ORD
OPEN RULES DIALOG

Getting Started

How to develop and maintain Web-based Questionnaires

ORD (OpenRules Dialog) is a software product that allows business analysts to develop and maintain web-based dialogs (questionnaires) with complex interaction logic. This Getting Started tutorial describes how to create a basic ORD questionnaire using a credit card application as an example.

OpenRules, Inc.

October-2019

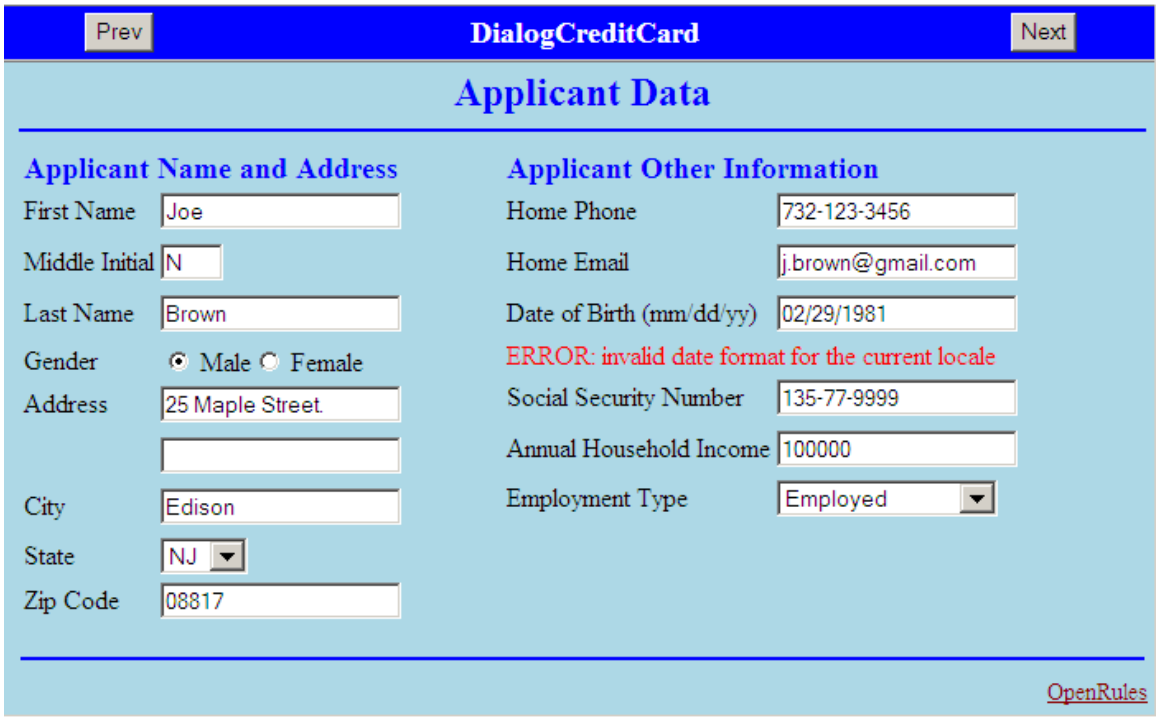
Table of Contents

ORD Introduction

OpenRules Dialog (ORD) is a software product built on top of the open source business rules management system "OpenRules". It allows non-technical people to develop and maintain web-based questionnaires (dialogs) using only Excel and without any necessity to learn different web programming techniques. A questionnaire is a web application that can be described in terms of pages, sections, and different types of questions. Layouts of pages, sections, questions, and complex relationships between them can be expressed in a very natural way using simple and intuitive Excel tables. This document describes how to create a basic ORD questionnaire using a simple Credit Card application as an example.

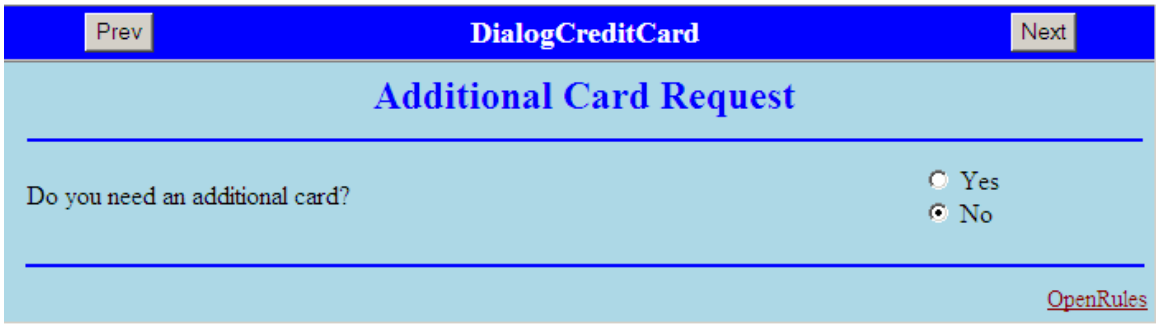
User Scenario

The following screens demonstrate a user session during which a credit card application being created and submitted. The first questionnaire page looks like on this picture:



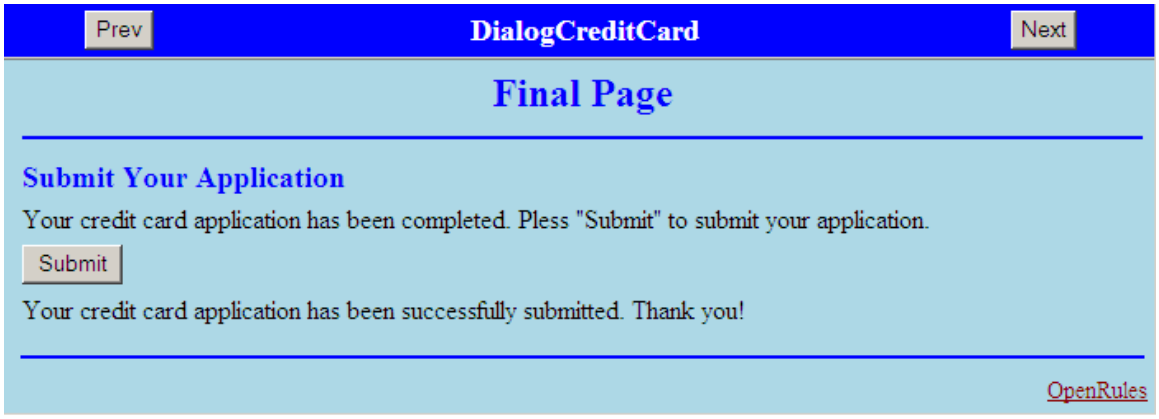
After clicking on the button "Next" a user will receive the error (in red) that was caused by the fact the 1981 is not a leap year. Similarly this page validates formats of Zip Code, Phone and Email, SSN, It also checks that the Household Income is at least 10,000. If no errors found, a user will see and fill in the following page:

If on the first page a user selects other than "Employed" Employment Type this page will be skipped. The next page is:



If a user selects "Yes" this page will be updated to this one (to show hidden additional questions):

And finally after clicking on the "Next", a user will see this page:



From here a user may go back to check his/her answers or click on the button "Submit" to submit this application.

How the Questionnaire is Organized

Your project DialogCreditCard contains all necessary files that needed to create and deploy your questionnaire as a web application. As a questionnaire developer you deal only with two Excel files located in your project directory DialogCreditCard/war/rules:

1) **Main.xls**: describes the structure of your project and allows you to change its look-and-feel (otherwise no changes in this file are required)

2) **Rules.xls**: describes the content and interaction logic of your questionnaire in the following tables:

    a. **pages** – a list of all your pages

    b. **sections** – a list of all your sections

c. **questions** – a list of all your questions

d. **answers** – a list of defaults and possible answers. If there are no answers you still need to keep this table (even when it is empty). This table may use answer lists defined in the table similar to the one defined in the table "possibleAnswers" (this is not a key word and standard lists may be organized differently)

e. **autoResponses** – a list of all your automatically calculated responses. If you do not have auto-responses you still have to keep this table after deleting all rules (rows starting with the third one)

f. **navigationRules** – this decision table defines special conditions when the default sequential order of pages, sections, or questions is violated and different order of pages is required. If you do not have special navigation rules, you may omit this rules table

g. **updateRules** – this decision table defines special conditions for pages updates such as: a hide/show question, a hide/show section, a response to a question X with an answer Y, etc. If you do not have special update rules, you may omit this rules table.

h. **customQuestions** (optional) – this table defines application-specific layouts for custom questions.

Pages

ORD considers any questionnaire as a sequence of web pages through which a user can navigate using actions "Next" and "Prev". Here is the Excel table "**pages**" that has been used for this questionnaire:

Rules pages extends pagesTemplate					
#	Page ID	Page Name	Hidden	Section Column 1	Section Column 2
1	ApplicantData	Applicant Data		ApplicantNameAddress	ApplicantOtherInfo
2	ApplicantEmployment	Employment Data		EmploymentData	
3	AdditionalCardRequest	Additional Card Request		RequestAdditionalCard	
4				Additional Card Data	
5	FinalPage	Final Page		FinalSection	

As you may guess, this table describes our 4 pages with the following unique IDs:

Page 1. ApplicationData

Page 2. ApplicantEmployment

Page 3. AdditionalCardRequest

Page 4. FinalPage.

The Page 1 consists of 2 sections, and ApplicantOtherInfo is allocated on the right of the ApplicantNameAddress. Each page may have (or may not have) a title that will be shown on

the top of the page. You may use any string as a page ID as long as all page IDs are unique. The order of pages in this table reflects the natural navigation order through them assuming that every page has navigation buttons "Next" and "Prev". When the order of pages depends on the entered answers or other factors (for example the page "ApplicantEmployment" may be skipped if the applicant is not employed), these special cases could be described in special navigation rules (see below).

Note. The very first column is usually used for decorative purposes only: it may contain a page order number that has no particular meaning and may be left empty. However, you cannot remove this column or merge cells inside it.

Sections

The previous table "pages" defines a relative layout for sections inside every page. The sections themselves are defined in the proper Excel table "**sections**" that has been used for this questionnaire:

Rules sections extends sectionsTemplate					
#	Section ID	Section Name	Hidden	Question Column 1	Question Column 2
1	ApplicantNameAddress	Applicant Name and Address		ApplicantFirstName	
1				ApplicantMiddleInitial	
1				ApplicantLastName	
1				Gender	
1				AddressLine1	
1				AddressLine2	
1				City	
1				State	
1				ZipCode	
2	ApplicantOtherInfo	Applicant Other Information		HomePhone	
2				HomeEmail	
2				DOB	
2				SSN	
2				Income	
2				EmploymentType	
3	EmploymentData	Employment Data		Employeer	
3				BusinessAddressLine1	
3				BusinessAddressLine2	
3				BusinessCity	
3				BusinessState	
3				BusinessZipCode	
3				BusinessPhone	
3				BusinessEmail	
4	RequestAdditionalCard			NeedAdditionalCard	
4	Additional Card Data	Additional Card Data	Yes	NameOnTheCard	
4				AdditionalDOB	
4				AdditionalSSN	
5	FinalSection	Submit Your Application		Completed	
5				SubmitApplication	
5				Submitted	

This table defines all sections and their content (questions). For example, the section ApplicantOtherInfo consists of 6 questions that will be placed one under other. Each section may have (or have not) a title that will be shown on the top of the section. You may use any string as a section ID as long as all section IDs are unique (across all pages). The column "#" is similar to the one for pages and can contain any information or be empty.

Questions

The previous table "section" defines a relative layout for questions inside every section. The questions themselves are defined in the following Excel table "**questions**":

Rules **questions** extends questionsTemplate					
Question Id	Question Name	Question Type	Size	Hidden	Validation
ApplicantFirstName	First Name	TextBox			
ApplicantMiddleInitial	Middle Initial	TextBox	2		
ApplicantLastName	Last Name	TextBox			
Gender	Gender	RadioButton			
AddressLine1	Address	TextBox			
AddressLine2		TextBox			
City	City	TextBox			
State	State	ComboBox			
ZipCode	Zip Code	TextBox			Range 1 99999
HomePhone	Home Phone	TextBox			REGEX [0-9]{3}-[0-9]{3}-[0-9]{4}
HomeEmail	Home Email	TextBox			EMAIL
DOB	Date of Birth (mm/dd/yy)	TextBox			DATE
SSN	Social Security Number	TextBox			SSN
Income	Annual Household Income	TextBox			Range 10000 10000000
EmploymentType	Employment Type	ComboBox			
NeedAdditionalCard	Do you need an additional card?	RadioButtonVertical Submit			
NameOnTheCard	Name on the card	TextBox			
AdditionalDOB	Date of Birth (mm/dd/yy)	TextBox			DATE
AdditionalSSN	Social Security Number	TextBox			SSN
Employeer	Employeer	TextBox			
BusinessAddressLine1	Address	TextBox			
BusinessAddressLine2		TextBox			
BusinessCity	City	TextBox			
BusinessState	State	ComboBox			
BusinessZipCode	Zip Code	TextBox			Range 1 99999
BusinessPhone	Business Phone	TextBox			Phone
BusinessEmail	Business Email	TextBox			Email
Completed	Your credit card application has been completed. Pless "Submit" to submit your application.	Message			
SubmitApplication	Submit	ActionButton			
Submitted	Your credit card application has been successfully submitted. Thank you!	Message			

You may use any string as a question ID as long as all question IDs are unique (across all sections and all pages). There are no limits for question names just you have to keep in mind that long names could affect automatically adjusted question/section layouts.

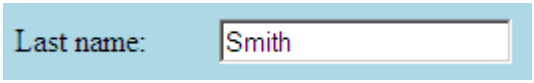
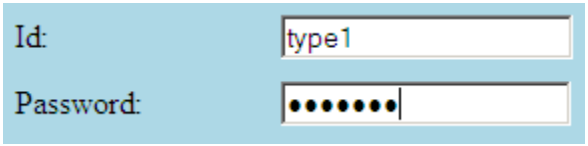
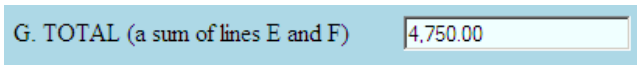
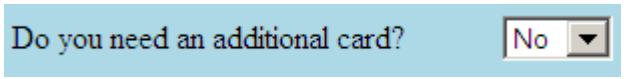
The columns "Size" allows you to change the size (number of displayed characters) in a TextBox's input field. The default size is 20. The validation criteria are described below.

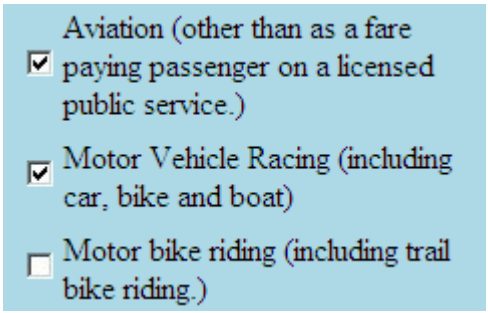
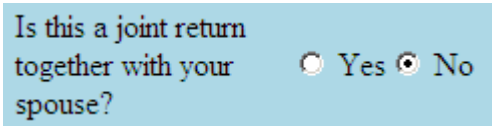
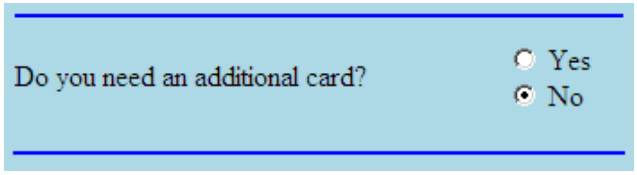
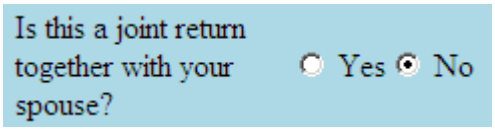
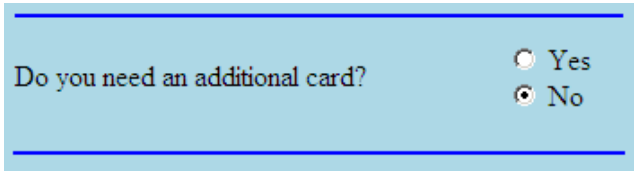
There are no limits to the number of questions and usually all questions for all sections are defined in the table with the name "**questions**". If there are too many questions you may split them in separate tables that look like this one but have different names, for example "questions1" and "questions2". Then you have to add another table with the name "questions" that will invoke these two tables:

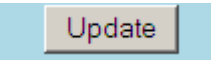
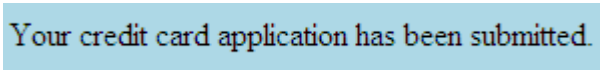
Method void **questions**(Section section, String questionId)
questions1(section, questionId); questions2(section, questionId);

Question Types

The current version of OpenRules Dialog supports the following question types:

Question Type	Comment	Example
TextBox	A regular question with a label (name) and an input field of a certain size	
Password	Similar to TextBox but the entered text would be display as stars	
ReadOnly	Similar to the TextBox but the input field is automatically calculated and cannot be changed (indicated by a slightly different background – Azure)	
TextArea	Similar to the TextBox but the input field may consist of several lines	
ComboBox	Label with input that allows selection from possible answers	
ComboBox Submit	The same as a regular ComboBox but refreshes the page upon a selection	 Selection of "Yes" may show a hidden information for an additional card

Question Type	Comment	Example
CheckButton	Allows the selection of one or more multiple alternatives. Each alternative should be defined by a separate check button. The question's name is used as the check button text. The question's ID is used as a part of the name of the proper action, e.g. Check-Aviation or Uncheck-Aviation	
RadioButton	Allows the selection of one choice from among multiple alternatives	
RadioButton Vertical	The same as a regular RadioButton but places the choices vertically	
RadioButton Submit	Allows the selection of one choice from among multiple alternatives and submits this choice to the server to refresh other pages, e.g. to hide/show related questions	
RadioButton VerticalSubmit	The same as a regular RadioButtonSubmit but places the choices vertically	

Question Type	Comment	Example
ActionButton	Creates a push-button whose name could be associated with some action defined in navigation rules.	 If no action is defined, pushing (selecting) the button will force the page to be refreshed with a possible recalculation of all auto-responses
ActionImage	Creates a push-button with a custom image whose path is defined by the question's name. The question ID is used to define the last user action that could be used in the navigation rules.	 An image that represents a delete button
ActionHyperlink	Creates a hyperlink whose text corresponds to the question's name. The question ID is used to define the last user action that could be used in the navigation rules.	 Can be used to represents different menus
Message	Displays a text defined as the question name	
Empty	Creates an empty question that can be used as a cell filler	
Custom	This question can be defined by a customer using the rules table "customQuestions"	 This custom hyperlink generates and displays a PDF document base on the answered questions

Input Validation

You may associate a validation criterion with any TextBox – see for example "SSN" and "Range 1000 100000000" in the above table "questions". ORD will compare the entered answer with the question's validation criterion, and if it is violated, ORD will place an error message (in red) right below the invalid answer. Here is the list of currently supported validation criteria:

Validation Criterion	Comment	Example of the Error Message
Range <from> <to>	Validates that the answer to the question is an integer or real number that should be within the interval [from;to]	ERROR: The answer should be within 1 and 100
DATE	Validates that the answer is a valid date for the current locale, e.g. 20/5/2009 is invalid for the US locale	ERROR: invalid date format for the current locale
EMAIL	Validates that the answer is a valid email address	ERROR: invalid email address
URL	Validates that the answer is a valid URL address	ERROR: invalid URL address
SSN	Validates that the answer has a valid Social Security Number format	ERROR: SSN should have a format 999-99-9999
CREDITCARD	Validates that the answer has a valid CREDIT Card Format	ERROR: invalid credit card number
REGEX <pattern>	Validates if the answer matches the regular expression defined by regular expression "pattern", e.g. SSN corresponds to the regular expression "[0-9]{3}-[0-9]{2}-[0-9]{4}"	ERROR: <answer> doesn't match the pattern <pattern>
USERID	Validates if the answer is a valid user ID by calling a custom method "*validateUserID*"	ERROR: invalid user ID
PASSWORD	Validates if the answer is a valid password by calling a custom method "*validatePassword*"	ERROR: invalid password

More validation criteria could be added without programming (see the file DialogForms.xls).

Another way to validate user input is to add special conditions directly in the "navigationRules" – see an example in the Dialog1040EZ tutorial.

Answers (default and possible)

You may define the default and possible answers for all or some questions using an Excel table similar to this one:

Data Answers answers		
Question Id	**Default Answer**	**Possible Answers**
EmploymentType	Employed	employmentTypes
NeedAdditionalCard	No	yesno
State	MA	USstates
BusinessState	MA	USstates
ZipCode	11371	
Income	100000	
BusinessZipCode	11371	
Gender	Female	gender

Here the possible answers "USstates" and "yesno" refer to the custom Excel Data table "**possibleAnswers**":

Data PossibleAnswers possibleAnswers	
id	choices
ID	**Answers**
employmentTypes	Employed
	Unemployed
	Full Time Student
	Retired
gender	Male
	Female
yesno	Yes
	No
USstates	AL
	AK
	AR
	AZ
	CA
	CO
	CT
	…
	WY

Automatically Calculated Responses

Answers to some questions could be automatically calculated based on the answers to other questions and possible other information (for example, data coming from a database). This particular questionnaire does not include auto-responses and the proper table is empty. However, rules for auto-responses could be quite complex and may look like below:

Rules **autoResponses** extends autoResponsesTemplate	
Question Id	**Auto Response**

AdjustedGrossIncome	{
	double wages = d.getDoubleAnswer("Wages");
	double taxableInterest = d.getDoubleAnswer("TaxableInterest");
	double unemploymentCompensation = d.getDoubleAnswer("UnemploymentCompensation");
	double answer = wages + taxableInterest + unemploymentCompensation;
	format(answer);
	}

A	{
	double wages = d.getDoubleAnswer("Wages");
	format(wages + 250);
	}

Navigation Rules

ORD provides an automatic support for navigation between and inside pages. The order of pages in the table "**pages**" reflects the natural navigation order assuming that every page has navigation buttons "Next" and "Prev". When the order of pages, section, and questions depends on the entered answers or other factors, these special cases could be described in special navigation rules table "navigationRules". Here is the proper table for our questionnaire:

Rules **navigateRules** extends navigateDialogTemplate					
C1	C2	C3			A1
IF Current Page is	**AND Action is**	**AND**			**THEN Go to Page**
		Answer to Question	**Is or IsNot**	**Value**	
ApplicantData	Next	EmploymentType	IsNot	Employed	AdditionalCardRequest

It has only one rule that directs ORD to skip the page ApplicantEmpoyment and go to the page "AdditionalCardRequest" if the answer to the question "EmploymentType" is not "Employed".

Update Rules

ORD allows a questionnaire designer to define special conditions and execute the proper actions when the content of certain pages is updated by a user. This page updating logic can be defined in the decision table "***updateRules***" that for our questionnaire looks like this one:

Rules updateRules extends updateDialogTemplate									
C1	C2	C3			A1		A2		A4
IF Current Page is	AND Action is	AND			THEN Hide/Show Question		AND Hide/Show Section		AND Set Dialog Status
		Answer to Question	Is or IsNot	Value	Hide	Question	Hide	Section	
AdditionalCardRequest		NeedAdditionalCard	Is	No			Hide	Additional Card Data	Additional Card Data is hidden
AdditionalCardRequest		NeedAdditionalCard	Is	Yes			Show	Additional Card Data	Additional Card Data is shown
FinalPage	Submit				Show	Submitted			

The first and second rules hide/show the section "Additional Card Data" based on an answer to the question "NeedAdditionalCard" entered on the same page "AdditionalCardRequest".

The update rules are organized in such way that allows you to effectively implement show/hide features on any level: page, section, and question. The standard OpenRules class Dialog supports the following methods:

- hidePage(String pageID)
- showPage(String pageID)
- hideSection(String sectionID)
- showSection(String sectionID)
- hideQuestion(String questionID)
- showQuestion(String questionID)

There are more choices to represent navigation rules that are described in the proper template in the standard ORD file DialogRules.xls. You may use more convenient columns "Hide/Show Section" and "Hide/Show Question", invoke any standard or custom Java methods from the column "Execute Code", and much more.

Note. For efficiency reasons the hidden pages, sections or questions are not analyzed during the page "refresh" that occurs every time during a user session when ORD delivers a new page or updates the current one.

Custom Questions and Actions

Along with predefined question and action types you may create your own questions/actions using the type "Custom". To do this you have to define the proper layout in the table "**customQuestions**". This particular questionnaire does not include custom questions.

CUSTOMIZATION

You may essentially customize your application by changing only layout tables in the file Main.xls. You also may create your own version of the standard library "openrules.forms.lib" and completely customize the generated questionnaire (web application).

Changing Look&Feel

To customize a look&feel of your questionnaire, it is usually enough to change layouts of the main graphical form defined in the worksheet "Main Layout" inside the file Main.xls. Here is the default main layout for the DialogCreditCard project:

Layout TableLayout mainLayout(String dialogName)
mainTitleBar(dialogName)
mainCurrentLayout()

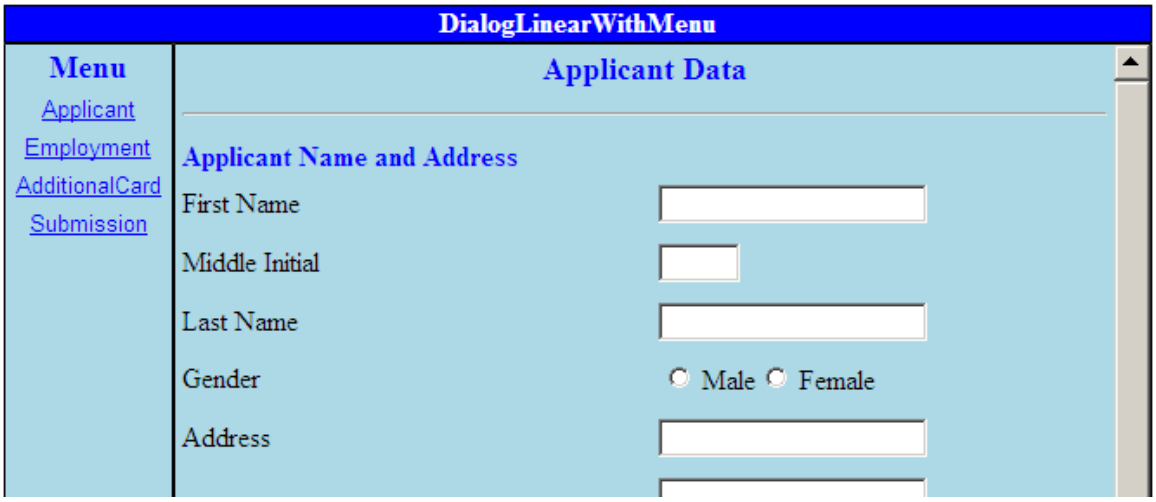
Layout TableLayout mainTitleBar(String dialogName)		
actionButton("Prev")	<td style="color:white;font-size:medium;font-weight:bold"> <C> dialogName </C> </td>	actionButton("Next")

Layout TableLayout mainCurrentLayout()	
 	dialog().nextLayout
 	footerLayout()

Layout TableLayout footerLayout()	
<C> dialog().status </C>	 OpenRules

The name of the main table "**mainLayout**" is fixed but its content can be essentially changed. Basically this layout defines a title bar with two push-buttons "Next" and "Prev" and a footer. The actual content of every page is defined automatically as *dialog().nextLayout* which is calculated every time when any user action is submitted to the server. Our rules table "navigationRules" does exactly this – specifies the next layout.

You may completely change the "mainLayout" without any changes in the questionnaire content and interaction logic. For example, along with the project DialogCreditCard, the standard ORD installation also includes the project "**DialogWithMenu**" created based on the DialogCreditCard but with a different "mainLayout". Here is a new look&feel for the Page 1:



Here is the corresponding layout from Main.xls:

Layout TableLayout mainLayout(String dialogName)	
<td style="color:white;background-color:blue;font-size:medium;font-weight:bold"> <C> dialogName </C> </td>	
<td width="10%" style="vertical-align:top"> <C> menuLayout() </C> </td>	<td width="90%"> <C>mainCurrentLayout() </C> </td>
footerLayout()	

Layout TableLayout menuLayout()
<td style="color:blue;font-size:large;font-weight:bold"> Menu </td>
actionHyperlink("Applicant")
actionHyperlink("Employment")
actionHyperlink("AdditionalCard")
actionHyperlink("Submission")

Layout TableLayout mainCurrentLayout()
dialog().nextLayout

Layout TableLayout footerLayout()	
<C> dialog().status </C>	 OpenRules

Predefined Rules and Forms

The basic ORD concepts such as Dialog, Page, Section, Question, and Answer are defined as Java classes inside the standard OpenRules package "*com.openrules.forms.gui.jsp*". Thus, they are automatically included in the standard OpenRules releases (starting from 5.3.2).

All standard forms (layouts) and rule templates are defined in the Excel files supplied with the standard OpenRules project "*openrules.forms.lib*":

- Dialog.xls
- DialogMain.xls
- DialogForms.xls
- DialogRules.xls
- Validators.xls.

The proper style sheets are defined in the file openrules.forms.lib/css/lib.css, but can be overwritten by project specific style sheets such as DialogCreditCard/war/css/project.css.

Integrating with Existing Software

You may integrate your questionnaire with any 3rd party Java package by adding the proper import-statements to the Environment table in Main.xls. If you want to use your own Java object inside your custom rules, you may add these objects to the instance of the class Dialog that is inherited from the standard Java HashMap. For example, in the file "index.jsp" you may create your own object "customer" and add it to the dialog as follows:

```
Dialog.put("customer",customer);
```

Then inside your Excel rules you always can get an access to this object by using something like this:

```
Customer customer = (Customer) dialog().get("customer");
```

This mechanism gives you a complete control over the integration of the Excel-based questionnaire with your Java application.

If necessary ORD allows you to bring the entire power of OpenRules Forms – see http://www.openrules.com/docs/man_forms.html.

ATTACHMENTS

DialogCreditCard: Main.xls
DialogCreditCard: Rules.xls