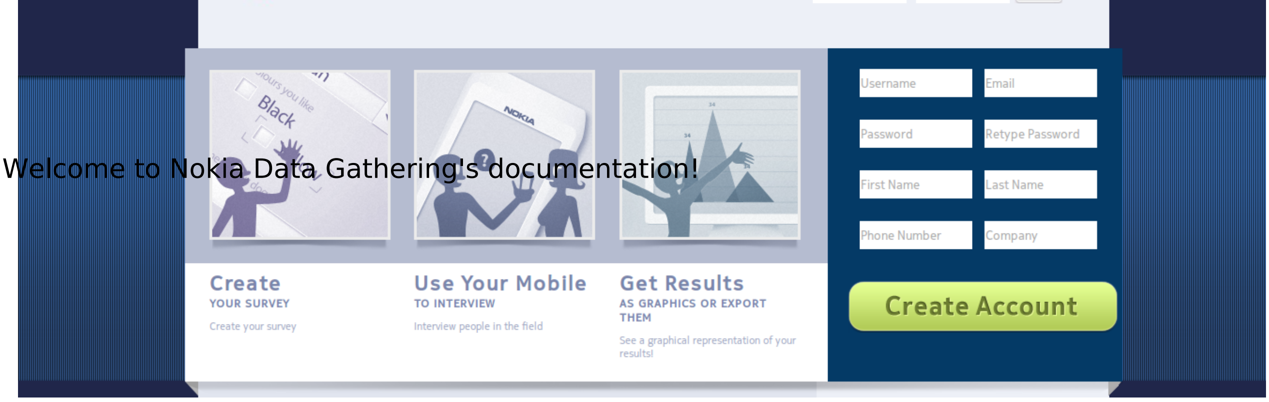
Contents

Welcome to Nokia Data Gathering's documentation!



Mailing List, Source Code, Older Releases and Documentation

Contents:

Introduction

Nokia Data Gathering is a solution that helps organizations to *collect field data using mobile phones* instead of paper forms, PDAs or laptops. Since mobile phones can send data from many remote locations, collected data can be transmitted in near real-time for analysis. This makes collection of data much faster, more accurate and more cost effective to gather especially in remote locations when dealing with critical issues, such as public health, agricultural stock levels, emergency services and alike. The solution consists of *two modules, server and mobile phone*, to enable smooth information transfer from the survey administrators to the field workforce and vice versa. The process includes the creation of questionnaires, their delivery to mobile phones and the subsequent integration and analysis of results. Mobile based data collection offers lots of benefits compared to traditional paper and pen based method as well as compared to data collection with help of a PC:

- *Speed and efficiency*: ability to transmit and process time-critical data quickly. The more remote the location, the greater the improvement
- *Data accuracy*: removes the need for interpretation of hand-written data and potential mistyping of information when it is transferred from paper to legacy systems
- *Reduced operating costs*: reduces duplication of work (data entry) and removes unnecessary travel. Further the costs of delayed decisions can be considerable.
- *Reduced environmental impact*: minimizes the frequency and need for heavy load transportation and carrying papers in and out - especially at remote locations.
- *Usability*: mobile phones as data collection tools are light-weight small devices with long battery life. Furthermore many of them provide GPS in-build and easier to use than PCs while standing/on the move.

Compared to other mobile based data collection tools, Nokia Data Gathering offers high usability (easy to use Java and Windows phone based clients), complete end-to-end system (including mobile and server modules (with integrated survey editor) allowing organizations to host their own server and have security in the knowledge that the data collected by them will remain in-house) and ability to graph and map the results.

Solution Overview

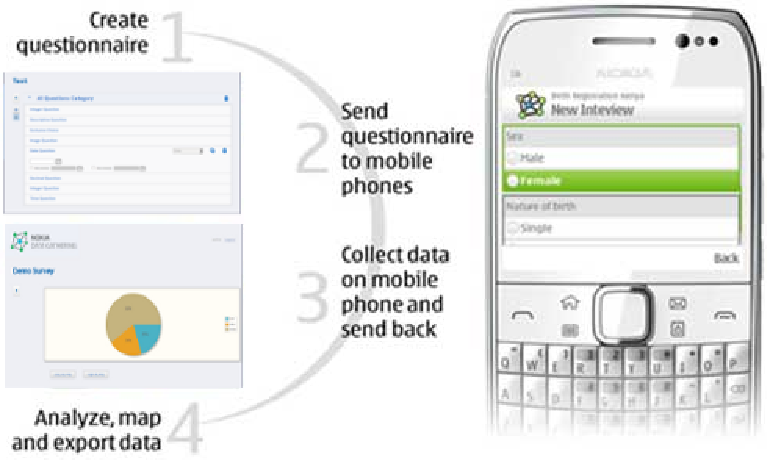
- *Server*: The Server Module is used to create and send surveys to mobile phones, receive interview results, administer users, questionnaires and responses, map data using GPS-based data, graph results and export data. The server can receive interview results in near real-time, provided the field personnel are within range of a mobile voice or mobile data network.

Additionally, the server can be connected to a GSM modem to send messages and receive responses from the mobile phones of the field personnel.

- *Mobile Phone Application*: The Mobile Phone Module is the only part of the system that is visible to the field personnel. It is the software that appears on their mobile phone, presented as an easy-to-fill questionnaire.

The process with Nokia Data Gathering begins with survey creation, which enables the production of tailored questionnaires. The questionnaires can then delivered to the field workforce wirelessly using a normal mobile telecommunications network. Having received the questionnaire(s) on their mobile phones, the field workforce can then use their phones to enter and store the responses to questions. The system also allows to geo-tag data with GPS location information, providing an additional layer of information and helping to validate data. Once the information is collected, the solution enables them to send responses back for instant analysis, again via a mobile network.

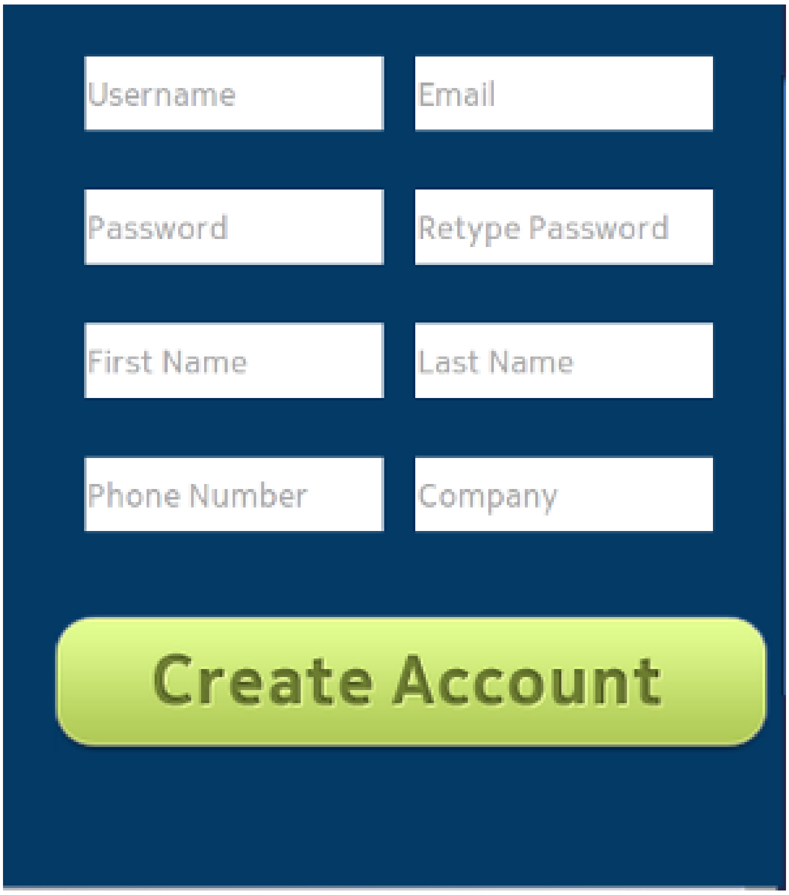
Nokia Data Gathering then is used to create and send surveys to mobile devices via HTTP, receive responses, export data to Excel, administer users, questionnaires and responses, graphically represent survey results and map real-time GPS-based data. It can also send sms messages to users.


Create questionnaire
1
Send questionnaire to mobile phones
2
Collect data on mobile phone and send back
3
Analyze, map and export data
4
Demo Survey
New Inteview
Sex
Male
Female
Nature of birth
Single
Back

Nokia Data Gathering work flow

Getting started

To get started with Nokia Data Gathering Playground, you need to register on the server and enter the data that is requested.

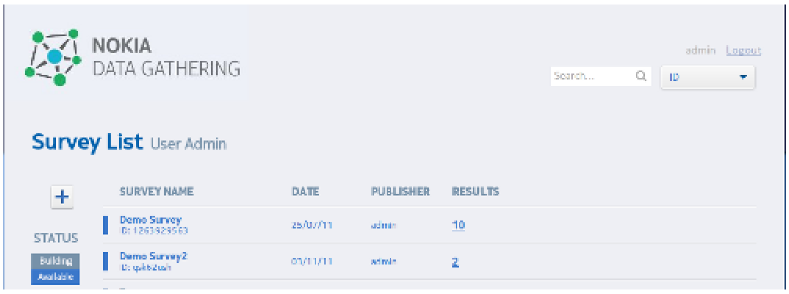


The registration form

After you have filled in the information and submitted, a confirmation email will be sent. The registration is complete after clicking on the link on the email. You can then log in to the server.

Survey List

After logging in the list of surveys that are available for that user are shown

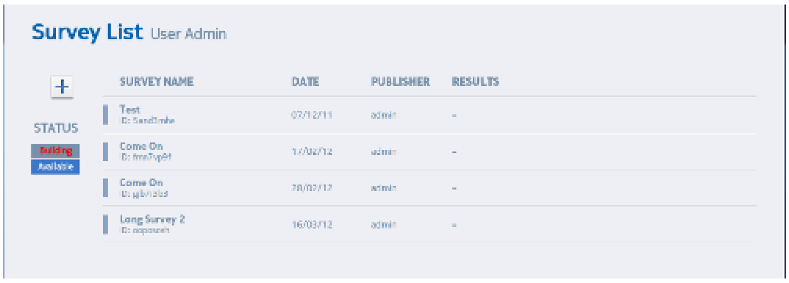


SURVEY NAME	DATE	PUBLISHER	RESULTS
Demo Survey ID: 1263323563	25/07/11	admin	10
Demo Survey2 ID: upkb2ush	07/11/11	admin	2

Survey List

with each survey having a **SURVEY NAME**, **DATE**, **PUBLISHER** and a number of **RESULTS** sent back for that survey. If responses have come back for a particular survey then the number of responses becomes a link to the *Results* page.

Surveys can be filtered by clicking on either **Building** or **Available**



Survey List User Admin

STATUS

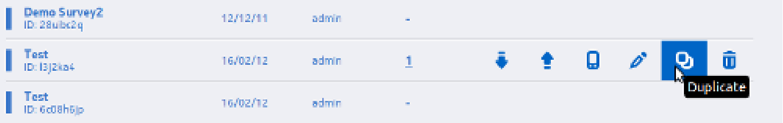
Building

Available

SURVEY NAME	DATE	PUBLISHER	RESULTS
Test ID: [illegible]	07/12/11	admin	-
Come On ID: [illegible]	17/02/12	admin	-
Come On ID: [illegible]	28/02/12	admin	-
Long Survey 2 ID: [illegible]	16/03/12	admin	-

Survey Filter to show only surveys currently being built

and to return to the default list showing all surveys (i.e to turn off the filter) click the red highlighted text. Surveys are made **Available** by being sent to a device. After a survey is made **Available** it can no longer be edited. It can however be duplicated and then edited again. To do this click the duplicate icon on the survey toolbar.



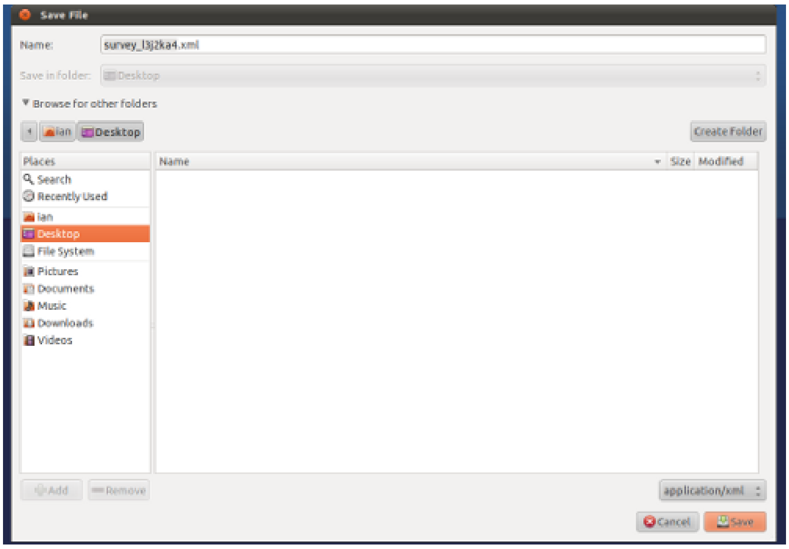
Demo Survey2 ID: 28ubc2q	12/12/11	admin	-
Test ID: l3j2ka4	16/02/12	admin	1
Test ID: 6c08h6jp	16/02/12	admin	-

Duplicate a survey

The new duplicated survey can be edited and saved as needed until it is sent to a device. Then its **Building** status will change to **Available** and no further editing is allowed. Such a process is an easy way to implement simple version control for surveys.

Other functions available in the survey toolbar are (in order) **Download**, **Upload**, **Send**, **Edit** and **Delete**.

Clicking **Download** opens a browser File Save dialog and the raw survey xml can be saved. This is a useful function if for example the survey can only be passed onto a device using USB or bluetooth as may be the case if no cellular network is available.



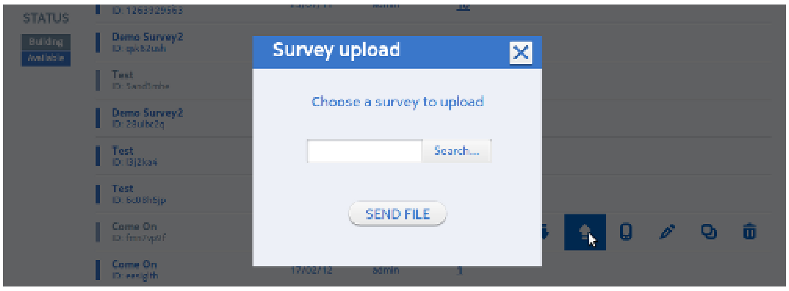
Save File

Name: survey_l3j2ka4.xml

Save in folder: Desktop

Download a survey

Upload is the same process but in reverse and is only available when **Building** surveys.

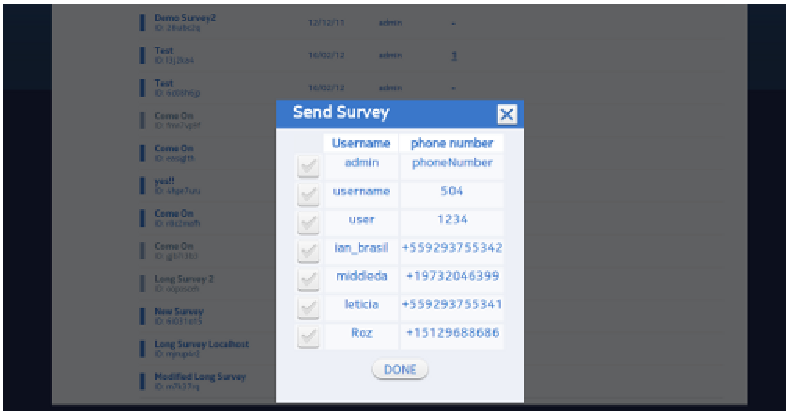


Survey upload

Choose a survey to upload

Survey Upload dialog box

and is a useful function if for example a user has an existing survey created using another Open Rosa compatible survey builder that they wish to use with Nokia Data Gathering. Clicking **Send** presents a list of Users and their corresponding telephone numbers



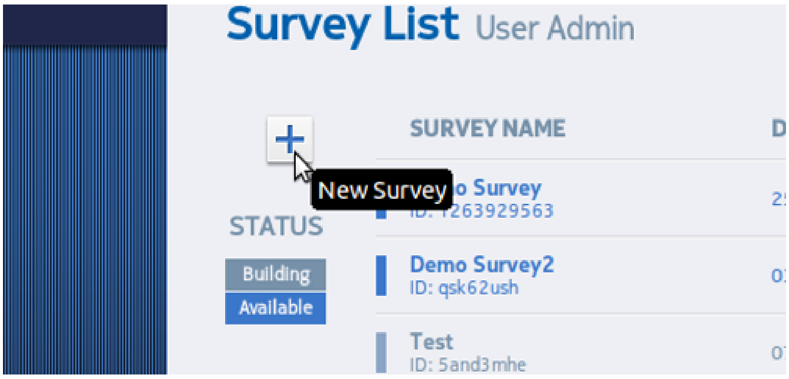
Send Survey

	Username	phone number
✓	admin	phoneNumber
✓	username	504
✓	user	1234
✓	ian_brasil	+559293755342
✓	middleda	+19732046399
✓	leticia	+559293755341
✓	Roz	+15129688686

Send Survey dialog with Usernames and Phone Numbers

select the check box next to the Username and then the Done button to send the survey to that user.

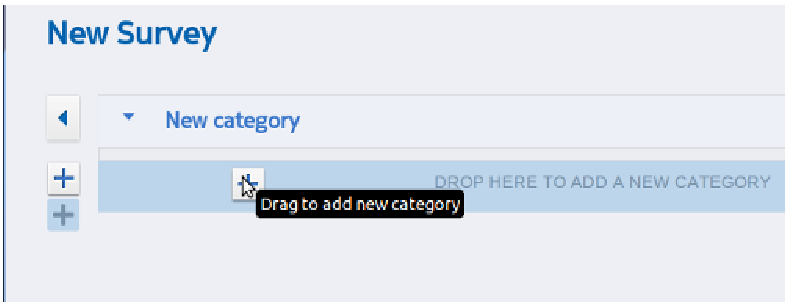
The final button on the survey toolbar is **Edit** and clicking this on a survey in the **Building** state shows the *Editor* which is integrated in Nokia Data Gathering. If rather than editing an existing survey you would like to create a new one click the plus + icon



Survey Creation

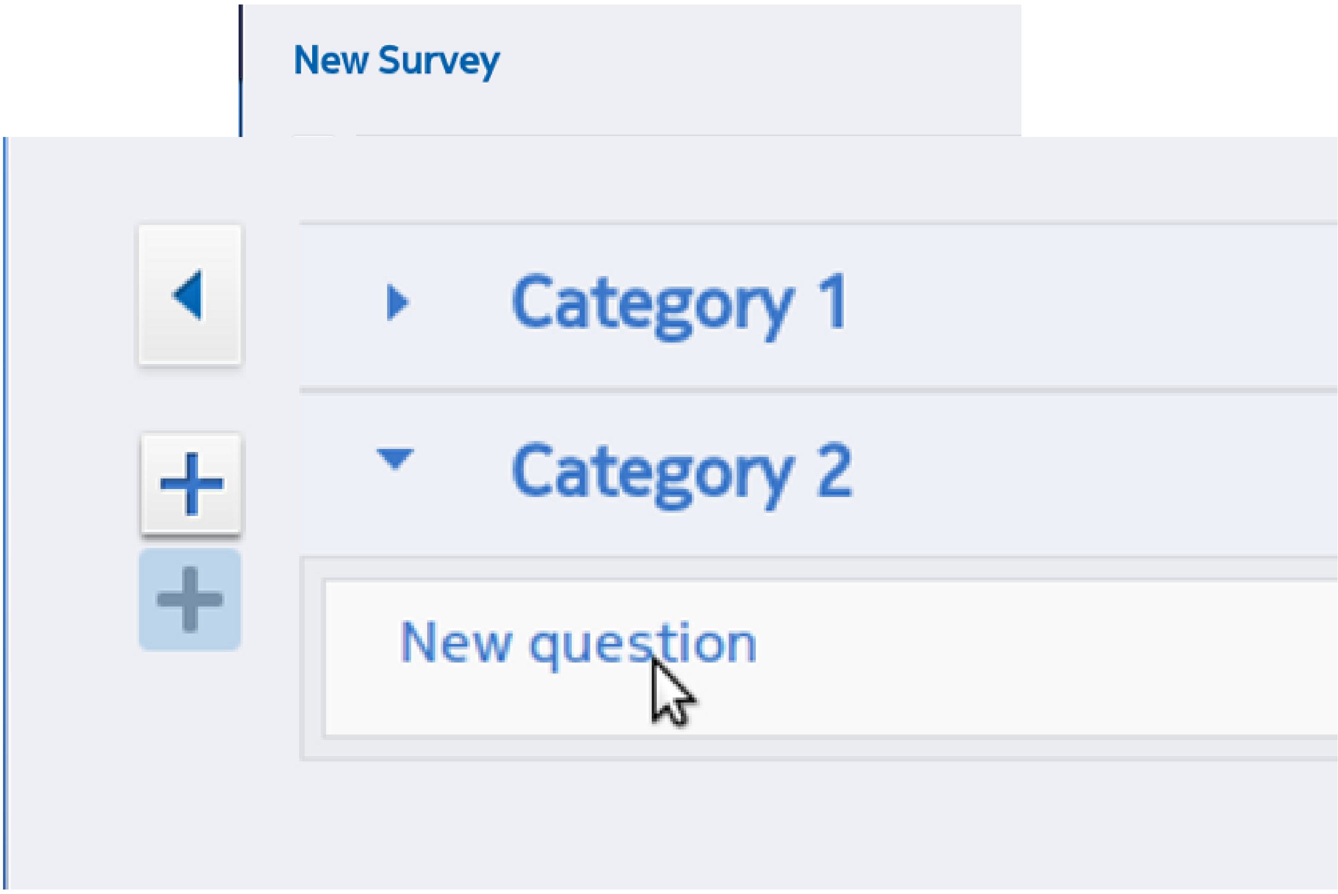
Editor

A new or existing survey has a name which can be clicked to edit and any number of categories and questions. Actually adding categories and questions is done by dragging from the **Drag to add new category** and **Drag to add new question** plus + buttons.



A new category being dragged into a survey

drop either above or below existing categories and the new category will appear. Click the name to edit



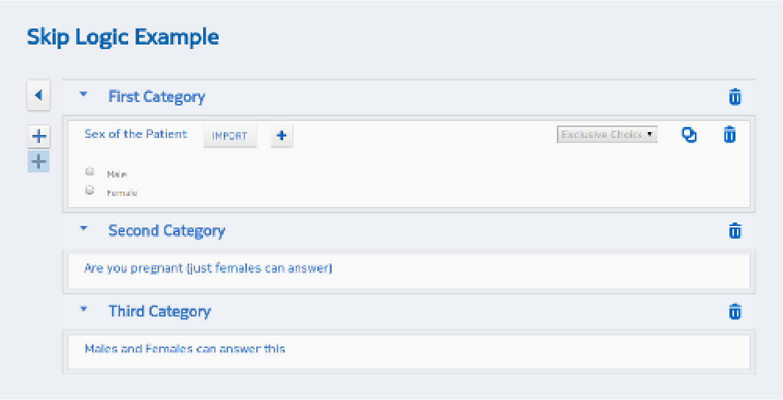
Clicking the question text will open out into the questions toolbar

Skip Logic

Skip Logic which is also called relevance is a way to make a question be skipped based on the value of another question.

The best way to explain how this works is through an example which is taken from the XForms Training Materials Suppose we only want to ask the question 'Are you pregnant?' if the sex of the patient is female. In the *Editor* create a survey with an exclusive choice question, the 'Are you pregnant' question and another random question which all respondents can answer. They do not need to be in separate categories but are shown this way for clarity.

Skip Logic



Skip Logic Example

First Category

Sex of the Patient

IMPORT

Exclusive Choice

Male

Female

Second Category

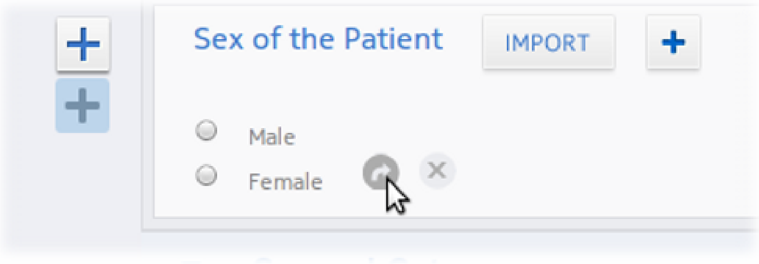
Are you pregnant (just females can answer)

Third Category

Males and Females can answer this

Skip Logic creation in the survey editor

Hovering next to the available options in the exclusive choice question makes an arrow appear.



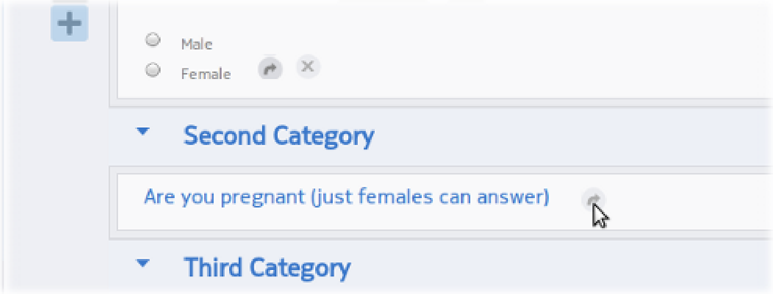
Sex of the Patient

IMPORT

Male

Female

This arrow can now be dragged and dropped on the question to be skipped to if this answer option is selected



Male

Female

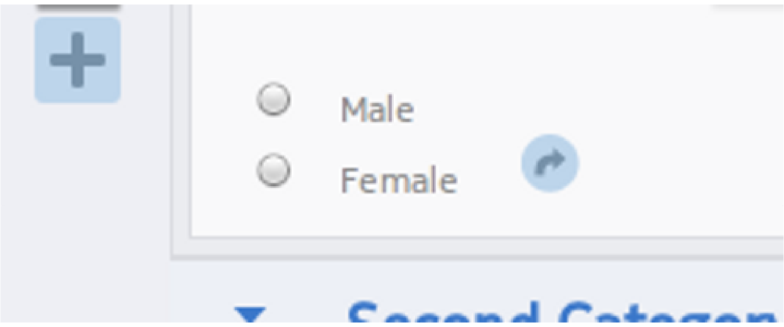
Second Category

Are you pregnant (just females can answer)

Third Category

Arrows can be dragged to represent skip logic relationships

The relationship is then visually represented by a change in the arrow color on the exclusive choice option



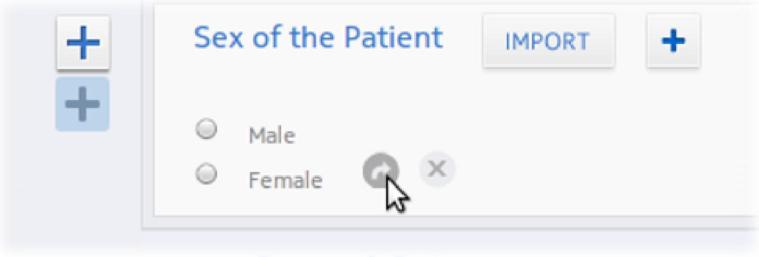
Male

Female

Second Category

Skip Logic realtionship represented by a change in arrow color

Anyone now answering 'Male' to the Sex of the Patient exclusive choice question will now not be able to answer the question 'Are you pregnant'.



Males will still be able to see the question but it will be greyed out and they will be unable to answer it.

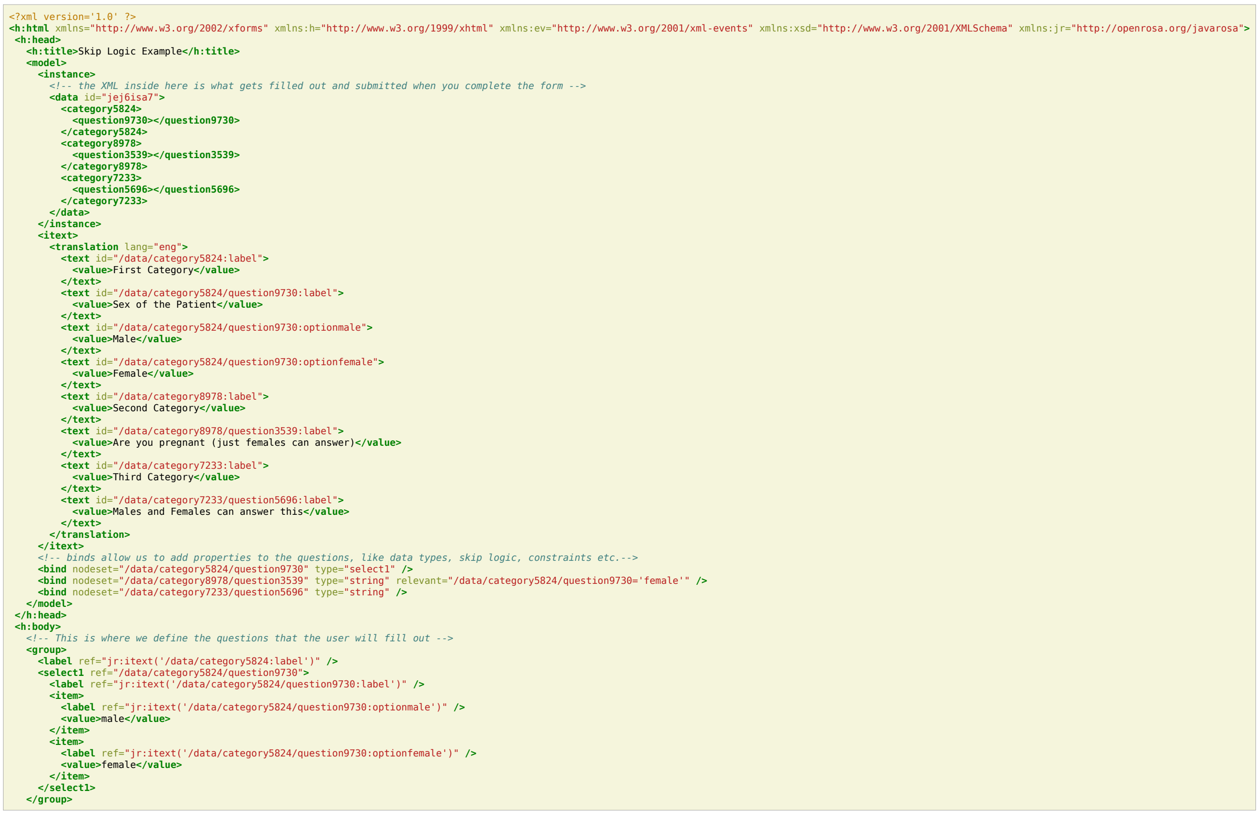
Technical Details

In the actual survey XML the binding is added to the question that is conditionally skipped, not the question whose answer decides whether the other question is skipped. This is an important distinction that can be confusing. For example, here is the bind for the previous example.

<bind nodeset="/data/category8978/question3539" type="string" relevant="/data/category5824/question9730='female'" />

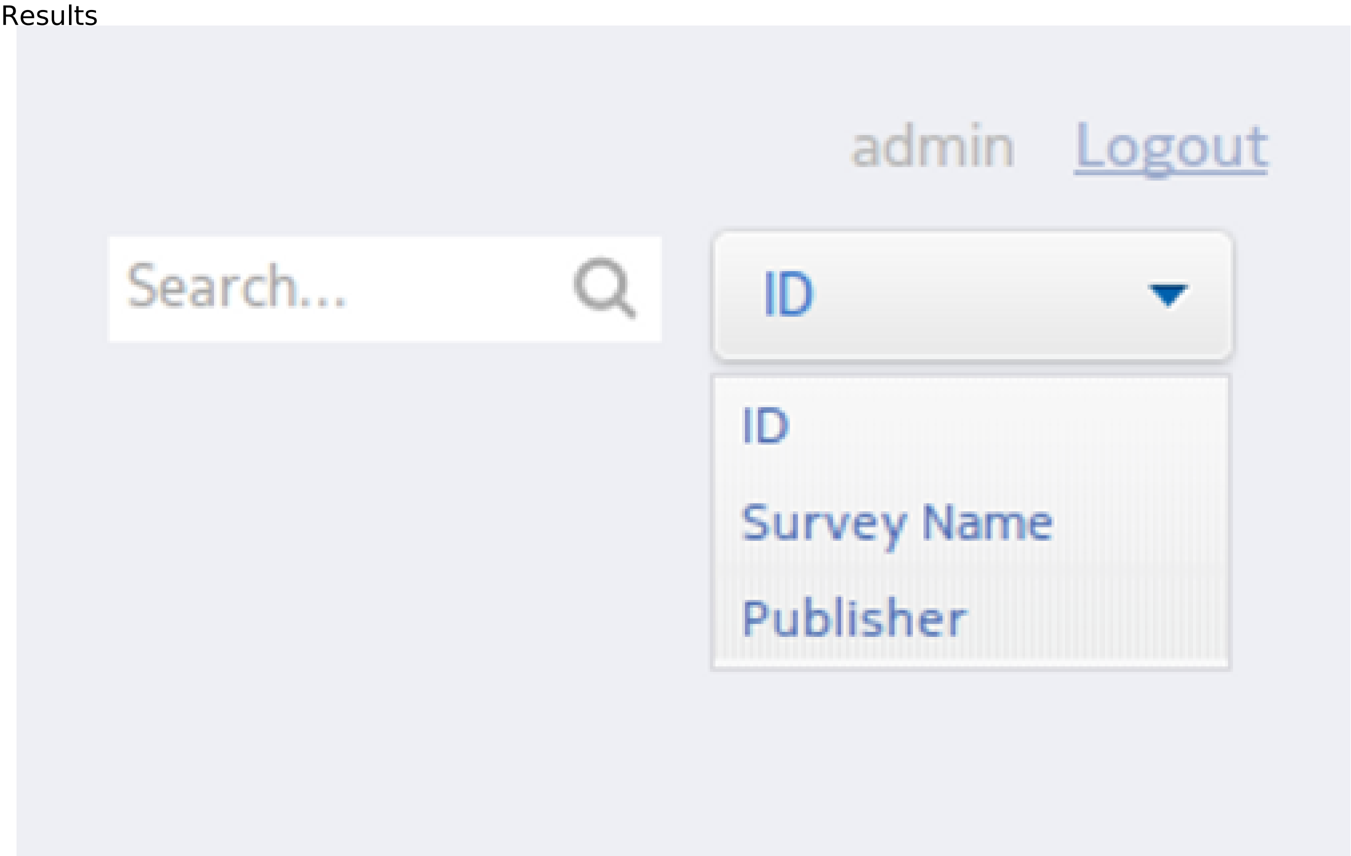
which means bind the question3539 in category8978 to the answer 'Female' of question question9730 in the category5824.

Here is the full commented survey XML for the skip logic example



```xml
<?xml version='1.0' ?>
<h:html xmlns="http://www.w3.org/2002/xforms" xmlns:h="http://www.w3.org/1999/xhtml" xmlns:ev="http://www.w3.org/2001/xml-events" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:jr="http://openrosa.org/javarosa">
 <h:head>
   <h:title>Skip Logic Example</h:title>
   <model>
     <instance>
       <!-- the XML inside here is what gets filled out and submitted when you complete the form -->
       <data id="jej6isa7">
         <category5824>
           <question9730></question9730>
         </category5824>
         <category8978>
           <question3539></question3539>
         </category8978>
         <category7233>
           <question5696></question5696>
         </category7233>
       </data>
     </instance>
     <itext>
       <translation lang="eng">
         <text id="/data/category5824:label">
           <value>First Category</value>
         </text>
         <text id="/data/category5824/question9730:label">
           <value>Sex of the Patient</value>
         </text>
         <text id="/data/category5824/question9730:optionmale">
           <value>Male</value>
         </text>
         <text id="/data/category5824/question9730:optionfemale">
           <value>Female</value>
         </text>
         <text id="/data/category8978:label">
           <value>Second Category</value>
         </text>
         <text id="/data/category8978/question3539:label">
           <value>Are you pregnant (just females can answer)</value>
         </text>
         <text id="/data/category7233:label">
           <value>Third Category</value>
         </text>
         <text id="/data/category7233/question5696:label">
           <value>Males and Females can answer this</value>
         </text>
       </translation>
     </itext>
     <!-- binds allow us to add properties to the questions, like data types, skip logic, constraints etc.-->
     <bind nodeset="/data/category5824/question9730" type="select1" />
     <bind nodeset="/data/category8978/question3539" type="string" relevant="/data/category5824/question9730='female'" />
     <bind nodeset="/data/category7233/question5696" type="string" />
   </model>
 </h:head>
 <h:body>
   <!-- This is where we define the questions that the user will fill out -->
   <group>
     <label ref="jr:itext('/data/category5824:label')" />
     <select1 ref="/data/category5824/question9730">
       <label ref="jr:itext('/data/category5824/question9730:label')" />
       <item>
         <label ref="jr:itext('/data/category5824/question9730:optionmale')" />
         <value>male</value>
       </item>
       <item>
         <label ref="jr:itext('/data/category5824/question9730:optionfemale')" />
         <value>female</value>
       </item>
     </select1>
   </group>
```

Results



User Admin