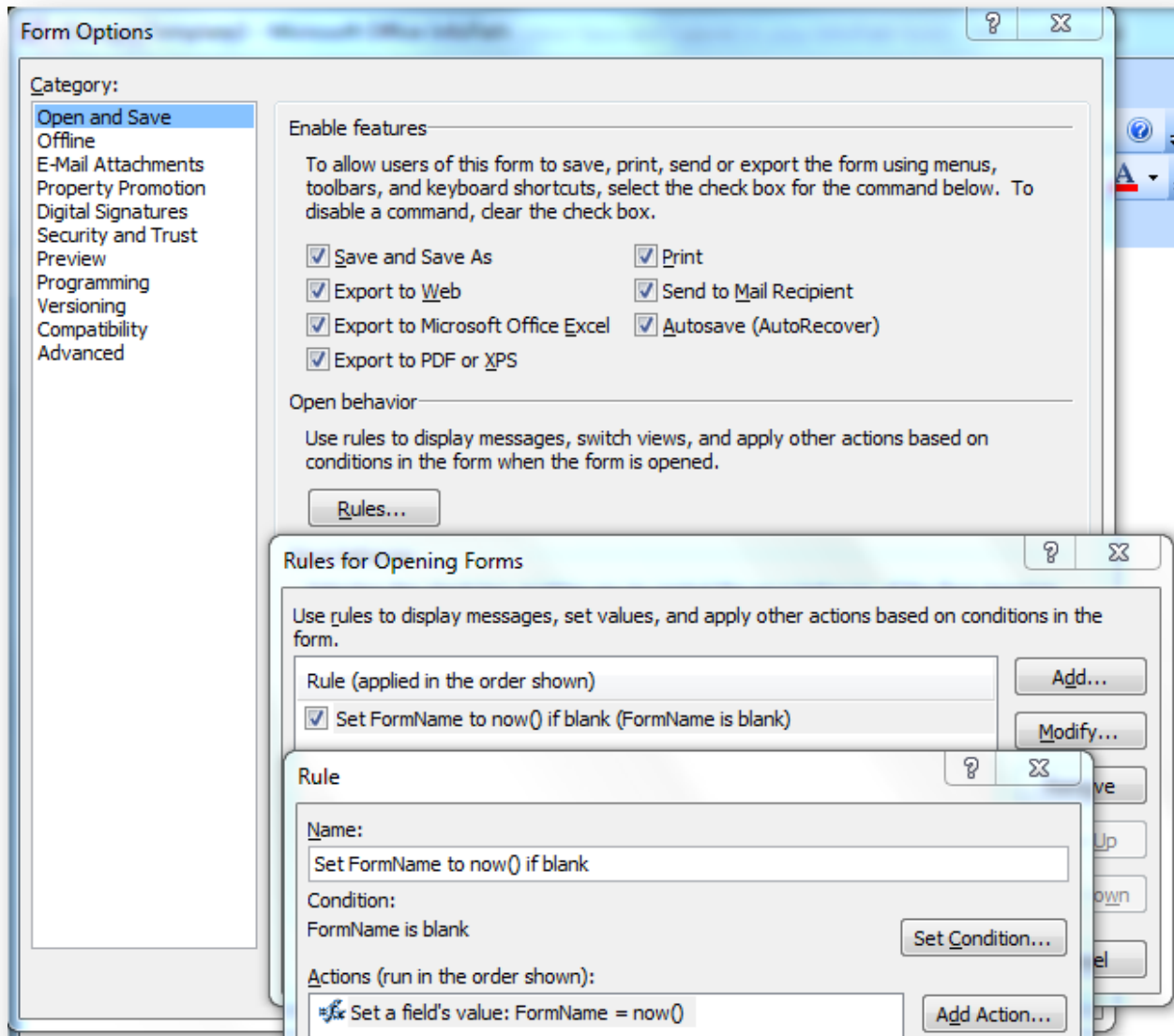
- A. Create a new InfoPath form
 1. In InfoPath, design a new, blank form.
 2. Go to **Tools > Form Options > Compatability** and check the checkbox to **Design a form template that can be opened in a browser or InfoPath.**
 3. Select **File > Save As** and save the XSN to your local drive.
 4. Close the form.
- B. Inject qRules
 1. Launch the qRules Injector.
 2. Browse to select the XSN you created in A.
 3. Click **Inject.**
 4. Close the qRules Injector.
 5. Open the form in Design mode.

Steps A and B are also covered by the qRules User Guide. For more details, please refer to that document.

In order to submit to SharePoint we need a unique form name. There are countless ways to accomplish this. Step C below is just one possibility.

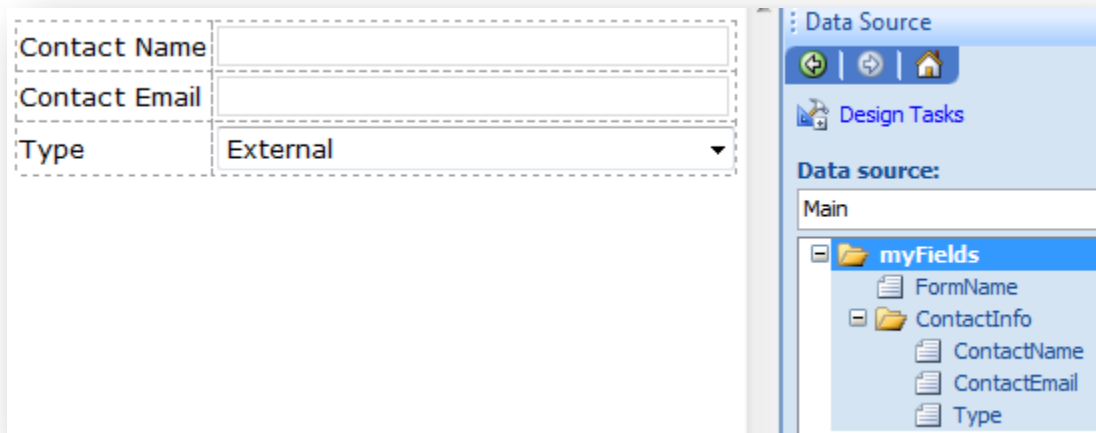


- C. Create a unique name for your form.
 1. Go to **Tools > Form Options**.
 2. Click on **Rules** and then click **Add**.
 3. Add an action that sets the **FormName** field to the function **now()**.
 4. Add a condition such that this rule only executes if the **FormName** field is blank.

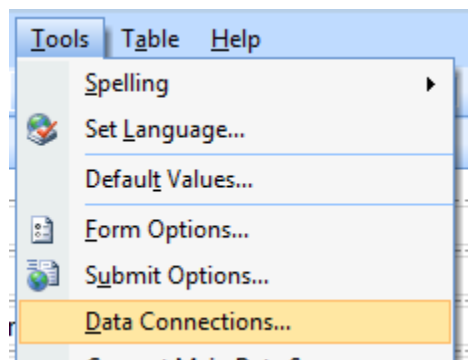


5. Click OK to close all dialogs.
- D. Add some controls to your canvas. Because there are countless ways to accomplish this, this tutorial will not enforce a specific design or schema. For the purposes of this example, we have designed the form as seen in the screenshot below. The key is that our dropdown, called Type in this case, holds two options: 'External' and 'Internal'. The selection made in this dropdown will later determine to which SharePoint URL the form is submitted.

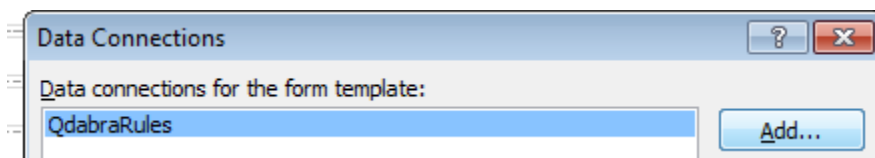




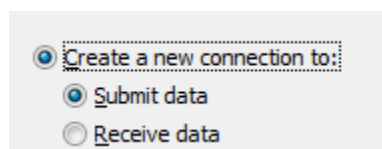
- E. Optionally, you may want to promote properties (via **Tools > Form Options > Property Promotion**).
- F. Create a submit data connection
 1. Go to **Tools > Data Connections...**



2. Click on the **"Add"** button.

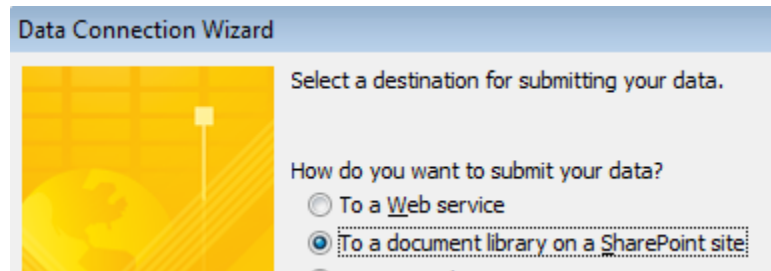


3. In the **"Data Connection Wizard"**, click **"Create a new connection to"**, **"Submit data"** and click **Next**.

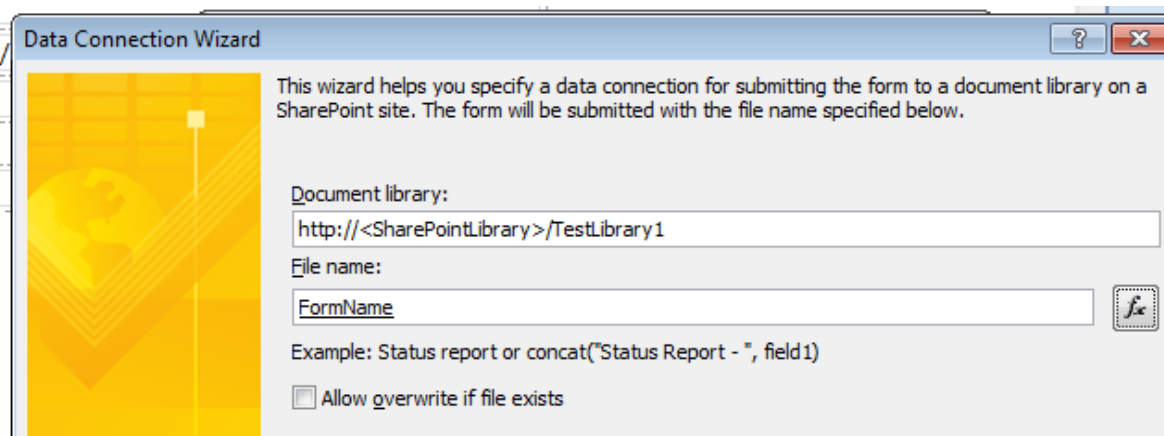


4. Select **"To a document library on a SharePoint site"** to submit your data and click **"Next"**.

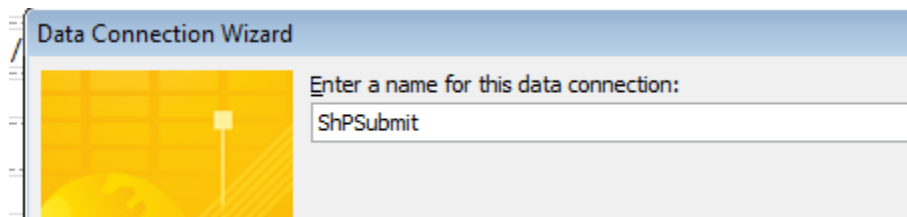




5. Enter the “**Document library**” location on a Share Point site (ex: ***http://<SharePointSite>/External***) and enter a “**File name**” and click “**Next**”.

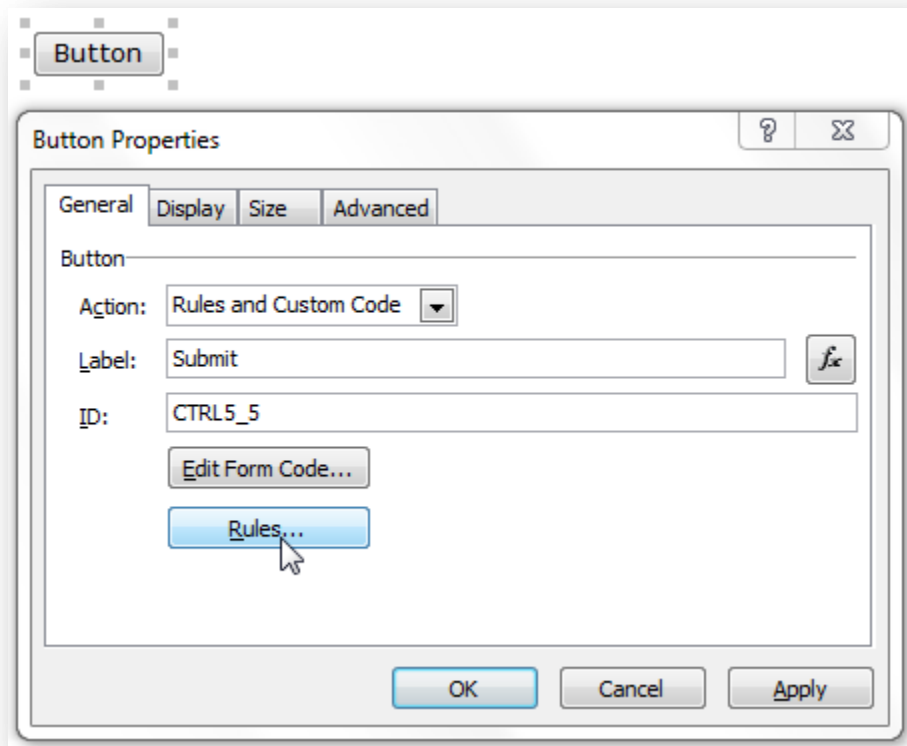


6. Enter a name for this data connection. For this tutorial we named it “**ShPSubmit**”. Click “**Finish**”.

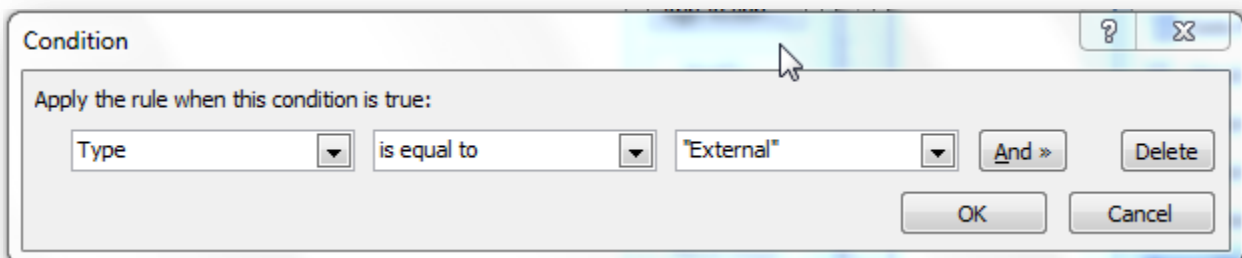


- G. Add and configure a submit button
 1. Add a button to your canvas, double click on it and change the label to ‘Submit’.



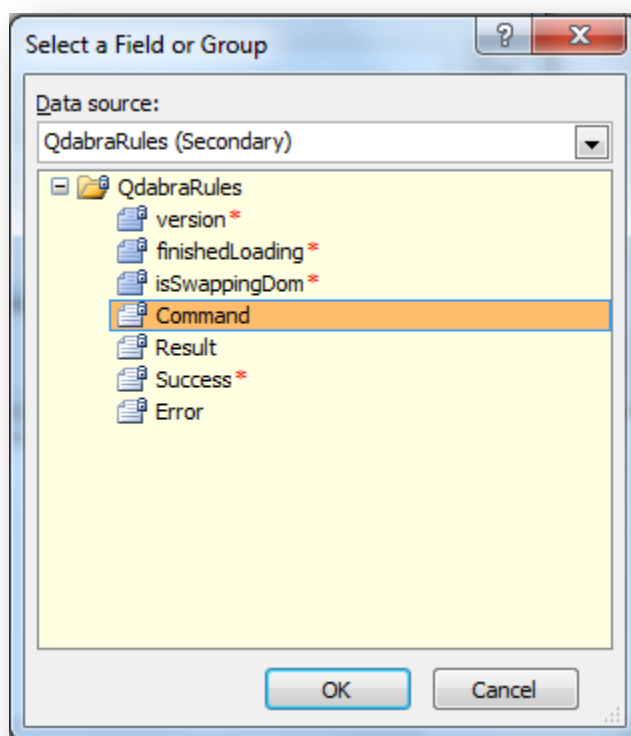


2. Click on **Rules**, and then click **Add**.
3. Click on **Set Condition** and add a condition such that the rule only executes when **Type** equals 'External'.



4. Click on **Add Action** and choose **Set a field's value** from the Actions dropdown.
5. For the **Field**, select the **Command** node from the **QdabraRules** secondary data source.





6. For the **Value**, enter ***ChangeSubmitUrl /submit=ShPSubmit /url=http://<SharePointSite>/External*** and click **OK**.
 7. Click again on **Add Action**.
 8. Select **Submit using a data connection** action and select the **ShPSubmit** data connection. Click **OK**.
 9. Repeat steps 3 through 8, but use 'Internal' for the condition and ***ChangeSubmitUrl /submit=ShPSubmit /url=http://<SharePointSite>/Internal*** for the **Value**.
 10. Click OK to close all dialogs.
- H. Finish and Publish the form
1. Drag and drop the QdabraRules **Result**, **Success**, and **Error** fields into the view. If there is an error, these fields will give us the information necessary to troubleshoot.
 2. Save the form.
 3. Publish the form to 2 different SharePoint libraries. In this tutorial let's assume we published the form to <http://<SharePointSite>/External> and <http://<SharePointSite>/Internal>. Please refer to the qRules User Guide (section: **Use qRules in InfoPath Forms Services**) for detailed steps on how to publish the form to SharePoint.

You can test the solution by opening the form, in the browser, from either SharePoint URL. You'll notice that the form is "smart" and will submit to the correct SharePoint URL according to the selection made in the dropdown.



Save | Save As... | Close | Print View

Contact Name Mike External Smith

Contact Email mes@acme.com

Type External

Submit

☒ Success

Result:

http://dbxl-test3/External

Error:

Save | Save As... | Close | Print View

Save | Save As... | Close | Print View

Contact Name Mike Internal Smith

Contact Email mis@acme.com

Type Internal

Submit

☒ Success

Result:

http://dbxl-test3/Internal

Error:

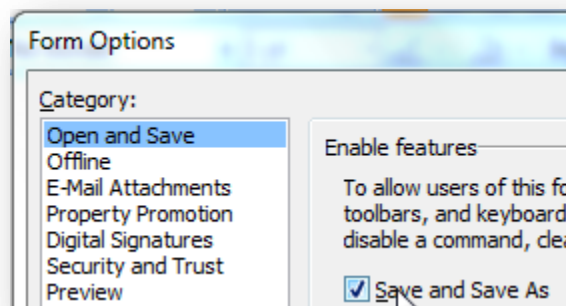
Save | Save As... | Close | Print View



SETSAVE

Introduction

SetSave is a powerful command that allows you to manipulate the 'Save' option available in your InfoPath forms. The form designer will usually go to **Tools > Form Options > Open and Save** to enable or disable **Save** and **Save As**. The SetSave command allows that option to be toggled on the fly. Not only that, but the command allows us to toggle **Save** and **Save As** separately.



The command also allows us to:

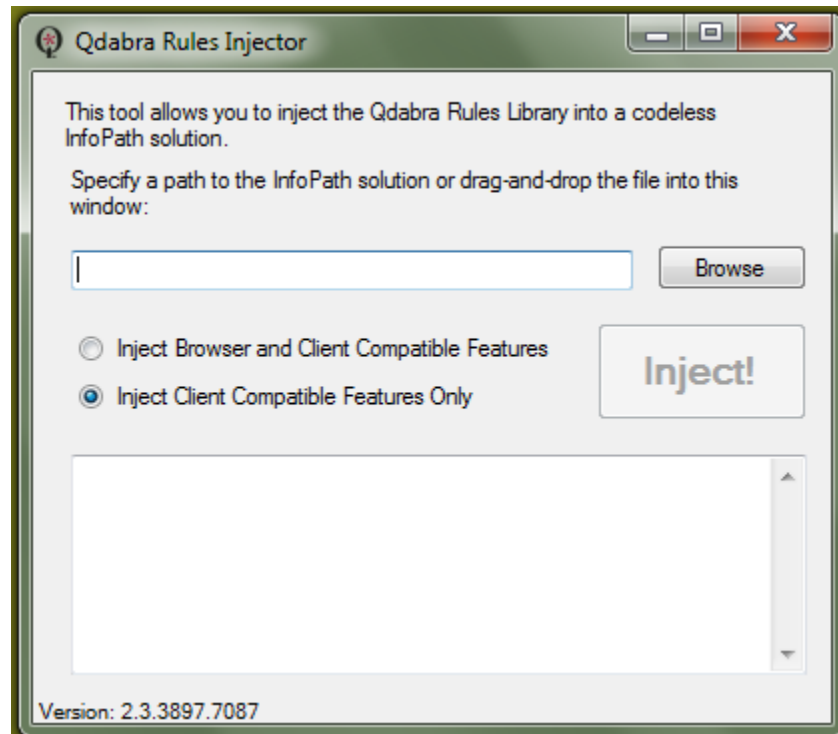
- Block all save operations
- Conditionally prevent **Save** when the save location (path) begins with the value indicated in the parameter /saveloc
- Provide a guidance message to users when **Save** or **Save As** are not allowed.
- Suggest a location in the **Save As** dialog.
- Suggest a filename in the **Save As** dialog.

Command arguments:

- **save**: Allow or prevent the Save action (allow | block)
- **savemsg**: Message to display if Save is blocked (* to clear)
- **saveloc**: Block Save if location starts with this (* to clear)
- **saveas**: Allow or prevent the Save As action (allow | block)
- **saveasmsg**: Message to display if Save As is blocked (* to clear)
- **saveasloc**: Suggested location in Save As dialog (* to clear)
- **saveasfile**: Suggested filename in Save As dialog (* to clear)

Note that this command is NOT available in browser-enabled forms. When injecting, you must select the "Client" option as seen below.





Tutorial

This tutorial will illustrate the use of the SetSave command. After executing these steps, you'll be prepared to adapt this command into your forms.

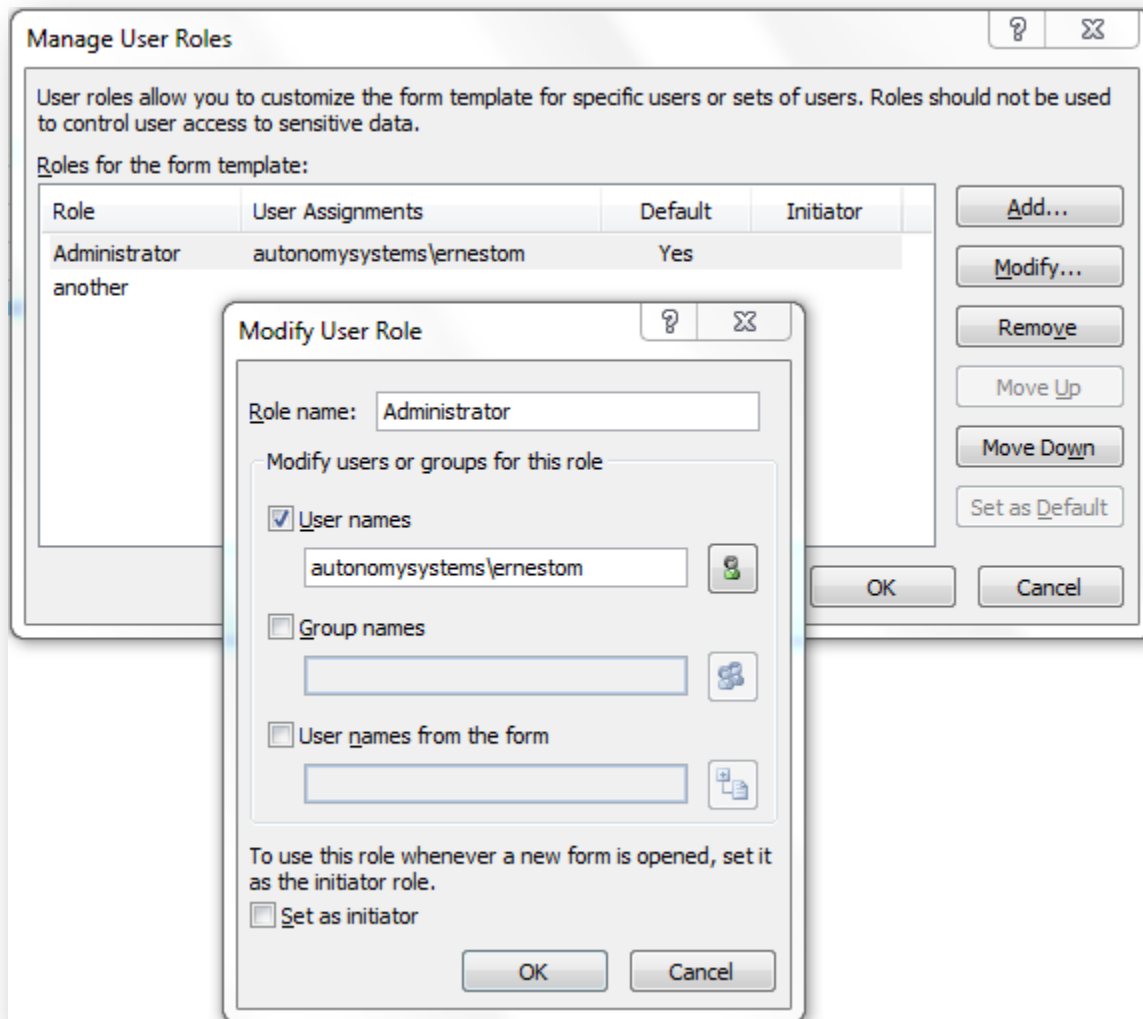
- A. Create a new InfoPath form
 - 1. In InfoPath, design a new, blank form.
 - 2. Select **File > Save As** and save the XSN to your local drive.
 - 3. Close the form.
- B. Inject qRules
 - 1. Launch the qRules Injector.
 - 2. Browse to select the XSN you created in A.
 - 3. Select the **"Inject Client Compatible Features Only"** option and click **Inject**.
 - 4. Close the qRules Injector.
 - 5. Open the form in Design mode.
 - 6. Drag and drop the QdabraRules **Result**, **Success**, and **Error** fields into the view. If there is an error, these fields will give us the information necessary to troubleshoot.

Steps A and B are also covered by the qRules User Guide. For more details, please refer to that document.

- C. By default, a new InfoPath form allows **Save** and **Save As**. Let's leave that as-is.
 - 1. Go to **Tools > Form Options**.
 - 2. Verify that **Save** and **Save As** are checked.
 - 3. Click **OK**.
- D. Add an Administrators group



1. Go to **Tools > User Roles** and click **Add**.
2. Enter a **Role name**, check **User names** and enter a user alias.
3. Click **OK**.
4. Enter another User Role, necessary for testing purposes later on.



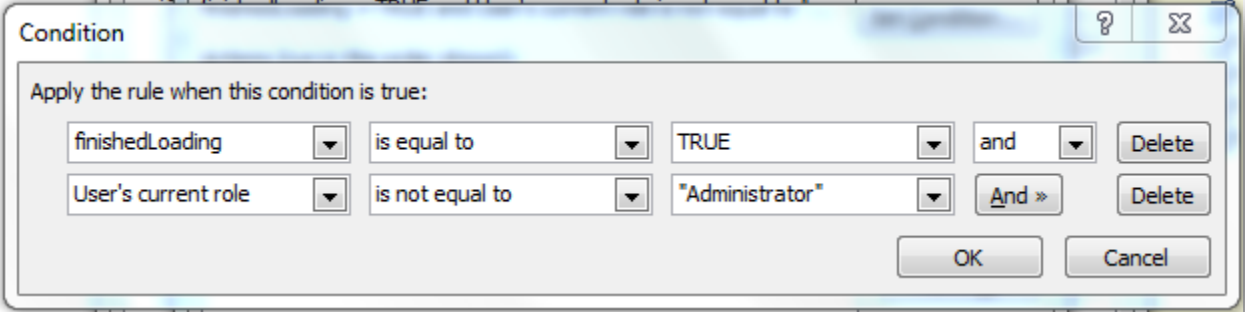
- E. Disable **Save As** for users who are not Administrators.
1. In the Data Source taskpane, switch to the QdabraRules secondary data source.
 2. Double click on the finishedLoading node.

This field is set to FALSE by default and switches to TRUE once the form has finished loading. This allows us to execute qRules commands on load.

3. Click on the **Rules and Merge** tab and click **Add**.
4. Click on **Set Condition**.
5. Select the **finishedLoading** field from the first dropdown and **'TRUE'** from the third dropdown.



6. Click **And** to add another condition.
7. Select **User's current role** from the first dropdown and '**Administrator**' from the third dropdown. In the middle dropdown, select "**is not equal to**".



Condition

Apply the rule when this condition is true:

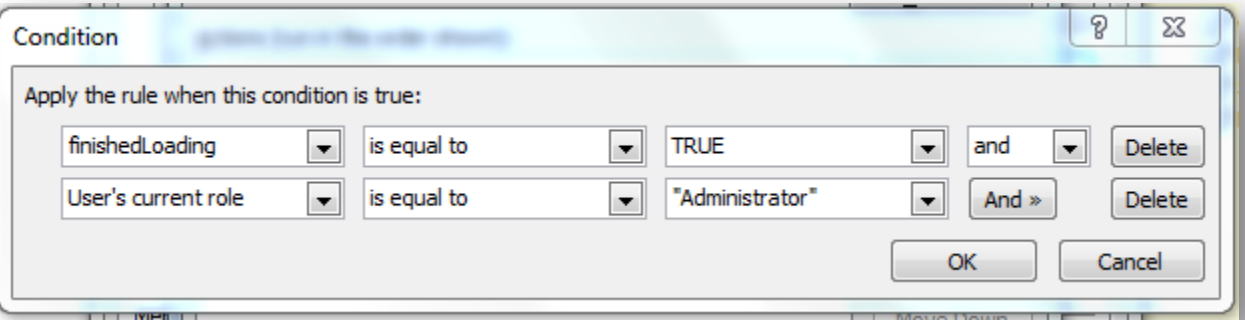
finishedLoading	is equal to	TRUE	and	Delete
User's current role	is not equal to	"Administrator"	And >>	Delete

OK Cancel

8. Click **OK**.
9. Click on **Add Action**.
10. Select **Set a field's value** from the **Action** dropdown.
11. For **Field**, select the **Command** node from the **QdabraRules** secondary data source.
12. For **Value**, enter the command

SetSave /save=block /saveas=block

13. Click **OK** twice.
- F. Suggest a location and file name for Administrators.
1. Click **Add** to create another rule.
 2. Add a condition similar to the one created for the previous rule, except now the condition catches when the user's current role is "Administrator".



Condition

Apply the rule when this condition is true:

finishedLoading	is equal to	TRUE	and	Delete
User's current role	is equal to	"Administrator"	And >>	Delete

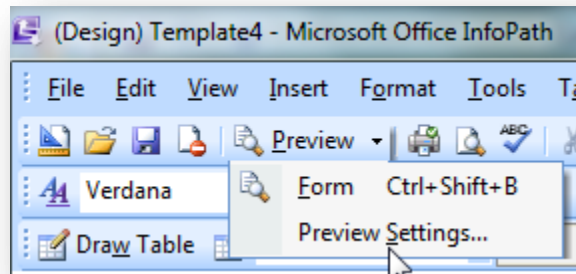
OK Cancel

3. Click **Add Action**.
4. Select **Set a field's value** from the **Action** dropdown.
5. For **Field**, select the **Command** node from the **QdabraRules** secondary data source.
6. For **Value**, enter the command

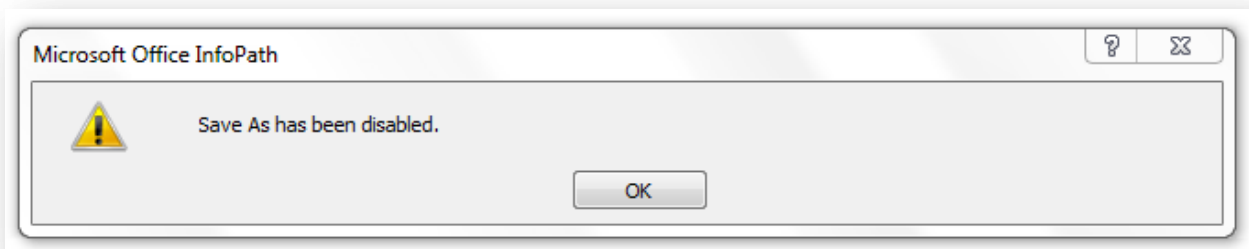


SetSave /save=block /savemsg=Save is not allowed. Please use File > Save As. /saveas=allow /saveasloc=C:\Data /saveasfile=SetSaveExample.xml

7. Click **OK** twice.
- G. Test the solution as a non-Admin
 8. Click the down arrow next to the Preview button to then select **Preview settings**.

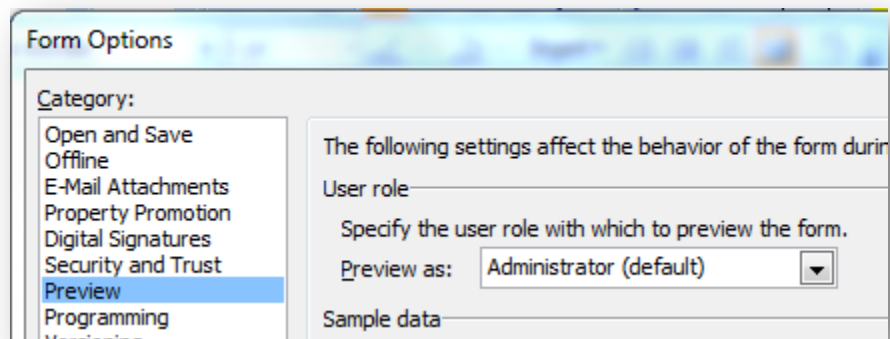


9. Select the non-admin user and click **OK**.
10. Click **Preview**.
11. Click **Save** or **File > Save As**. You'll receive the following message:

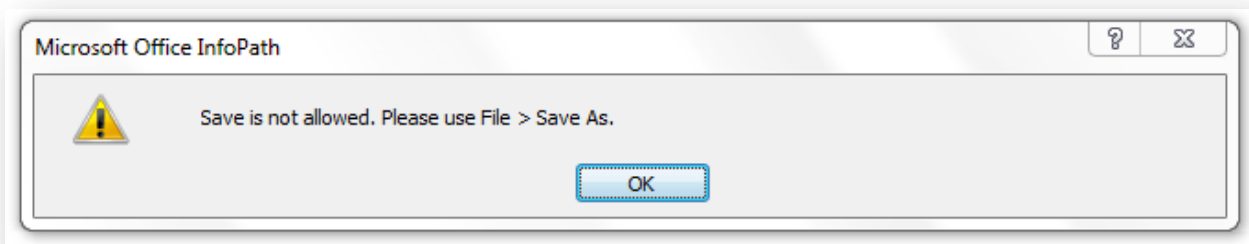


- H. Test the solution as the Administrator
 1. Click the down arrow next to the Preview button to then select **Preview settings**.
 2. Select the Administrator role and click **OK**.





3. Click **Preview**.
4. Click **Save** and save the form locally.
5. Click **Close Preview**.
6. Open the xml saved locally and click **Save**. You will see our customized message.



7. Click **OK** to dismiss that message, then click **File > Save As** to see that
 - i. **Save As** is allowed, and
 - ii. **Save As** suggests a location and a filename.



SUPPORT

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

For advanced information on the SaveAs property please read [this MSDN article](#).

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>

