

USE QRULES TO CREATE UNIQUE FORM NAMES

PRODUCT: qRules v3.3
LAST UPDATED: August 25, 2011

SCENARIO

Just like the Vehicle Identification Number for your car, your business process needs to uniquely identify the documents in your system. Unique IDs enable opening forms via links from a Web server or email message.

To prevent new forms from overwriting existing ones you'll want to name them on save or submit. Relying on your users to manually enter a name that is both meaningful and unique is not a good idea. Just like email subject lines, some users will have meaningful names and some won't. Consistency will not be kept if it isn't enforced and users may overwrite existing data without paying attention.

VALUE PROPOSITION

qRules includes commands to support uniquely naming your form. You can access these commands via rules when your form is opened or submitted to automatically create a name. Automating the process of creating a unique name will also speed up the submit process since users won't have to type in a name.

Using rules to automate creating unique names for your forms is a best practice for the following reasons:

- Prevents users from overwriting existing data by mistake
- Enhances comprehension of form content
 - Consistent names are easier for humans to parse
 - Meaningful names can describe the contents of the form
- Eases migration of data since unique names can travel from server to server
- Better integration with other systems
 - Links to open forms show more meaningful information about them
 - Forms can be saved to server or database without fear of overwrites

This document will show you how you can uniquely identify your form using two techniques. As a bonus, we also show a related technique of how to randomize entries of a repeating section using Random Numbers.

REQUIREMENTS

- qRules v2.3 or above



GENERATE A GUID

GUID stands for [Globally Unique ID](#). Wikipedia defines a GUID as follows: “A **globally unique identifier** or **GUID** (pronounced /'gwid/) is a special type of identifier used in [software](#) applications to provide a unique reference number. The value is represented as a 32 character hexadecimal character string, such as {21EC2020-3AEA-1069-A2DD-08002B30309D} and usually stored as a 128 bit integer.”

Benefits:

- GUIDs are independent – you can create a GUID without relying on data from your form
- GUIDs don't reveal anything about your form

Limitations:

- GUIDs don't carry any meaning – you can't tell what's in your form by looking at the GUID
- GUIDs are hard for humans to handle – they are longer than credit card numbers and contain alphabet characters (however, sometimes you don't want people to know what's in a form by looking at its title, hence the benefit above)

You can generate a GUID using qRules GenerateGuid command.

Command Overview

GenerateGuid generates a new GUID in the format specified by the format argument. The command returns the generated GUID in the Result field of the QdabraRules data source. Format can be one of the following:

- N = 32 digits: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
- D (default) = 32 digits separated by hyphens: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx.
- If you are using the default, you do not need to specify a format.
- B = 32 digits separated by hyphens, enclosed in brackets: {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
- P - 32 digits separated by hyphens, enclosed in parentheses: (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)

For example, you can use this command to create a GUID without hyphens: **GenerateGuid /format=N**

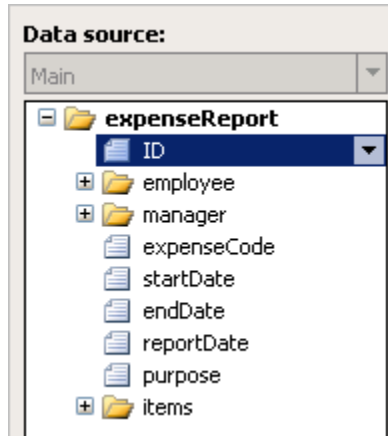
Rich Client Form

Follow these steps to create a sample form that can be used in the InfoPath client:

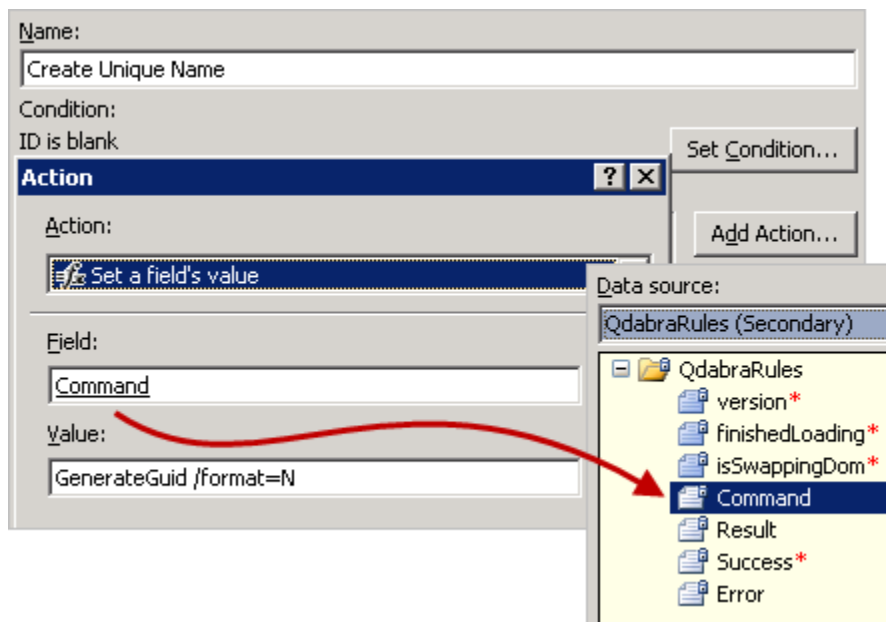
1. Inject qRules in your form (if you haven't already). Please refer to the qRules User Guide for more information on this.
2. Open your form in InfoPath Designer (we'll use the InfoPath 2007 out-of-box sample Expense Report)



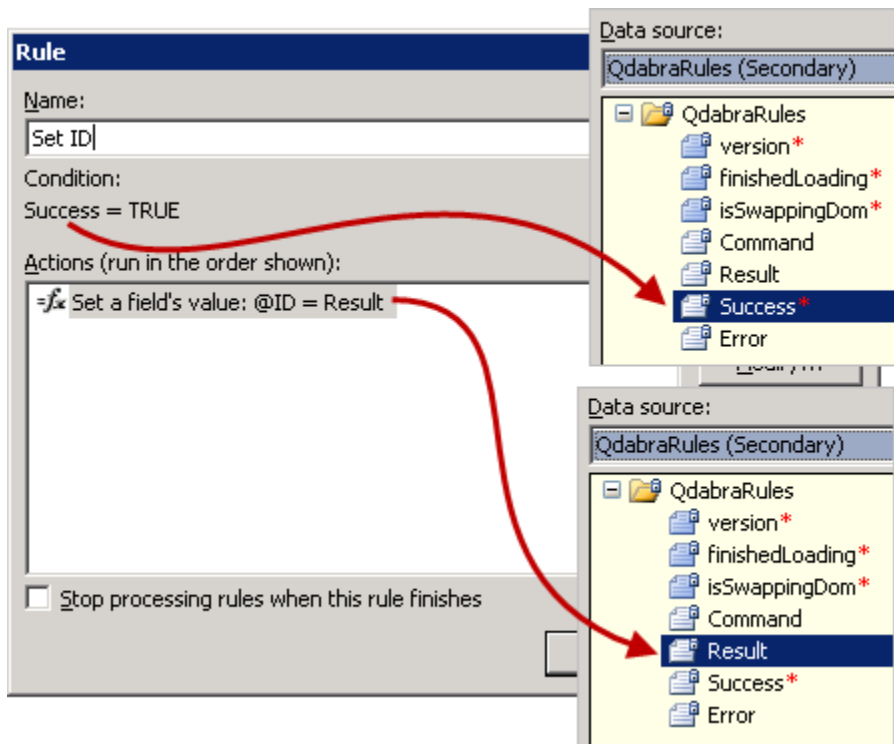
3. Add an attribute under root node called ID:



4. Modify the **Submit** button changing the *Action* to 'Rules and Custom Code' define these rules:
 - a. *Create Unique Name* only runs if **@ID** is blank and has one action
 - i. Sets qRules Command to **GenerateGuid /format=N**

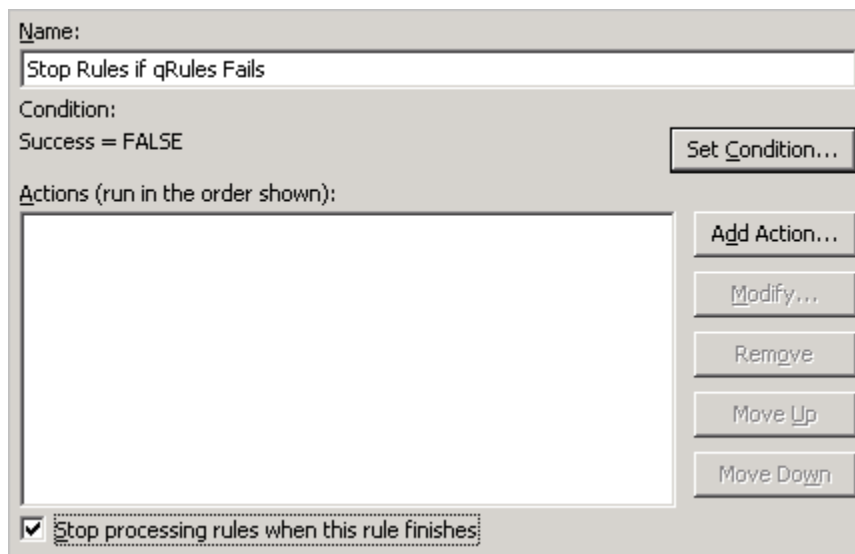


- b. *Set ID* only runs if qRules Success is TRUE. This also has one action.
 - i. Set @ID to qRules result



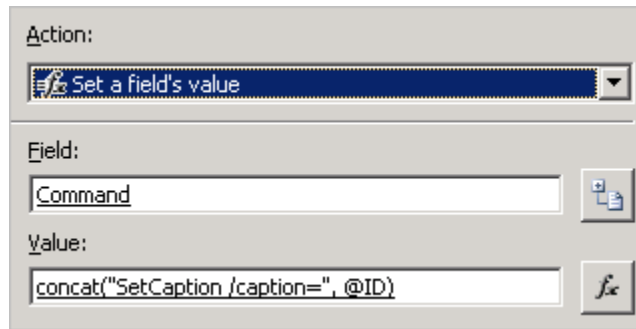
Note: if you have other qRules commands executing in your form, it would be wise to add a second condition to this Rule, such that it only executes if Success is TRUE and if @ID is blank (same condition as above)

- c. *Stop Rules if qRules Fails*
 - i. Rule is checked to stop processing others if it fails

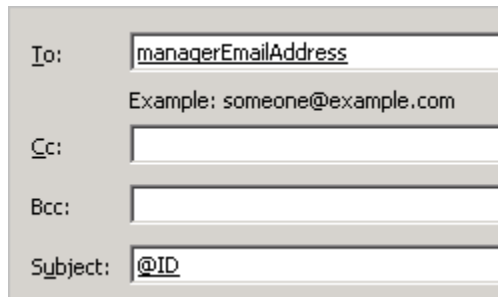




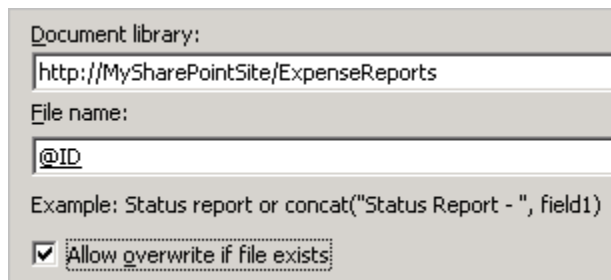
- d. *Set Caption* always runs
 - i. Sets qRules command to `concat("SetCaption /caption=", @ID)`



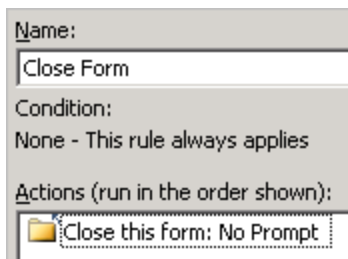
- e. Other rules that submit the form
 - i. If submitting via email: Use `@ID` for email subject.



- ii. If submitting to a SharePoint library: Use `@ID` as the file name.



- f. *Close form*




5. Disable **Save and Save As** (in *Tools > Form Options > Open and Save*) to prevent manually overriding the GUID for form name.
6. Publish your form and submit a few sample forms to test the functionality.

Browser Form in SharePoint

Browser forms require InfoPath Form Services which ships in SharePoint 2007 (MOSS) or SharePoint 2010 Enterprise. Browser forms cannot be run in preview mode – you have to publish. Same as Rich Client but don't use Set Caption rule (since it isn't supported for browser forms)

1. Refer to the qRules v2.3 User guide (located in **Start > All Programs > Qdabra > Tools > qRules**), section: *"Using qRules in Infopath Forms Services (IPFS)"* for steps on how to:
 - a. Browser-enable your form template.
 - b. Set your qRules-enabled buttons to ALWAYS send data to the server.
 - c. Publish your form as an admin-approved template.
 - d. Upload the form template to your InfoPath Forms Server via SharePoint Central Administration.
 - e. Activate the form template to your site collection.
 - f. Create or set your SharePoint library to allow management of content types.
 - g. Add the form template in your SharePoint library as a content type.



CREATE A KEY USING CONCAT()

Unlike GUIDs, Keys are not generated. Instead, you create them from parts of your form or from InfoPath macros. InfoPath supports the XPATH macro language, which includes the Concat function. You can use the Concat function in a rule when you set a field. Concat() can be used to concatenate fields from your form's data sources together. You can also add system information, such as the username and current date and time to ensure uniqueness.

Benefits:

- Keys are easy for humans to understand and refer to
- Keys carry information and can describe the form's contents

Limitations:

- Keys are more difficult to create. Keys are defined by the form designer. If the form designer doesn't do a good job, duplicate keys can result. That's why it's important to use extra information like username and date time to prevent duplicates.
- Keys can be long and contain special characters which will cause problems when making links. Once again, the form designer must ensure that the size of the key is kept in check using form validation.

What fields make a good key?

It depends on your form. Keys will become the name of your form so you want to make sure that you choose a name which is appropriate for the business process. Here's a list of common fields used to create a name with notes.

- Form Template Name – the name of your XSN
 - Use if you are storing all of your forms in a location where there may be other forms. Examples: a form library in SharePoint that supports multiple content types, folder on a file server where all XML forms are archived.
 - Don't use if the storage location is specific to that form template.
- Description – any field in the form that describes the form
 - Use if the field isn't too long and plain text
 - Don't use if the field is long, contains special characters, or encodes rich text.
- Username – logged in user or a field in the form
 - Use if the submitter of the form describes the form. Example: author of a document.
 - Don't use if you have multiple authors, lots of workflow, or people who submit on the behalf of others (unless they have the ability to type in the username).
- Time Zone
 - Use if you have forms submitted from different time zones with the possibility duplicate
 - Don't use if you are using username and users are always submitting from one timezone with no submit-on-behalf logic

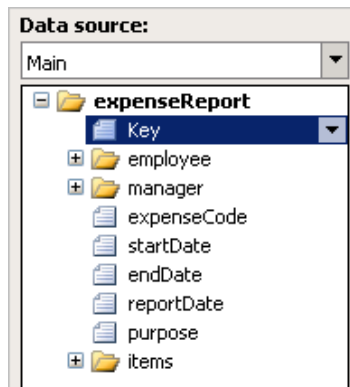


- Current Date and Time
 - Use to make the form unique. Use qRules to get milliseconds if you have large quantity of forms being submitted at same time.
 - Don't use if you have some other unique identifier (for example, a GUID) that you are concatenating to the key

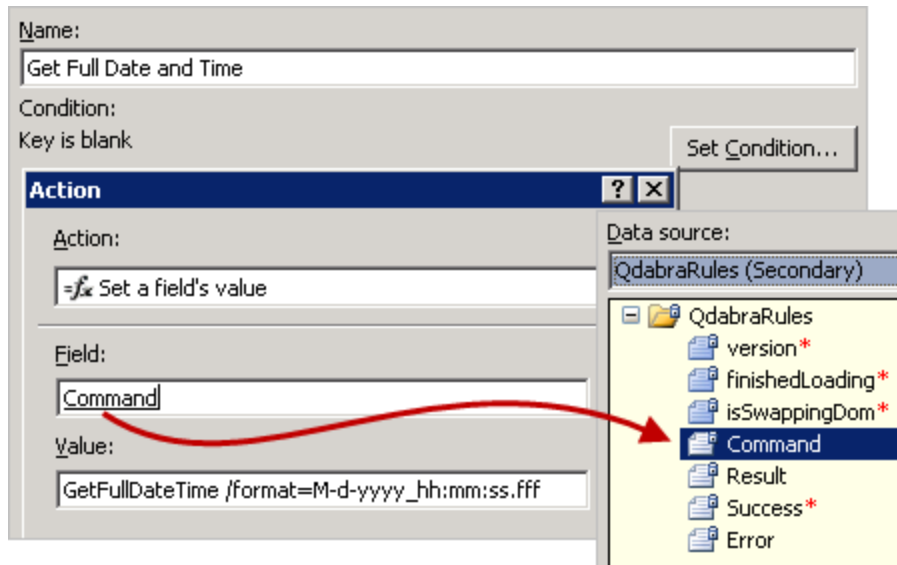
Example

In the following example, we will be concatenating the Form Template Name, Username and the Current Date and Time. Follow these steps to create a sample form that can be used in the InfoPath client:

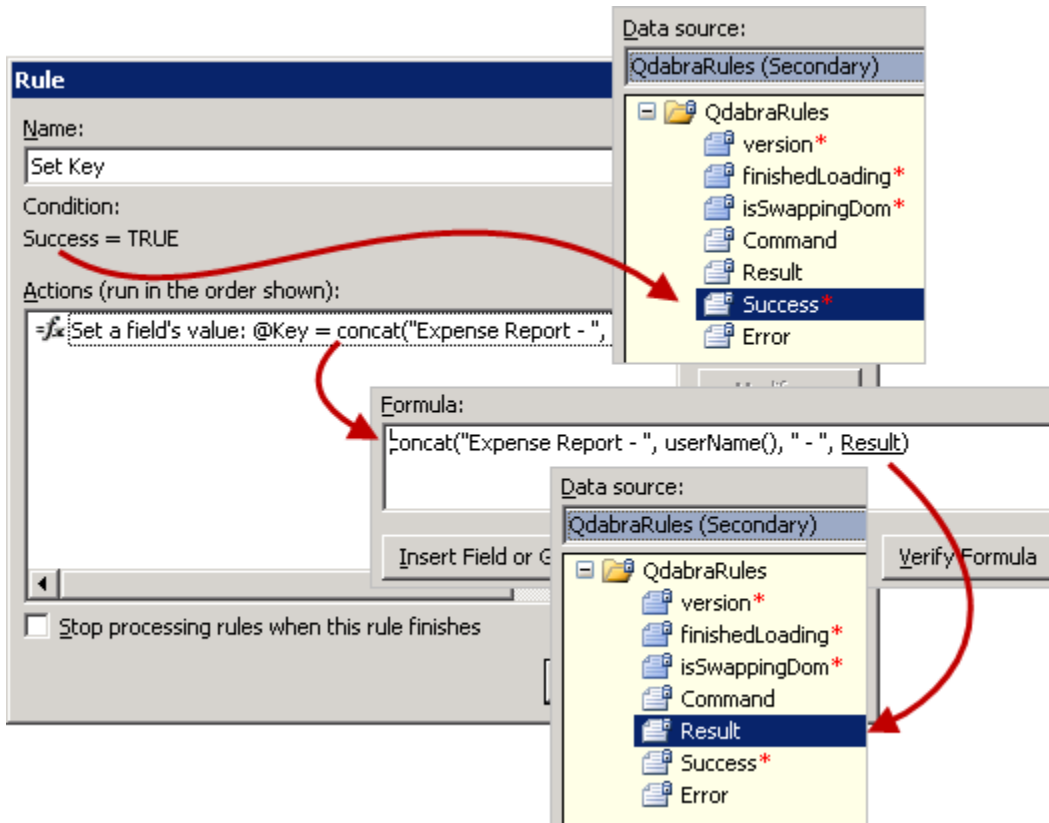
1. Inject qRules in your form (if you haven't already). For more, please refer to the qRules User Guide.
2. Open your form in InfoPath Designer (we'll use the InfoPath 2007 out-of-box sample Expense Report)
3. Add an attribute under root node called **Key**:



4. Modify the **Submit** button changing the *Action* to 'Rules and Custom Code' define these rules:
 - a. *Get Full Date and Time* only runs if **@Key** is blank and has one action:
 - i. Sets qRules Command to **GetFullDateTime /format=M-d-yyyy_hh:mm:ss.fff**
(This command syntax would result to something like 11-17-2010_05:30:07.015)



- b. Set Key only runs if qRules Success is TRUE and has one action
 - i. Set @Key to `concat("Expense Report - ", userName(), " - ", Result)`



The screenshot displays the Qdabra software interface for configuring a rule. The main window is titled "Rule" and contains the following fields:

- Name:** Set Key
- Condition:** Success = TRUE
- Actions (run in the order shown):**
 - Set a field's value: @Key = concat("Expense Report - ", ...)

A "Formula" dialog box is open, showing the formula: `concat("Expense Report - ", userName(), " - ", Result)`. Below the formula is a "Verify Formula" button. Two "Data source" dropdown menus are visible, both set to "QdabraRules (Secondary)". The first dropdown has "Success*" selected, and the second has "Result" selected. Red arrows indicate the flow of data from the dropdowns to the formula and then to the action.

- c. Stop Rules if qRules Fails
 - i. Rule is checked to stop processing others if it fails
 - d. Set Caption always runs
 - i. Sets qRules command to `concat("SetCaption /caption=", @Key)`
 - e. Other rules that submit the form
 - i. If submitting via email: Use @Key for email subject.
 - ii. If submitting to a SharePoint library: Use @Key as the file name.
 - f. Close form
5. Disable **Save and Save As** (in *Tools > Form Options > Open and Save*) to prevent manually overriding the Key for your form name.
 6. Publish your form and submit a few sample forms to test the functionality.



RANDOM NUMBERS

Random numbers aren't always unique so they aren't a good candidate for uniquely naming your form. However, they can be useful within your form to randomize the order of questions in a survey, for example.

Benefits:

- Random numbers can be used to reorder content in your form
- Random numbers can be whole numbers from 1 on up, or decimal numbers

Limitations:

- Random numbers aren't always unique
- Random numbers don't contain any descriptive information

You can generate a random number using qRules **GenerateRandomNumber** command.

Command Overview

GenerateRandomNumber generates a random number allowing setting of upper and/or lower bounds for the random number.

Arguments:

- max (optional): Exclusive upper bound
- min (optional): Inclusive lower bound, must have a maximum number to use

Examples:

- *GenerateRandomNumber /max=100 /min=20*
- *GenerateRandomNumber /max=100*

Randomizing Repeating Content – General Overview

To randomize repeating content in your form, the repeating group must include a sort ID in your main or secondary XML data source. You will need to count the number of items and use qRules' **GenerateRandomNumber** command to randomly pick numbers/items from the list.

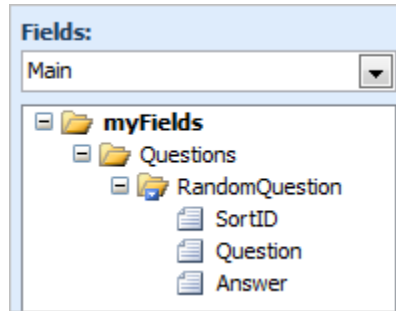
Step-by-step Example

For our example, we will create a form that generates a random question – you can expand this form and turn it into a quiz form or a questionnaire.

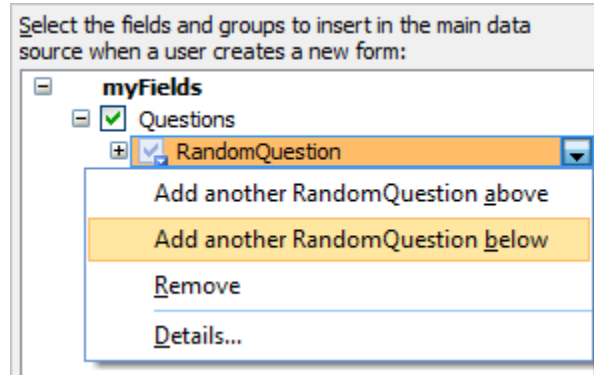
1. Create a blank InfoPath form and inject it with qRules.
2. Open your form in InfoPath Designer.



3. Add the following fields:



4. Drag the **RandomQuestion** node as a repeating table on your canvas and then add a button labeled 'Generate'.
5. Go to the *Data* tab > *Default Values*, right-click on the **RandomQuestion** repeating node and select *Add another RandomQuestion below* (add at least 5).



6. For each node, specify the default values of the **SortID** (1-5) and the **RandomQuestion** (use any default question desired). Leave the **Answer** fields blank.
If you preview your form, you should see something like this:

Sort ID	Question	Answer
1	What was the name of your first pet?	
2	What is the title of your favorite book?	
3	What is your mother's maiden name?	
4	Who is your all-time favorite movie character?	
5	What is the last name of your closest childhood friend?	
<input checked="" type="checkbox"/> Insert item		
		<input type="button" value="Generate"/>



7. Add the following rules to your button:

(These are repeated rules that have the same action executing the *GenerateRandomNumber* qRules command. We repeat the rule 8 times, which seem like the max, and this isn't elegant. It would probably be better to use an XML file with some fields for the logic – set the SortID field to the generated number, if the result corresponds to a question that has already been answered, run the command again, set the field again, and so on...)

- a. Rule #1 – no condition, set the value of the QdabraRules *Command* field to **GenerateRandomNumber /max=6 /min=1**

Condition:
None - Rule runs when button is clicked

Rule type:
Action

Run these actions: *
Set a field's value: Command = "GenerateRandomNumber /max=6 /min=1"

- b. Rule #s 2-8:

Condition: The expression **my:Questions/my:RandomQuestion[my:Answer != "" and my:SortID = xdXDocument:GetDOM("QdabraRules")/QdabraRules/Result]**

Action: Set the value of the QdabraRules *Command* field to

GenerateRandomNumber /max=6 /min=1

Condition:
my:Questions/my:RandomQuestion[my:Answer != "" and my:SortID = xdXDocument:GetDOM("QdabraRules")/QdabraRules/Result]

Rule type:
Action

Run these actions: *
Set a field's value: Command = "GenerateRandomNumber /max=6 /min=1"

8. Right-click on the **RandomQuestion** repeating table, and in the properties dialog, uncheck the box 'Allow users to insert and delete rows'.

Default settings

Allow users to insert and delete rows:
 Edit the default values of the repeating table
 Customize the name of the commands used to insert and delete rows

Show insert button and hint text:



9. Add a conditional formatting to the **RandomQuestion** repeating table to not show rows which **SortID** is not equal to the generated number (QdabraRules *Result* field) and **Answer** is blank:

Condition: *

SortID ≠ Result and

Answer is blank

Rule type:

Formatting

Formatting: *

B
 I
 U
 abc
 A
 ↕

(hidden)

Hide this control

10. Add a conditional formatting to the **Answer** textbox so that blank answers are shaded:

Condition: *

Answer is blank

Rule type:

Formatting

Formatting: *

B
 I
 U
 abc
 A
 ↕

AaBbCcYyZz

11. Preview your form and test – keep clicking on the *Generate* button, answer some questions, etc. You'll get something like this:

Sort ID	Question	Answer
1	What was the name of your first pet?	Sofie
4	Who is your all-time favorite movie character?	
5	What is the last name of your closest childhood friend?	Smith



DOCUMENT IDS

A Document ID (DocID) is a number from 1 on up that records the number of the document. Document IDs are used in server-based systems such as SharePoint to number documents. When you save your form to a SharePoint Library or List, SharePoint will automatically assigns a unique integer to your document or list item.

Benefits:

- DocIDs are short and easy to reference

Limitations:

- DocIDs aren't unique across servers and will cause migration problems later
- DocIDs are hard to insert into your form because the server assigns them. Some developers will query SharePoint to get the largest ID and then set the ID in their form, but this is not a good practice since two documents can be submitted simultaneously. So, to prevent duplicates you have to submit, query for ID, and submit again. To query for ID requires something unique in the form which means you'll probably need a GUID or Key just to make sure you are querying the correct form that you just saved.

Due to the above limitations, we don't recommend DocIDs for your form.

Resources

Clay Cobb has posted a method to set the ID in a form based on SharePoint ID. This method doesn't prevent duplicates but may be good enough for some people out there.

InfoPathDev contains documents on how to use DBXL's DocID feature. qRules comes with a command to simplify setting the DocID on submit to allow users to save their form and continue editing. If the DocID isn't injected after initial save, subsequent saves create duplicate records.



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.

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>

