

# USING QRULES WITH CUSTOM CODE

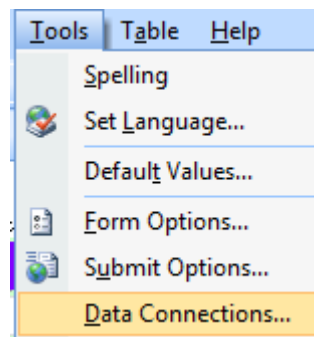
**PRODUCT:** qRules  
**LAST UPDATED:** May 7, 2014

qRules v2.2 is the first version of qRules that can co-exist with other form code. If custom code already exists in your InfoPath form, you'll need to manually add the qRules DLL instead of injecting. This document will walk you through the steps.

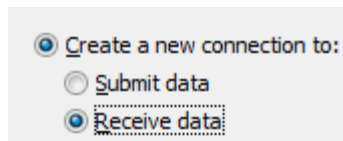
To accomplish the steps below, you will need to have a form with custom code and the correct development environment for your given version of InfoPath. You need to have **qRules version 2.2 or later**.

## INFOPATH 2007

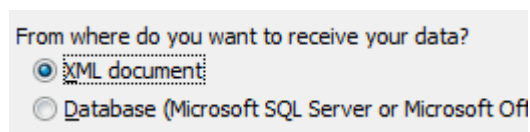
1. Right click on your xsn and select **Design** to open the form in design mode.
2. Add the **QdabraRules** data connection:
  - a. Select **Tools > Data Connections...**



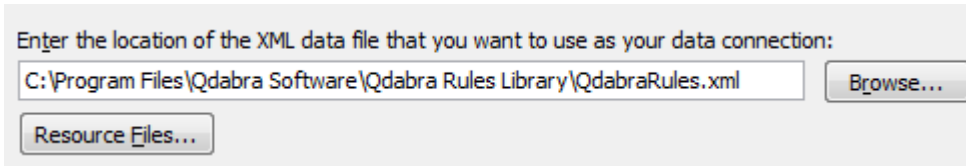
- b. Click **Add** button.
- c. Select **Create a new connection to > Receive data**. Click **Next**.



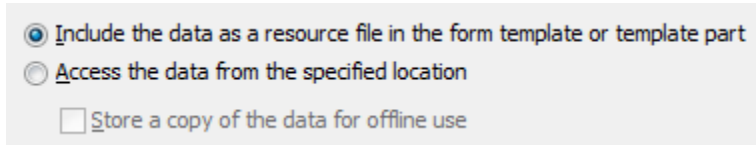
- d. Select **XML document** as the data source. Click **Next**.



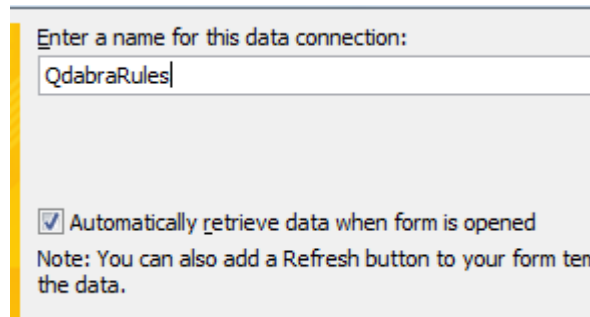
- e. Click on the **Browse...** button and navigate to the qRules installation folder and select the **QdabraRules.xml** file. Click **Next**.



- f. Make sure, that **Include the data as a resource file in the form template or template part** is selected if you have an option. Click **Next**.

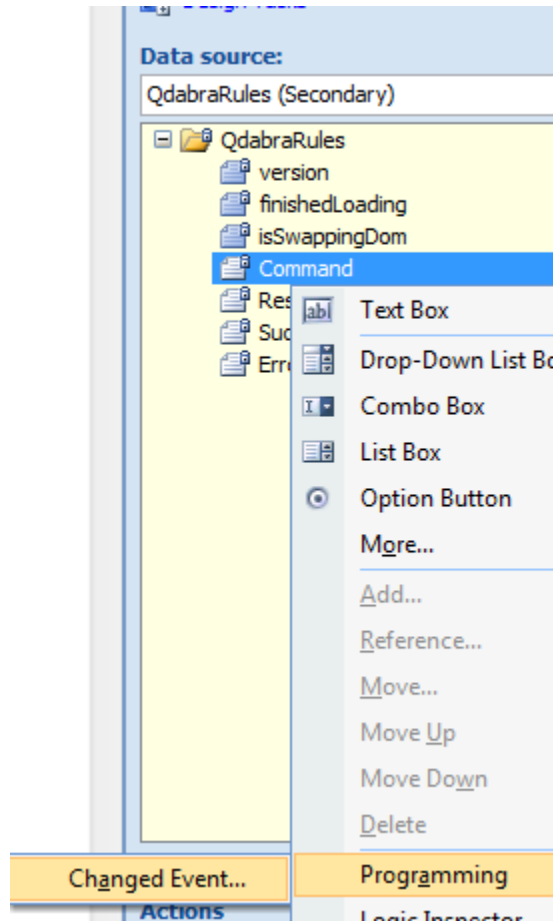


- g. Name this data connection **QdabraRules** and make sure **Automatically retrieve data when form is opened** is checked. Click **Finish**.

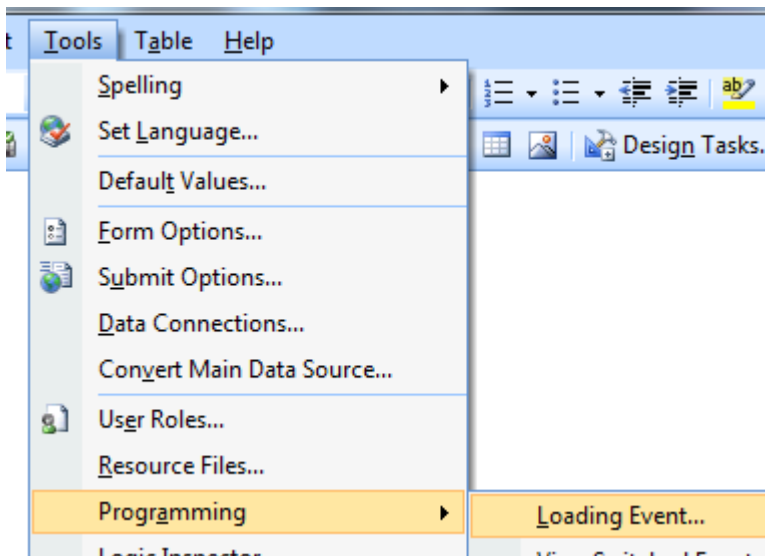


3. Add code to your form (this document uses C# -- for Visual Basic syntax, please see the end of the document):
- Right click on the **Command** field in the **QdabraRules (Secondary)** data source in the Data Source task pane, and select **Programming > Changed Event....** InfoPath automatically adds **QdabraRules\_Command\_Changed** event to the FormCode.cs.



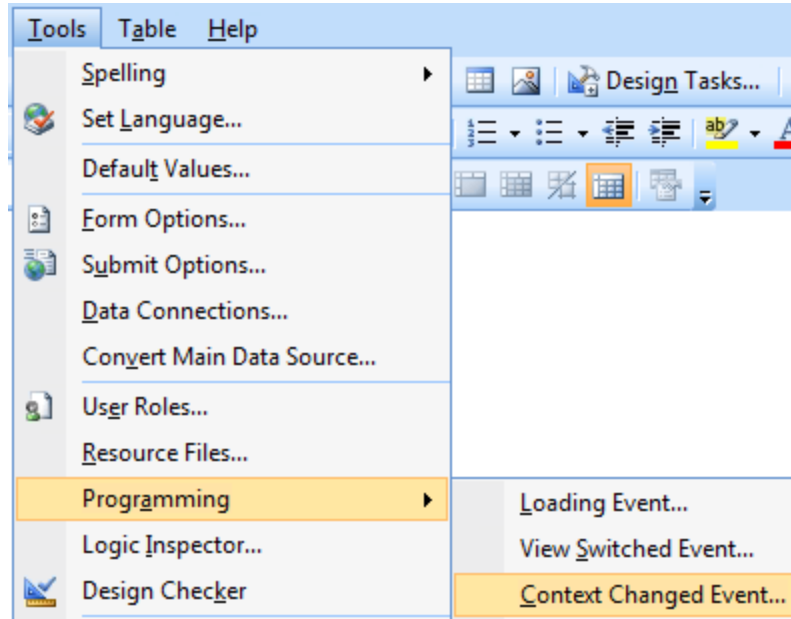


- b. Add a FormLoad event: click **Tools > Programming > Loading Event....** InfoPath automatically adds **FormEvents\_Loading** to the FormCode.cs.

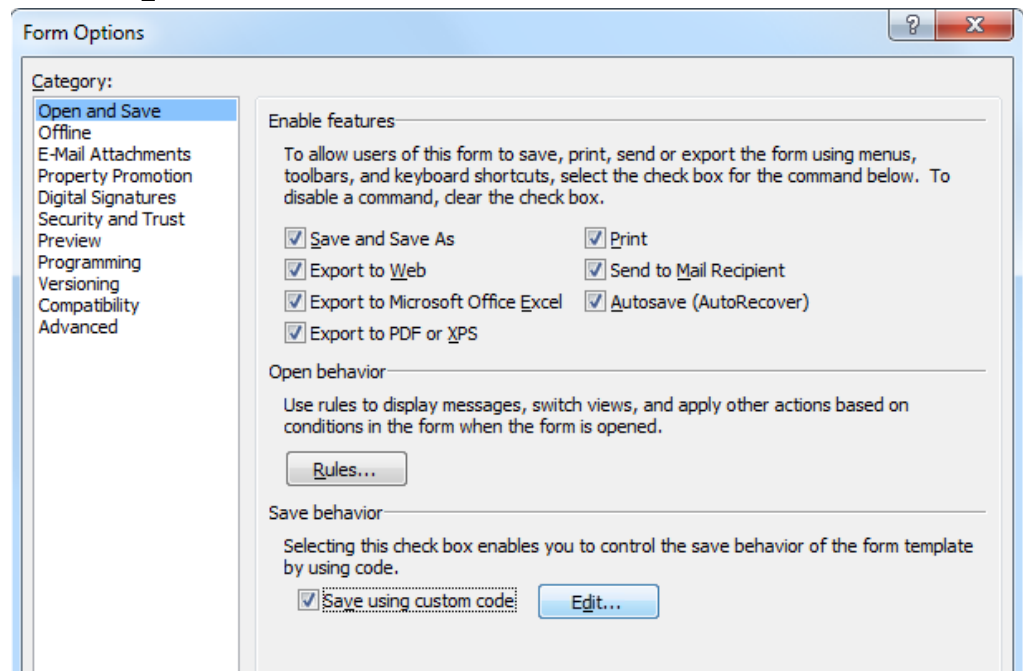


- c. If you are planning on using the client-only version of qRules, the following two events should be added. **NOTE:** these events are not compatible with browser forms.

- i. Add a Context Changed event: click **Tools > Programming > Context Changed Event....** InfoPath automatically adds **FormEvents\_ContextChanged** to the FormCode.cs.

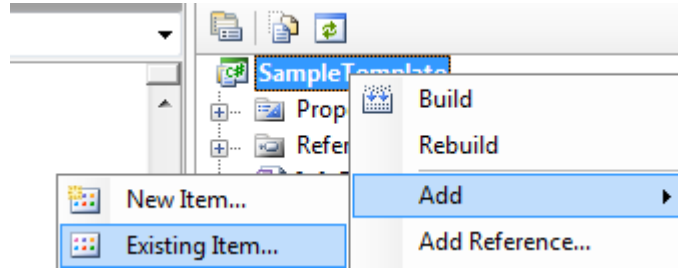


- ii. Add a Saved event: under **Tools > Form Options > Open and Save** check the **Save using custom code** checkbox then click the **Edit** button. InfoPath automatically adds **FormEvents\_Save** to the FormCode.cs.

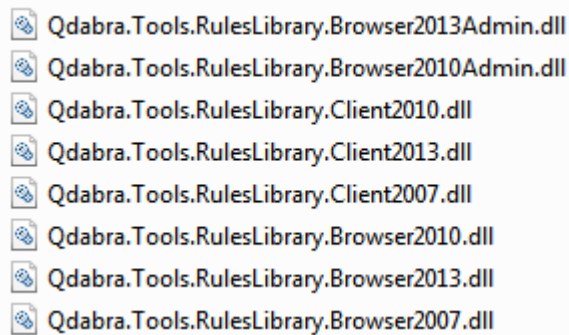


- d. In the Project Explorer window of Visual Studio, right click on the project name and select **Add > Existing Item...** from the context menu.



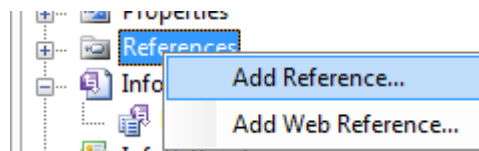


- e. Navigate to the qRules installation folder and select **Executable Files** from the **Files of Type** dropdown box.
  - i. For qRules 2.2, select the *Qdabra.Tools.RulesLibrary.dll* from the installation folder.
  - ii. For qRules 2.3, select the *Qdabra.Tools.RulesLibrary.dll* from either the Browser or the Client folder, depending on whether you are interested in Client only features, or Browser and Client.
  - iii. For qRules 2.4 up to 5.2, you'll need to select from the correct folder based on the version of your template and chosen compatibility and deployment choice. Folders without a specified year in the name are for InfoPath 2007.
  - iv. For qRules 5.2 and later, the version and compatibility information is now part of the DLL name:



All DLLs can be found in the ~\Qdabra Software\Qdabra Rules Library\DLLs folder.

- f. Right click on the **References** and select **Add Reference...** menu option.



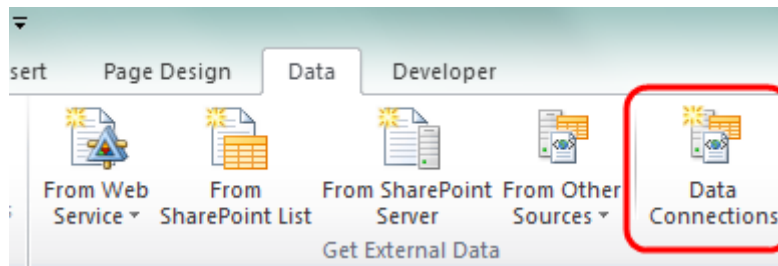
- g. Select the **Browse** tab, and this time browse to the DLL file added in step e above in your code project. Click **OK**.
- h. Please skip to the section titled "[All Versions – Add to the Code File](#)"

## INFOPATH 2010

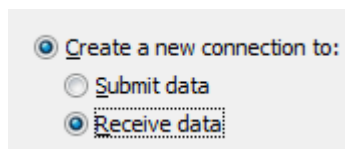
1. Right click on your xsn and select **Design** to open the form in design mode.
2. Add the **QdabraRules** data connection:



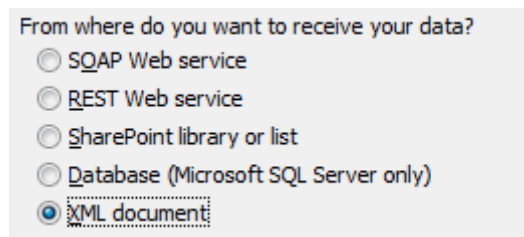
- a. Select **Data > Data Connections...**



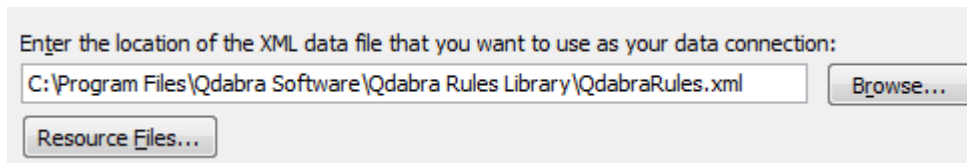
- b. Click **Add** button.  
c. Select **Create a new connection to > Receive data**. Click **Next**.



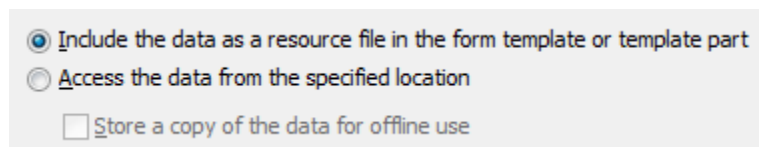
- d. Select **XML document** as the data source. Click **Next**.



- e. Click on the **Browse...** button and navigate to the qRules installation folder and select the **QdabraRules.xml** file. Click **Next**.



- f. Make sure, that **Include the data as a resource file in the form template or template part** is selected if you have an option. Click **Next**.



- g. Name this data connection **QdabraRules** and make sure **Automatically retrieve data when form is opened** is checked. Click **Finish**.



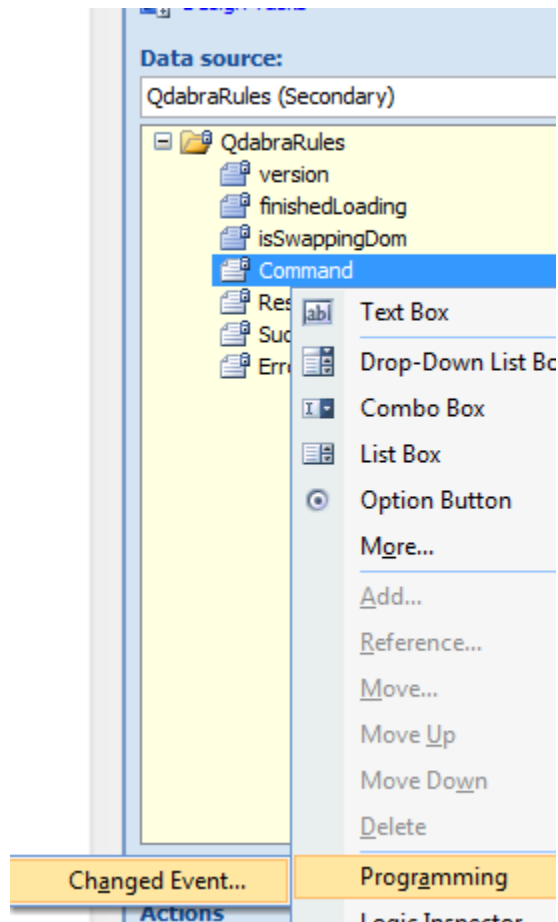
Enter a name for this data connection:

QdabraRules

Automatically retrieve data when form is opened

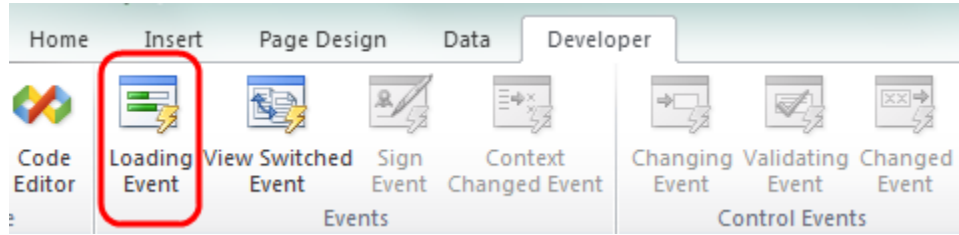
Note: You can also add a Refresh button to your form then the data.

3. Add code to your form (this document uses C# -- for Visual Basic syntax, please see the end of the document):
  - a. Right click on the **Command** field in the **QdabraRules (Secondary)** data source in the Fields task pane, and select **Programming > Changed Event....** InfoPath automatically adds **QdabraRules\_Command\_Changed** event to the FormCode.cs.



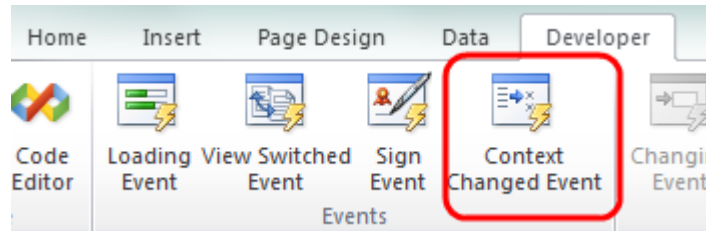
- b. Add a FormLoad event: click **Developer > Loading Event....** InfoPath automatically adds **FormEvents\_Loading** to the FormCode.cs.





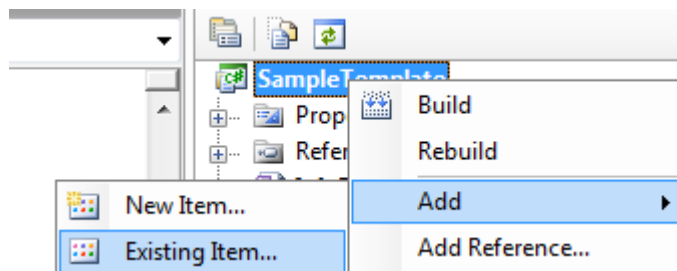
- c. If you are planning on using the client-only version of qRules, the following two events should be added. **NOTE:** these events are not compatible with browser forms.

- i. Add a Context Changed event: click **Developer > Context Changed Event...** InfoPath automatically adds **FormEvents\_ContextChanged** to the FormCode.cs.



- ii. Add a Saved event: under **File > Form Options > Filler Features** check the **Save using custom code** checkbox then click the **Edit** button. InfoPath automatically adds **FormEvents\_Save** to the FormCode.cs.

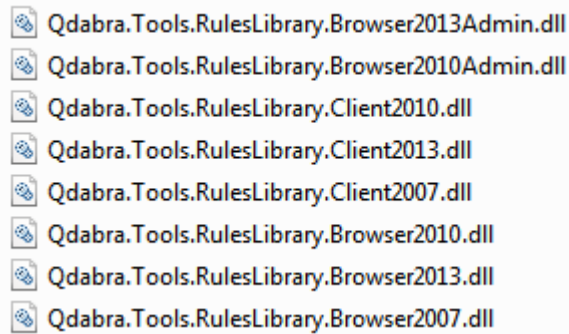
- d. In the Project Explorer window of Visual Studio, right click on the project name and select **Add > Existing Item...** from the context menu.



- e. Navigate to the qRules installation folder and select **Executable Files** from the **Files of Type** dropdown box.
- For qRules 2.2, select the *Qdabra.Tools.RulesLibrary.dll* from the installation folder.
  - For qRules 2.3, select the *Qdabra.Tools.RulesLibrary.dll* from either the Browser or the Client folder, depending on whether you are interested in Client only features, or Browser and Client.
  - For qRules 2.4 up to 5.2, you'll need to select from the correct folder based on the version of your template and chosen compatibility and deployment choice. Folders without a specified year in the name are for InfoPath 2007.
  - For qRules 5.2 and later, the version and compatibility information is now part of the DLL name:

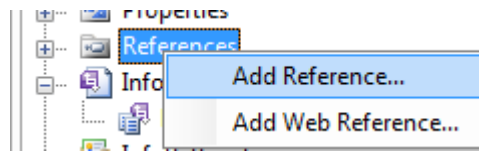






All DLLs can be found in the ~\Qdabra Software\Qdabra Rules Library\DLLs folder.

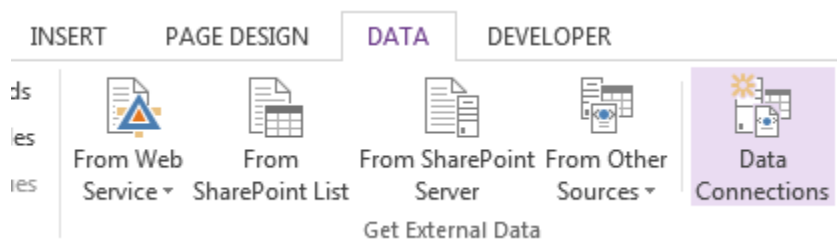
- f. Right click on the **References** and select **Add Reference...** menu option.



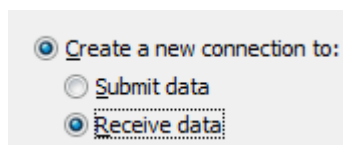
- g. Select the **Browse** tab, and this time browse to the DLL file added in step e above in your code project. Click **OK**.
- h. Please skip to the section titled "[All Versions – Add to the Code File](#)"

## INFOPATH 2013

1. Right click on your xsn and select **Design** to open the form in design mode.
2. Add the **QdabraRules** data connection:
  - a. Select **Data > Data Connections...**



- b. Click **Add** button.
- c. Select **Create a new connection to > Receive data**. Click **Next**.



- d. Select **XML document** as the data source. Click **Next**.



From where do you want to receive your data?

- SOAP Web service
- REST Web service
- SharePoint library or list
- Database (Microsoft SQL Server only)
- XML document

- e. Click on the **Browse...** button and navigate to the qRules installation folder and select the **QdabraRules.xml** file. Click **Next**.

Enter the location of the XML data file that you want to use as your data connection:

C:\Program Files\Qdabra Software\Qdabra Rules Library\QdabraRules.xml

Browse...

Resource Files...

- f. Make sure, that **Include the data as a resource file in the form template or template part** is selected if you have an option. Click **Next**.

- Include the data as a resource file in the form template or template part
- Access the data from the specified location
- Store a copy of the data for offline use

- g. Name this data connection **QdabraRules** and make sure **Automatically retrieve data when form is opened** is checked. Click **Finish**.

Enter a name for this data connection:

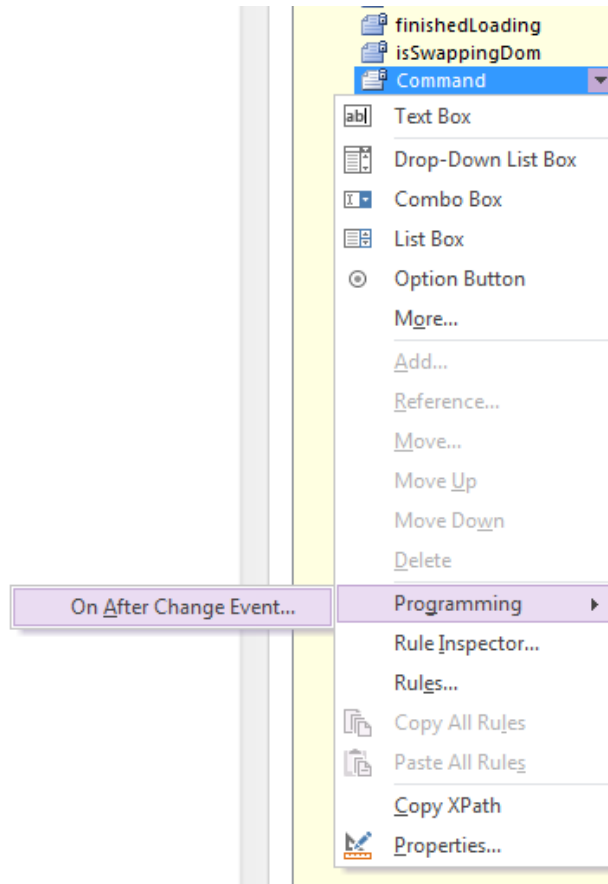
QdabraRules

Automatically retrieve data when form is opened

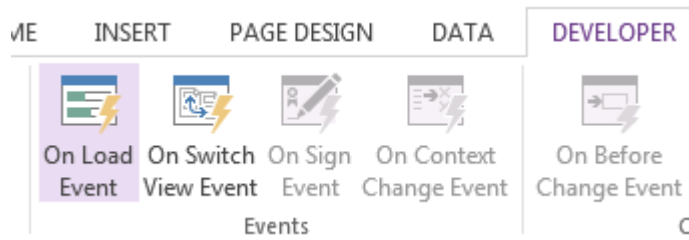
Note: You can also add a Refresh button to your form then the data.

3. Add code to your form (this document uses C# -- for Visual Basic syntax, please see the end of the document):
- a. Right click on the **Command** field in the **QdabraRules (Secondary)** data source in the Fields task pane, and select **Programming > On After Change Event....** InfoPath automatically adds **QdabraRules\_Command\_Changed** event to the FormCode.cs.





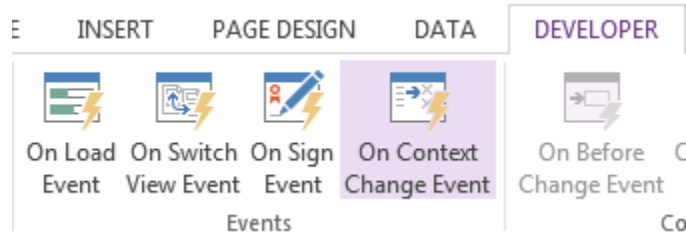
- b. Add a FormLoad event: click **Developer > On Load Event....** InfoPath automatically adds **FormEvents\_Loading** to the FormCode.cs.



- c. **If you are planning on using the client-only version of qRules**, the following two events should be added. **NOTE:** these events are not compatible with browser forms.

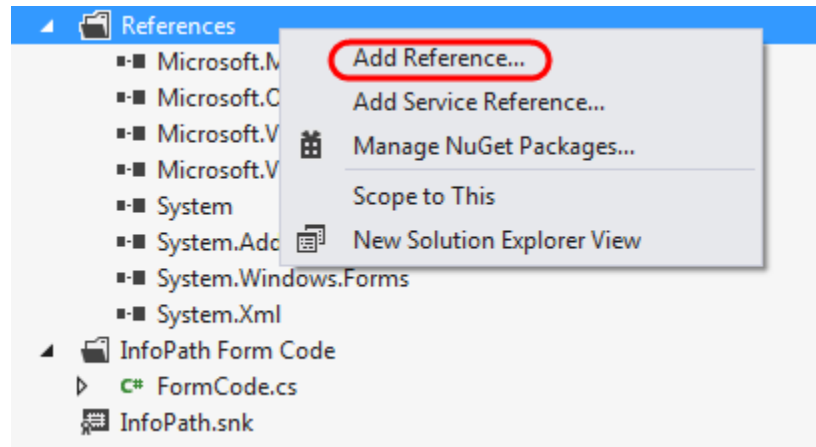


- i. Add a Context Changed event: click **Developer > Context Changed Event....** InfoPath automatically adds **FormEvents\_ContextChanged** to the FormCode.cs.

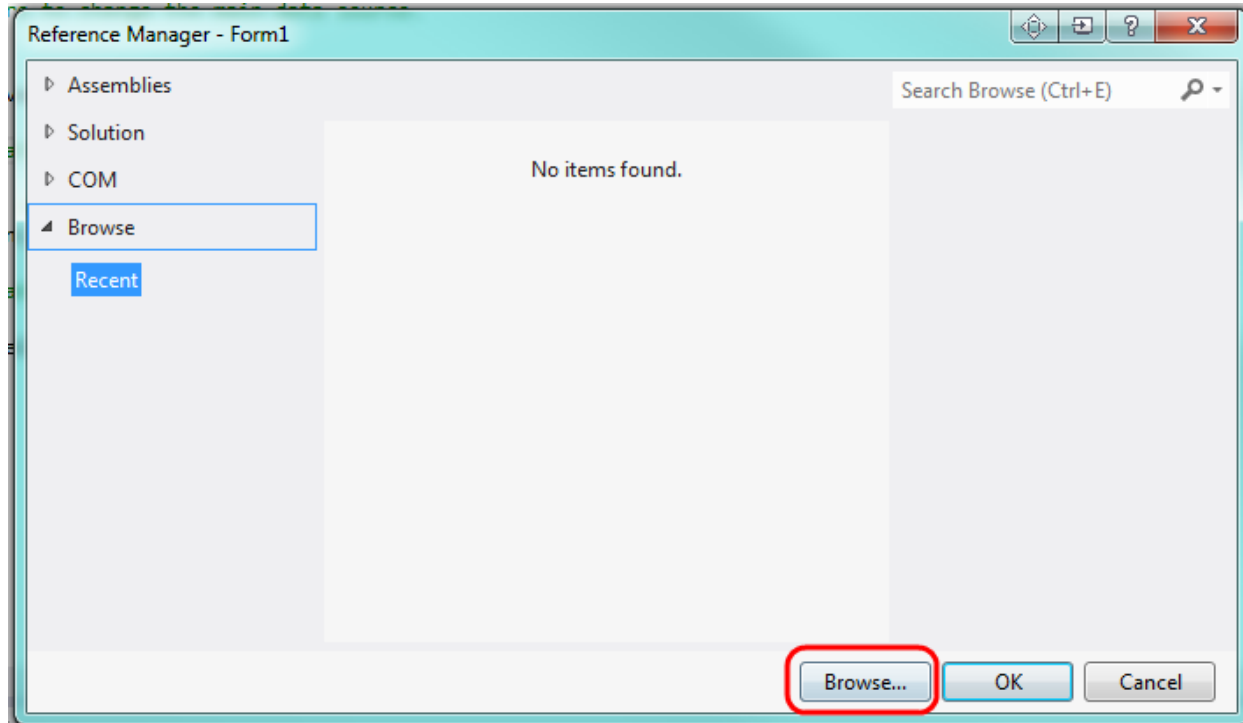


- ii. Add a Saved event: under **File > Form Options > Filler Features** check the **Save using custom code** checkbox then click the **Edit** button. InfoPath automatically adds **FormEvents\_Save** to the FormCode.cs.

- d. In the Project Explorer window of Visual Studio, right click on the References folder and select **Add Reference** from the context menu.



- e. In the Reference Manager, select Browse from the categories on the left, then click the Browse button:



- f. Navigate to the qRules installation folder and select the qRules DLL you wish to add a reference to.
- i. For qRules 2.2, select the *Qdabra.Tools.RulesLibrary.dll* from the installation folder.
  - ii. For qRules 2.3, select the *Qdabra.Tools.RulesLibrary.dll* from either the Browser or the Client folder, depending on whether you are interested in Client only features, or Browser and Client.
  - iii. For qRules 2.4 up to 5.2, you'll need to select from the correct folder based on the version of your template and chosen compatibility and deployment choice. Folders without a specified year in the name are for InfoPath 2007.
  - iv. For qRules 5.2 and later, the version and compatibility information is now part of the DLL name:

```

Qdabra.Tools.RulesLibrary.Browser2013Admin.dll
Qdabra.Tools.RulesLibrary.Browser2010Admin.dll
Qdabra.Tools.RulesLibrary.Client2010.dll
Qdabra.Tools.RulesLibrary.Client2013.dll
Qdabra.Tools.RulesLibrary.Client2007.dll
Qdabra.Tools.RulesLibrary.Browser2010.dll
Qdabra.Tools.RulesLibrary.Browser2013.dll
Qdabra.Tools.RulesLibrary.Browser2007.dll

```

All DLLs can be found in the ~\Qdabra Software\Qdabra Rules Library\DLLs folder.



- g. With InfoPath 2013, you will no longer need to include the assembly with the project. From [this MSDN page](#):

When you add a reference to a custom assembly in a managed-code form template project, that assembly is included within the form template file (.xsn) when your project is compiled and published.

#### ▲ Add and Reference a Custom Assembly

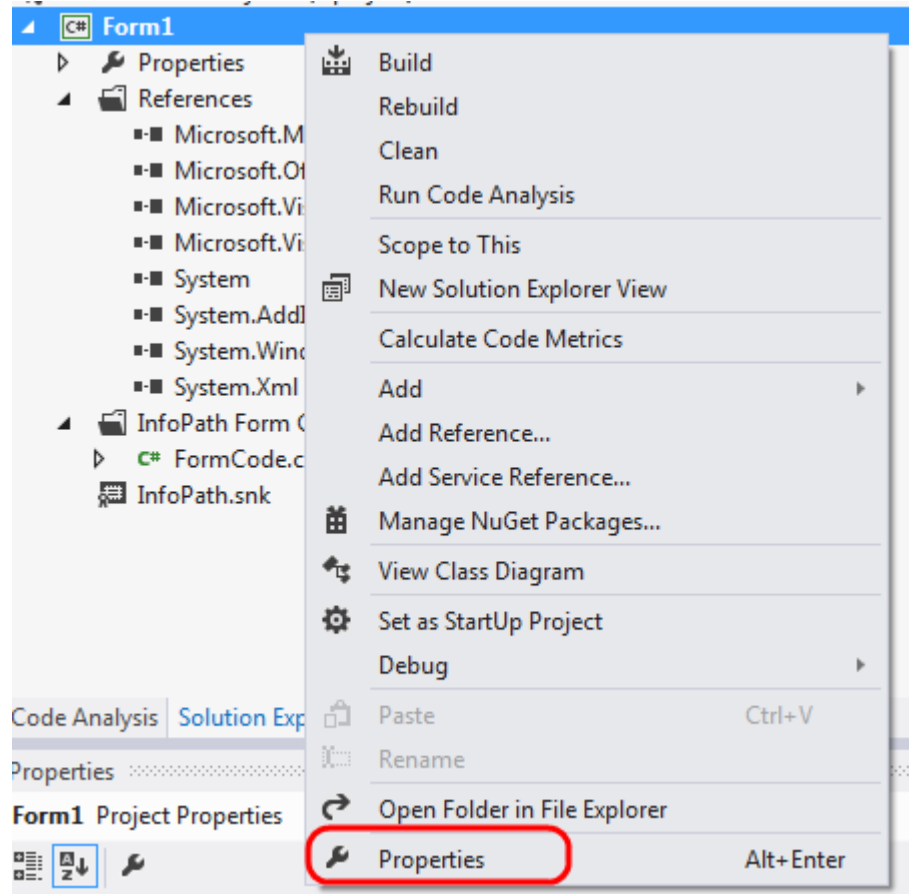
To avoid a conflict with how the InfoPath project system manages files that are added to the form template file, do not copy any custom assemblies that you want to reference into the top-level folder of a form template project. By default, this will be a path in the following format: `<drive>:\Users\UserName\Documents\InfoPath Projects\ProjectName`

If you do want to move custom assemblies that you reference to a location within the project folder, you must create a subfolder under the main project folder, and then copy and reference custom assemblies from that subfolder. However, be aware that creating a subfolder for referenced assemblies is not necessary. As long as a referenced assembly is not located within the project's top-level folder, the InfoPath project system will copy the assembly into the form template file (.xsn) when the project is compiled and published.

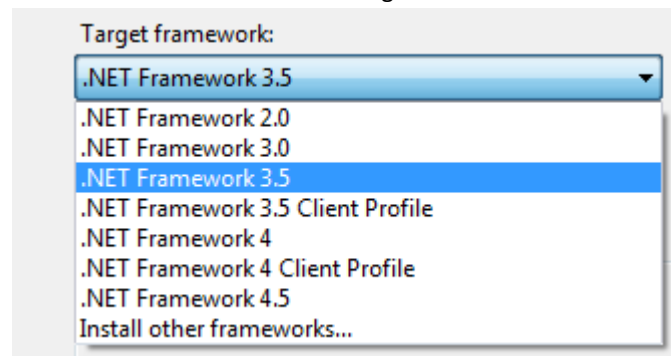
#### Reference a custom assembly from its default location

1. Open the form template project in Visual Studio 2012.
  2. On the **Project** menu, click **Add Reference**.
  3. Click the **Browse** tab, locate and specify the assembly, and then click **OK** to add the reference.
- h. If your form is InfoPath 2007 or 2010 compatible, Visual Studio will have set the target framework to 2.0. Recent versions of the qRules DLL require .NET 3.5. You can change the project compatibility in the project properties. Right click the project in the project explorer and select Properties:





Then select 3.5 or later for the Target Framework:



## ALL VERSIONS – ADD TO THE CODE FILE

1. In the **FormCode.cs** file, after all of the using statements at the top, add a using statement for the Qdabra.Tools.RulesLibrary.



```

using Microsoft.Office.InfoPath;
using System;
using System.Windows.Forms;
using System.Xml;
using System.Xml.XPath;
using mshtml;

using Qdabra.Tools.RulesLibrary;

```

2. Add this single line of code to the **Loading** event:

```
qRulesMain.FormLoad(this, Application, e);
```

3. Add this single line of code to the Changed event for the **Command** node:

```
qRulesMain.ExecuteCommand(this, Application, e);
```

4. **If you are planning on using the client-only version of qRules**, add this single line of code to the **Context Changed** event:

```
qRulesMain.ContextChanged (this, Application, e);
```

5. **If you are planning on using the client-only version of qRules**, add this single line of code to the **Save** event:

```
qRulesMain.FormSave (this, Application, e);
```

6. This is how the changed and loading events look:

```

public void QdabraRules__Command_Changed(object sender, XmlEventArgs e)
{
    qRulesMain.ExecuteCommand(this, Application, e);
}

public void FormEvents_Loading(object sender, LoadingEventArgs e)
{
    qRulesMain.FormLoad(this, Application, e);
}

```

7. **If you are planning on using the client-only version of qRules**, you will have two additional events:

```

public void FormEvents_ContextChanged(object sender,
ContextChangedEventArgs e)
{
    qRulesMain.ContextChanged(this, Application, e);
}

public void FormEvents_Save(object sender, SaveEventArgs e)
{
    qRulesMain.FormSave (this, Application, e);
}

```

8. Now compile this project, your project should compile with no errors.

You can now begin adding qRules commands to your InfoPath form.

If you are using Visual Basic, instead of C#, you'll need an Imports statement:





```
Imports Microsoft.Office.InfoPath
Imports System
Imports System.Xml
Imports System.Xml.XPath
Imports System.Windows.Forms
Imports mshtml

Imports Qdabra.Tools.RulesLibrary
```

And the changed and loading events will look like this:

```
Public Sub QdabraRules__Command_Changed(ByVal sender As Object,
ByVal e As XmlEventArgs)

    qRulesMain.ExecuteCommand(Me, Application, e)

End Sub

Public Sub FormEvents_Loading(ByVal sender As Object, ByVal e As
LoadingEventArgs)

    qRulesMain.FormLoad(Me, Application, e)

End Sub
```

If you are planning on using the client-only version of qRules, you will have two additional events, the context changed and save events, which for Visual Basic will look like this:

```
Public Sub FormEvents_ContextChanged(ByVal sender As Object,
ByVal e As ContextChangedEventArgs)

    qRulesMain.ContextChanged(Me, Application, e)

End Sub

Public Sub FormEvents_Save(ByVal sender As Object, ByVal e As
SaveEventArgs)

    qRulesMain.FormSave(Me, Application, e)

End Sub
```



## 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](mailto:Support@Qdabra.com).

You can also use the [InfoPathDev.com](http://InfoPathDev.com) [Qdabra Product support forums](http://InfoPathDev.com) 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](mailto:Support@Qdabra.com)  
**Website:** <http://www.qdabra.com>  
**Community:** <http://www.infopathdev.com>

