



QDABRA QRULES V2.1

How to use the FormatDate and GetWeekDay commands

INTRODUCTION

qRules v2.1 includes two new commands that allow you to work with the date fields in your InfoPath form.

GetWeekDay allows you to specify a date and a weekday, and the command will return the date value for the specified day of the week. This would allow us, for example, to obtain the Monday relative to the start of the current week. FormatDate, on the other hand, will allow us to change the format of a date value.

This tutorial will walk you through an example where the two commands are used together. First we will generate a list of Mondays to select from. After we have a list of Mondays, we will reformat the date values into a more user-friendly format.

PRE-REQUISITES

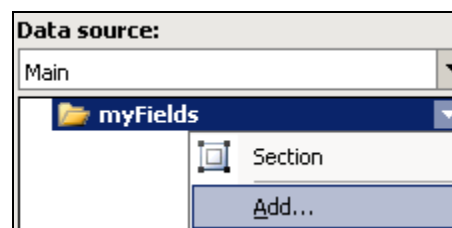
This tutorial assumes that you have qRules v2.1 already installed on your machine. For details on this, please see the User Guide included with the installer.

- 1) Create a new InfoPath form: In InfoPath, design a new, blank form. Select **File > Save As** and save the XSN to your local drive. Close the form.
- 2) Launch the qRules v2.1 Injector, browse to select the XSN you created and click Inject. Close the qRules Injector and open the form in Design mode.

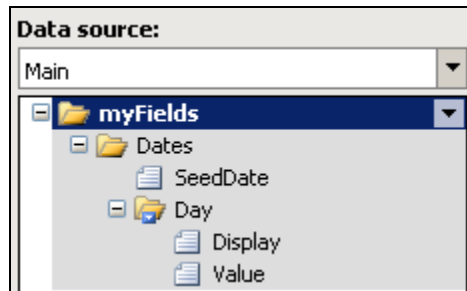
For more information on the steps above, refer to the User Guide.

GENERATE A LIST OF MONDAYS TO SELECT FROM

- A. We first need to add the fields we will need for this tutorial.
 1. Switch to the Data Source taskpane, right click and select **Add**.

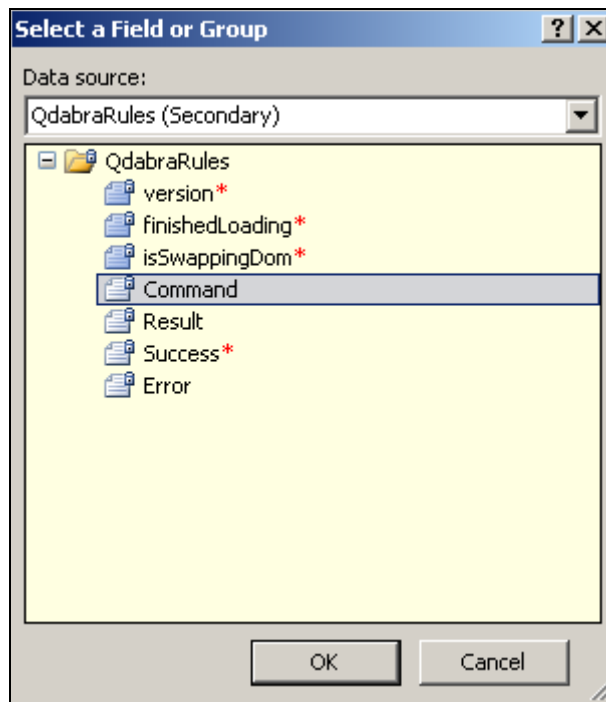


2. Design the data source seen below, where all of the fields are of data type *date*, except for Display, which should be a string.



B. Next we will create on load rules to generate values to these fields.

3. Go to **Tools > Form Options**. Under **Open and Save > Open behavior**, click **Rules**.
4. Click **Add**. Name your rule **Setup Monday Dropdown**.
5. Click **Add Action** and select **Set a field's value** from the **Action** dropdown.
6. For the **Field**, select the **Command** node from the **QdabraRules** secondary data source.

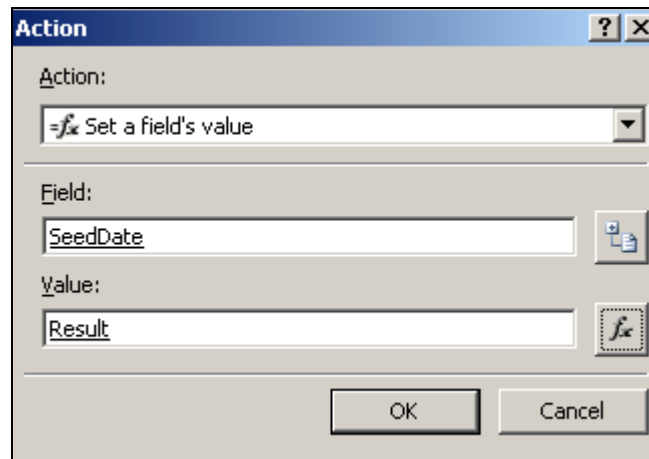


7. For the **Value**, we will use the `GetWeekDay` command to retrieve the Monday relative to today's date. When the command does not contain the `/date` parameter, the command will use today's date. The command is: ***GetWeekDay /day=Monday***

Now that the qRules command has executed, its result is stored in the **Result** node in the **QdabraRules** secondary data source. We must store this value in another field before we execute another command, since the result of any other command will overwrite the **Result** node.



8. Click **Add Action** and select **Set a field's value** from the **Action** dropdown.
9. The **Field** will be the **SeedDate** and the **Value** will be the **Result**, as shown below.



Next we will generate a list of Mondays based on this **SeedDate**. We can use the qRules command **SetValue** to add values to our repeating group.

10. Click **Add Action** and select **Set a field's value** from the **Action** dropdown.
11. For the **Field**, select the **Command** node from the **QdabraRules** secondary data source.
12. For the **Value**, click on the fx button, click on **Insert Function**, and select the concat function. Prepare the concat statement as seen below.

```
concat("SetValue /xpath=/my:myFields/my:Dates/my:Day[1]/my:Value /value=",  
xdDate:AddDays(my:Dates/my:SeedDate, 21))
```

What does this command mean? **SetValue** allows us to set the value of the given xpath, in this case, the first node in the repeating **my:Day/my:Value**. The value is another function, which adds 21 days to our **SeedDate**. This means we will be generating a Monday date three weeks into the future.

Before adding more dates, we need to add more rows into the repeating group. To do this:

13. Click **Add Action** and select **Set a field's value** from the **Action** dropdown.
14. For the **Field**, select the **Command** node from the **QdabraRules** secondary data source.
15. For the **Value**, enter **Insert /parent=/my:myFields/my:Dates /child=my:Day /count=6**. The **/count** parameter is set to six because we will be adding six more date values.

Now you have six new, blank rows in the repeating **my:Day** group. Populate them with dates by repeating steps 10, 11 and 12, and simply changing the index on **my:Day** and the number of days added to the **SeedDate**. The commands will look like this:



```
concat("SetValue /xpath=/my:myFields/my:Dates/my:Day[2]/my:Value /value=",  
xdDate:AddDays(my:Dates/my:SeedDate, 14))
```

```
concat("SetValue /xpath=/my:myFields/my:Dates/my:Day[3]/my:Value /value=",  
xdDate:AddDays(my:Dates/my:SeedDate, 7))
```

```
concat("SetValue /xpath=/my:myFields/my:Dates/my:Day[4]/my:Value /value=",  
my:Dates/my:SeedDate)
```

```
concat("SetValue /xpath=/my:myFields/my:Dates/my:Day[5]/my:Value /value=",  
xdDate:AddDays(my:Dates/my:SeedDate, -7))
```

```
concat("SetValue /xpath=/my:myFields/my:Dates/my:Day[6]/my:Value /value=",  
xdDate:AddDays(my:Dates/my:SeedDate, -14))
```

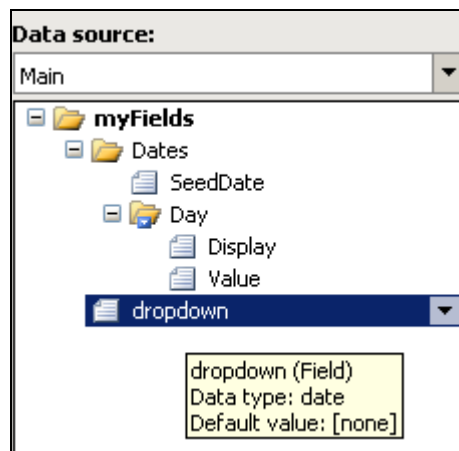
```
concat("SetValue /xpath=/my:myFields/my:Dates/my:Day[7]/my:Value /value=",  
xdDate:AddDays(my:Dates/my:SeedDate, -21))
```

As a result, now the **my:Day** repeating group has a **my:Value** field populated with the current week's Monday date, as well as the Monday dates from three weeks into the future and three weeks into the past.

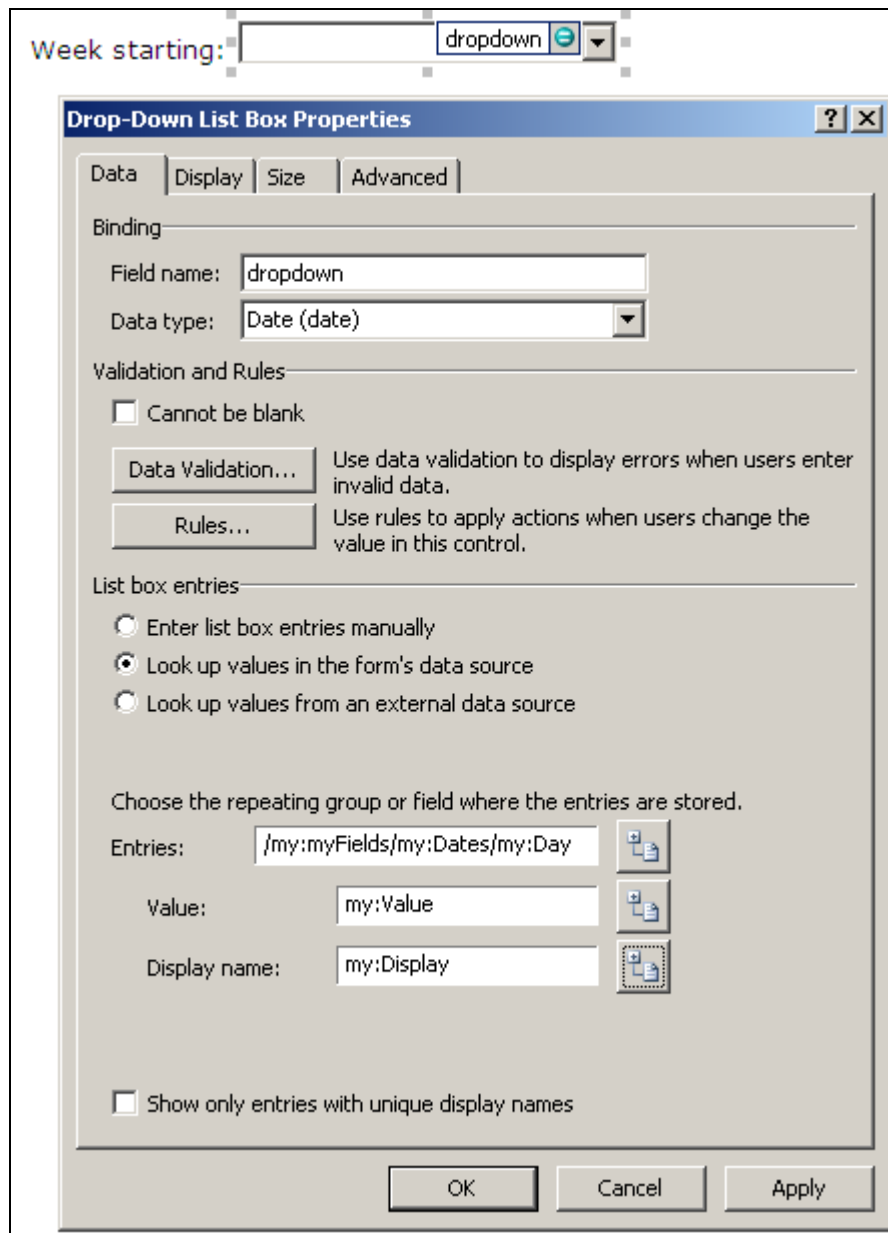
USE FORMATDATE TO SET THE DISPLAY TO A USER-FRIENDLY FORMAT

Despite our progress, the **my:Display** field is still not set. Fortunately, we can use `FormatDate` to do this and provide a user-friendly date format.

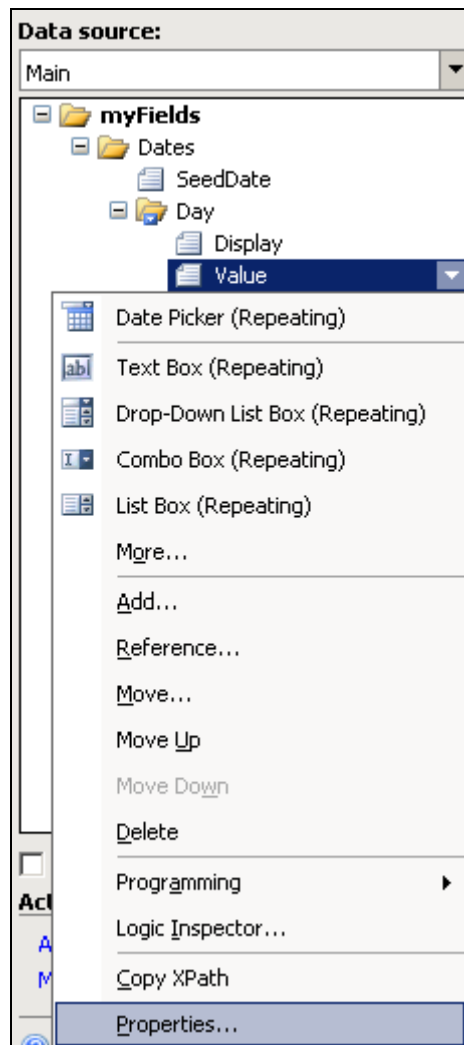
16. To your data source, add a dropdown field and display it in the form's view.



17. Set the dropdown to **Look up values in the form's data source**, with the **Entries**, **Value** and **Display** settings set as shown below.

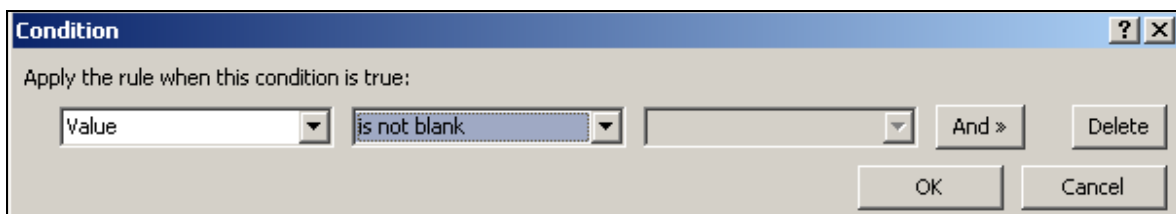


18. Right-click on the **my:Value** node and select **Properties**.



19. Switch to the **Rules and Merge** tab and click **Add**.

20. Click on **Set condition**. Select **Value** in the first dropdown and select **is not blank** from the middle dropdown.



21. Click on **Add Action**.

22. For the **Field**, select the **Command** node from the **QdabraRules** secondary data source.



- For the **Value**, click on the fx button, click on **Insert Function**, and select the concat function. Build the function like this, where the dot (.) represents the current field (**my:Value**).

`concat ("FormatDate /date=" , "." , " /format=d")`

- Click **Add Action** and select **Set a field's value** from the **Action** dropdown.
- The **Field** will be the **my:Display** node and the **Value** will be the **Result** from the **QdabraRules** secondary data source.

What do these steps accomplish? Every time that an entry is added to the **my:Value** field in the previous section, this new rule will format the date into a user friendly value and store it in the **my:Display** field.

- Preview the form to verify the values shown by the dropdown.

Week starting: [dropdown menu]

- 4/5/2010
- 3/29/2010
- 3/22/2010
- 3/15/2010
- 3/8/2010
- 3/1/2010
- 2/22/2010

OPTIONS FOR FORMATDATE

How do we know how to format the date? The format is controlled by the **/format** parameter. In the example above, we used the parameter 'd', but what does that mean? To find out, visit MSDN: <http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx>. There are numerous options for using FormatDate with custom formats and the MSDN link above explains these options.

To illustrate the power of FormatDate, we will change the command in step 23 to `concat("FormatDate /date=" , "." , " /format=dd MMMM yyyy")` and Preview the form to see the new date format in the dropdown.

Week starting: [dropdown menu]

- 05 April 2010
- 29 March 2010
- 22 March 2010
- 15 March 2010
- 08 March 2010
- 01 March 2010
- 22 February 2010



You can even use something like `/format="Arriving at " hh:mm` to get "Arriving at: 09:03", as shown below.

Command:
<code>FormatDate /date=2010-01-02T09:03 /format="Arriving at: "hh:mm</code>
Result: Arriving at: 09:03
<input checked="" type="checkbox"/> Success
Error: <input type="text"/>

In the examples above we show the usage of `dd MMMM yyyy` and `hh:mm` to create customized date/time formats. However, you can also use an escape character, like this: `%h`. This is what MSDN tells us:

To use any of the custom date and time format specifiers as the only specifier in a format string (that is, to use the "d", "f", "F", "g", "h", "H", "K", "m", "M", "s", "t", "y", "z", ":", or "/" custom format specifier by itself), include a space before or after the specifier, or include a percent ("%") format specifier before the single custom date and time specifier.

To summarize: while this command allows the developer to create any number of different date formats, it also opens the door for formatting issues. As you design your form, you'll want to debug your command carefully to make sure you obtain the desired result.