CONFIRM

- User Manual -

version 0.9

July 5, 1999

The information on these pages is preliminary and may contain reference to features that are not yet available. Specifications are subject to change without notice. Examples depicted herein are fictitious.

TABLE OF CONTENTS

1 Introduction

1.1 What is CONFIRM?

CONFIRM is Future Information Research Management (FIRM) AS' integrated platform for organizing, defining, conducting, and reporting on quantitative research projects. The system has a proven record of accomplishment for several types of research projects. It is particularly attractive for surveys that involve a lot of co-ordination of resources, such as longitudinal and comparative data collection from international markets. You achieve cost reductions on any scale, though, even on small projects that are short term and ad hoc in nature.

A typical research project consists of a set of related activities (project definition, questionnaire design, data collection, and reporting to name a few). Without proper system support, performing each task and the transition from one activity into the next may be both cumbersome and costly. *CONFIRM* addresses problems that waste hours of a researcher's time every day, such as manually moving data between different and more or less compatible systems, or distributing files and reports to involved parties.

- *CONFIRM* is **integrated**. It supports the entire research process from the design phase until delivery of the final report. You do not use one system for designing your questionnaires, a second for collecting data, a third for analysis, and a fourth for presenting the results. It is all there in the same package. This rule applies even if you opt to combine several methods of data collection. *CONFIRM* eliminates the need for exporting data from a system X and then importing, or even worse punching, them into a system Y. You keep everything you need and all that relates to your projects in one place.
- *CONFIRM* makes you **faster.** The system's high level of integration has allowed us to automate several jobs that traditionally consume a lot of time. You also gain access to a system that serves as a central repository for all your work. The potential for reuse is tremendous and you never have to do the same job twice.
- *CONFIRM* delivers **quality.** Tasks that need manual labor or that require movement of data between system boundaries are likely points of introducing errors. Because of the high level of integration and automation, the results you produce are less prone to error.
- *CONFIRM* is **flexible**. It can be applied to diverse multilingual research problems and gives you great freedom in the design of your surveys. Project artifacts such as questionnaires and reports are easily customized for each customer.
- *CONFIRM* is an **online application.** Because it is on the Internet, you and your customers can access it from all over the world, 7 days per week, 24 hours per day. All you need to use it is a free Web browser.
- *CONFIRM* is always **brand new**. You always run the latest version of our applications. The system resides on FIRM servers and all improvements we put in to the system are immediately available to you.
- *CONFIRM* is **inexpensive.** The system's centrally managed architecture implies that you save investments in server hardware and software. You do not have to spend money to lease high capacity Internet lines. Last, but not least, you can say farewell to the management expenses that typically constitute the main bulk of the lifetime cost of a computer system.
- *CONFIRM* is **secure.** Regardless of what many people say, the Internet is a secure communications medium. Network traffic travel from your organization's local network, through a network of lines leased from public telephone network providers, before reaching another locally controlled network at the other end of the communication stream. Using the Internet for communication is just as safe as placing a telephone call. As long as you trust the server on the other end of connection, very few things can go wrong; the probability of someone listening in on your traffic is very small indeed. FIRM also offers you the possibility to communicate with *CONFIRM* over the Secure Sockets Layer (SSL), an encrypted protocol that further secures all traffic between you and the system.

Most computer security breaches on the Internet, or anywhere else for that matter, are the result of poor server management that allows perpetrators to break into a system and then disclose or destroy information. Since we at FIRM specialize in providing an Internet service, you should feel confident that we have the necessary skill and pay constant attention to all issues that could endanger the security of your data.

1.2 About this manual

A quantitative research project involves several differnt *roles*, ranging from the questionnaire programmer to the telephone interviewer. CONFIRM offers different user interfaces for different roles.

New user accounts must be initiated by a CONFIRM administrator.

Although sometimes very tightly connected, it should be noted that the manual is divided into chapters by *function*, not by *role*.

1.3 CONFIRM users

1.3.1 The questionnaire programmer

The questionnaire programmer is probably the role which requires the most complete knowledge of the CONFIRM web application. The "programmer's guide" is covered in chapter 4. The programmers are typically responsible for these areas of a survey:

- programming questions and skip logic
- setting up project properties (public survey/limited survey etc)
- extend the library of interview layout templates
- compiling/generating interview files to make the project go "live"

1.3.2 Executives

These users are set up with permissions to view and comment on questionnaires and response data. They may also set up and order reports. They may NOT program questionnaires themselves. A survey can never go live without the help of a questionnaire programmer.

Parts of these "executive" users may be project managers working with a research agency; others will typically be the agency's end clients.

1.3.3 Interview managers

CATI projects require that each data collector company has one or more interview managers who administate a group of interviewers. This includes assigning interviewers with the correct survey and samples etc.

It is the responsibility of the interview manager to

- check on the status of the interview (how many have been completed by whom etc.)
- view average interview times (total or by interviewer)
- update response data after it has been collected through the CATI system (correcting punching errors etc)

For more on information on interview management, see "8 Interview management".

1.3.4 Telephone interviewers

The CATI interviewers have their own way of logging on to CONFIRM. Once recognized by CONFIRM, they are taken to the survey to which they are assigned. CONFIRM automatically displays the name/telephone number of the next respondent to call along with the assigned questionnaire.

1.3.5 The web survey respondents

The web survey participants are presented with an easy to use, standard HTML page for entering data into CONFIRM. The survey itself requires no knowlegde of CONFIRM whatsoever. Participants are directed to a survey through links on other web sites, or through email invitations sent directly to them.

The interview pages are easy-to-use standard HTML forms supporting the HTML 2.0 standard.

1.4 Web applications vs "normal" applications

1.4.1 How the Internet works

When you access a page on the web, i.e. http://www.firm.no/pub/index.html, a *request* is sent from your client (the web browser) to the web server where the actual page is located. The web server responds by returning the requested page. This is called a *response*. Once the web server has returned this response, the connection between the client and the server is lost.

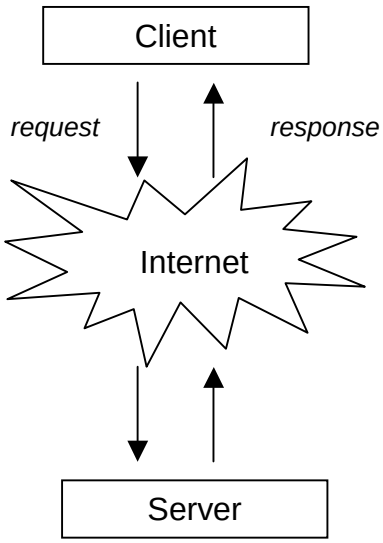

Figure 1: Client/server interaction

This means that the server does not know if the client is still alive/available at other times than when request/response information goes back and forth.

1.4.2 About login sessions

Whenever a CONFIRM user logs on to the application, a new *session* object is created on the server side. This object holds information about the user's *state* (current project etc). The web server allocates memory for this state infomation. By maintaining this information as the user moves from page to page, the individual requests are kept small and simple.

However, the web server does not want to keep track of this information when the CONFIRM user exits the web application. A user may exit the application without explicitly sending a "log off" command to the web server. If the time between user requests exceeds 15 minutes, an automatic *time out* occurs. CONFIRM assumes the user has logged off and frees the memory that holds information about the state. If a new request is sent after a time out, the user is redirected to the first page of the web application. State information like *current project* is lost.

1.5 Security issues

1.5.1 Secure Sockets Layer (SSL)

Regardless of what many people say, the Internet is a secure communications medium. Network traffic travel from your organization's local network, through a network of lines leased from public telephone network providers, before reaching another locally controlled network at the other end of the communication stream. Using the Internet for communication is just as safe as placing a telephone call. As long as you trust the server on the other end of connection, very few things can go wrong; the probability of someone listening in on your traffic is very small indeed.

FIRM offers you the possibility to communicate with *CONFIRM* over the Secure Sockets Layer (SSL), an encrypted protocol that further secures all traffic between you and the system.

1.5.2 Object security

While the previous section deals with Internet security in general, this section deals with security *inside* the CONFIRM application.

A project, a question, a report– these are all *objects* inside CONFIRM's powerful object oriented database. For each object, specific permissions can be set up, including *read* and *write* permissions.

When you initiate a new project (object) inside CONFIRM, the project is by default only visible to you, because noone else has read or write permissions. However, you may specify that other users or groups of users should have these permissions.

1.5.3 Respondent security and anonymity

Web surveys can be set up to run as *public surveys*, in which anybody who accidentally tries to access the web interview page may participate.

CONFIRM also supports *limited surveys*, in which a limited number of respondents are able to log on to the survey. In these surveys, a set of respondent background data may be merged with the response data to produce the final data file. It is the responsibility of the project administrator to make sure that this information is not misused in any way. In theory, the merging may take place without the respondent knowing it. The good thing about the merging is that it makes it possible to produce *respondent specific questionnaires*. For instance, a web panelist do not have to enter information about age or gender simply because this information already exist in his or her profile.

1.5.4 Back up of project meta data

By this we mean all the data of a project that is *not* response data. Every night, a back up copy of all this meta data is saved to external media and stored in a safe. The data is stored in a safe place, and may be restored should a system crash occur.

1.5.5 Back up of response data

It is possible to delete response data from the web application. You are always asked twice about doing this ("Are you 100% sure?"), but it *has* happened that a CONFIRM user has accidently deleted the response data of a running project.

It is the responsibility of the CONFIRM user to make sure that this does not happen. However, the MS SQL server 7.0 periodically runs full back up of all response data in the database. In most cases it is possible to restore most of the data from a recent back up. In the case of a system failure, external media back ups of response data also exists.

1.6 System requirements

1.6.1 CONFIRM users

CONFIRM is designed to function correctly in Microsoft Internet Explorer 4.0. However, there may be bugs in local (non-English) versions of Internet Explorer 4.0 that have been resolved in later versions

of this browser, and these bugs may impact the operation of CONFIRM. We therefore recommend our customers to apply the latest Microsoft Internet Explorer patch for their country. As of today, the recommended version is Internet Explorer 4.01 Service Pack 2.

1.6.2 Respondent requirement

All interview files supports the HTML 2.0 standard, which means that older versions of Netscape and MS Internet Explorer are supported.

1.6.3 3rd party downloads

The CONFIRM application does not rely heavily on 3rd party components. Currently two downloadable components are integrated:

Area	**Type**	**Company**	**See also**
Reporting	Chart component	Visual components	http://www.visualcomponents.com
Questionnaire design	Grid component	VideoSoft	http://www.videosoft.com/

2 Supported data collection methodologies

CONFIRM supports a wide range of data collection methodologies. They are all programmed using the same questionnaire programmer, and they all share the same database formats.

2.1 CATI

CATI interviewing is described in more depth in "7 CATI – Computer Aided Telephone Interviewing".

CONFIRM presents the interviewers with a web-based questionnaire that runs under MS Internet Explorer (v4 or higher). Major advantages includes:

- centralized database and quota control
- ease of distribution – different data collectors may work in teams

2.2 Web

Web interviewing is propably where CONFIRM offers the most cost-effective approach. Respondent self-completion of questionnaires is a lot cheaper than having a CATI center dialing up and recording data *on behalf of* the respondent.

Despite some rumors about web data not being valid data: Comparing web surveys to traditional data collection surveys, there is a growing body of evidence that in certain cirumstances you can get the same results.

For more info, see http://www.virtualsurveys.com

2.3 Paper

Although never the main focus of CONFIRM, tradional *paper questionnaires* still have their place in today's world of data collection.

As of yet (June 15, 1999), the paper questionnaire generator is *not* accessible from the CONFIRM application. Information about the paper questionnaire generator will be posted to the CONFIRM news pages.

For a preview, see "APPENDIX A: Paper questionnaire" for an auto-generated quesitonnaire in MS Word format.

2.4 Hybrid projects

CONFIRM also supports hybrid projects.

3 General features

Up to this point, the document has covered mostly *theoretical* aspects of the CONFIRM application. At this point, however, it is time to log on.

3.1 CONFIRM Login

Users log on to CONFIRM from this URL:

http://www.firm.no/private_poet/

This link can also be found at FIRM's home page, http://www.firm.no/.

Accessing this link will make your browser request a user name and a password, see picture. Enter your personal user name and password, and you are logged on. If the browser who tries to log on to CONFIRM does not meet the browser requirements, an error message will display.

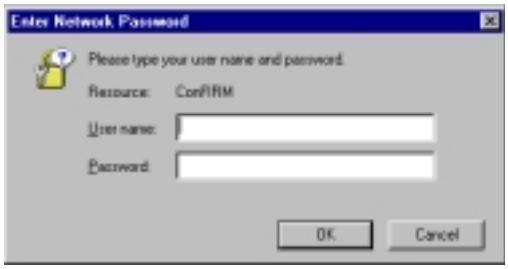

Figure 2: Login box

3.2 Welcome page

This page is also referred to as the *main page*. The first page of the application indicates some natural starting points for your session.

Figure 3: Welcome page

3.3 User settings

Each user has a profile set up at login. The profile holds information about the user's name, email address, preferred language, screen resolution.

The user may change his/her user settings at any time.

Setting	Changable	Description
User name	-	User login name. In combination with a password, this user name will log the user on to the CONFIRM application.
User key	-	Unique to each user. To make yourself “visible” to CONFIRM users outside your organization, you’ll have to supply them with your user key. (This is really a security issue.)
First name, last name	X	Name of user. Used mainly for interaction with other users.
Language	X	Prefered language of user. When viewing or ordering reports, this is language to which the data is translated.
Display	X	Use this for optimizing your screen resolution with the CONFIRM software.
Email	X	Email address of user. This address is used by the CONFIRM message system to send messages about updates etc. to the CONFIRM users. This is also the email address where requested report files are sent.
User level	(X)	Some users are set up with permissions to change their user level.

Table: User settings fields

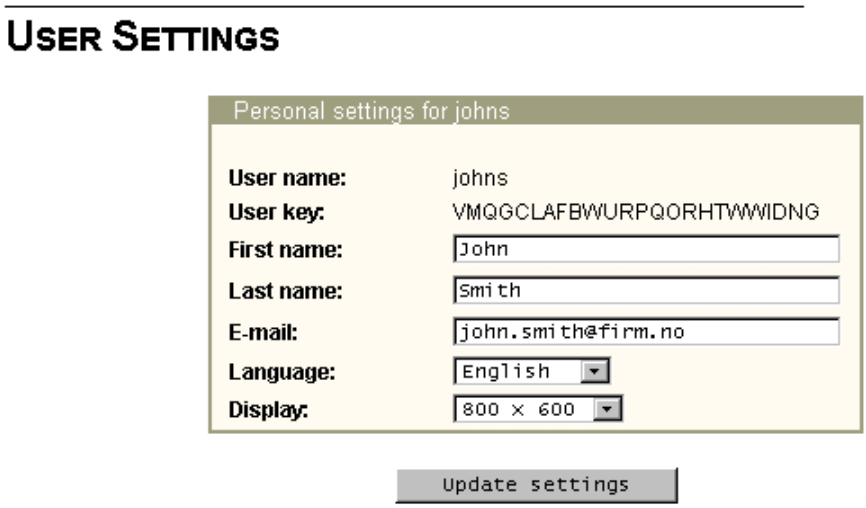

Figure 4: User settings page

3.4 Project list

The project list is the main project organizer. This list contains all projects for which the user has (at least read) permissions. The list is divided into two parts; *Favorite projects* and *Others*. The distinction makes it easier for the user to navigate to the desired project. Projects may be added or removed from the favorite list by clicking the *Add* and *Rem* (= remove) links. “Favorite projects” typically contains the projects that you are currently working on.

The list may be sorted by clicking the header of the column by which you want to sort the projects.

The list may be filtered in a number of ways, by *time* (*created* column) or by *search parameters* (all other columns). A wild card (*) is automatically added to the end of the search criteria.

Example: Specifying “in” in the *name* column will return all projects starting with the letters “in”.

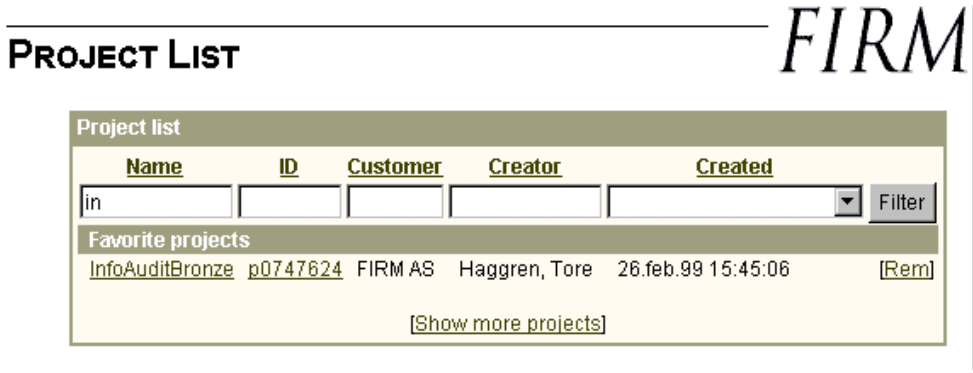

Figure 5: Filtered project list

3.5 Calendar

The calendar is another way of looking at your project list. It is visually broken down along the time axis. The project list may be filtered by customer (the project owner).

3.6 News page

Click *What’s new* to learn about the latest updates to CONFIRM.

3.7 Working in teams

Team working is a useful feature of the CONFIRM application.

For all projects it is true that the user who initiates the project is the project administrator and owner. A project is initially invisible to all other CONFIRM users. Personnel at FIRM are granted access for support purposes.

The project administrator may set up other CONFIRM accounts with permissions to view the new project. From the *Overview* page, click *Object permissions*. In this screen you specify who should have which permissions.

Initially, only users and groups within your organization are listed on the “object permissions” page. However, more users may be added by supplying CONFIRM with correct user keys of these users. Typically, an external user will send you his or her user key by email; this key is then copied and pasted into CONFIRM. (Click the *Add users* button to get to this page.)

3.8 CONFIRM Library

The CONFIRM library makes it easy to reuse previously created questions, modules or even complete questionnaires including skip logic and reporting. The users may only access questions/modules for which they have at least “read” permissions. For more information, see “3.8 CONFIRM Library”.

3.9 Templates

From the “General” menu, a *Templates* link leads to two subsections; MS Powerpoint templates and Web Interview (WI) templates. CONFIRM templates are always bound to a specific organization.

3.9.1 Web Interview templates

Web interview templates define the layout of web interview files. Currently a WI template is defined by

- name and description
- page body (i.e. background image or background color)

- page header and footer
- font attributes
- header/body/footer of individual questions

It is possible to add web interview templates by duplicating and editing existing ones. This requires some HTML knowledge.

3.9.2 MS PowerPoint templates

Predefined MS Powerpoint templates may be uploaded to CONFIRM. All templates uploaded are visible to CONFIRM users within the same organization. A user may select from these templates when ordering MS Powerpoint presentations.

3.10 Initiating a new project

From the main page, a new blank project may be initiated by clicking *Initiate new project*, then *Start from scratch*. This will automatically take you to the *Overview* page for the newly created project.

4 Programming questionnaires

4.1 Creating a new questionnaire

A project's questionnaire is created as a list of consecutive objects. Together they define the routing of the survey. You access the questionnaire by clicking on the menu item *Questionnaire*. An empty questionnaire will look like fig1. Note the three folders, *Routing*, *Scales and lists* and *New objects*.

Figure 6: Questionnaire view

4.2 Adding new objects to the questionnaire

You design your questionnaire by dragging objects from *the New objects* folder, and dropping them in the *Routing* folder. For instance, dragging and dropping a *Single* object from *New objects* into the *Routing* folder, will insert a new single punch form as the first question in the routing. Another way of inserting an object is to select the prefered destination of the object, then select the corresponding object-type from the *Insert* menu.

After you have inserted an object into the routing, it will appear in the routing as a new icon. Double click the object for editing.

Below you will find a description of the different available routing-objects.

4.2.1 Forms

There are currently five standard types of forms available:

Form type	Description
Info	Used for information pages in the survey.
Single	Single-punch question. The respondent can select one answer from a list.
Multi	Multi-punch question. The respondent can select many answers from a list.
Grid	A question type that lets the respondent rate a list along a scale.
Open	An open-ended question.

Table: Standard form types

To create and edit a form, insert it into the routing, then double click it. This will bring up a *details view* of the form on the right hand side of the routing. This view consists of several separate pages, described below. Click the buttons along the top of the frame to access the different pages. Information is saved whenever you move to a new page. Not all pages are available for all form types.

4.2.1.1 Text

On this page, you supply the text of the form, along with other related information.

Field	Description
Title	The title of the form. The title appears in the routing-tree. It also functions as a heading for the form when used in a web-survey, and as a descriptive label for the form when used in online reporting.
Text	The form text.
Instruction	Use this field to give descriptive instructions to interviewers in CATI-interviews. If you work in two or more languages at the same time, there will be a separate set of fields for each active language (see “4.2.1.8 Languages”).
Scratchpad	See “4.4 Scratchpad”.

Table: Text page

4.2.1.2 Answers

[Not available for Info-objects and Open-objects]

Use this page to define the list of answers, or categories, for a question. The main characteristics of an answer is its *text*, which is what the respondent or interviewer will see on the screen, and its *precode*, which is the underlying code stored in the database.

Figure 7: Answers page

The list (see figure) consists of one row for each answer or category, and each row consists of the following columns:

Field	Description
Language fields	The first columns in the list contain the text of the answer. There is one column for each active language (see "4.2.1.8 Languages"). This makes it easy to translate between several languages.
Precode	Defines the precode of the answer. If a precode is not supplied, the precode will be equal to the position in the list, starting from 1. Precodes should only contain alphanumeric characters, with no whitespace. The maximum length is 32 characters.
Colwidth	[Does only apply to grids in web-interviewing] Defines the specific minimum width, in pixels, of a scale-element in grids. Applied primarily to eliminate width-differences between scale-elements, so to create an unbiased scale.
Weight	A numeric weight. Use this field to define the individual weight of each category. You need to define weights to be able to calculate averages for the question in Online reporting.
Punch	[Applies to multipunch questions only] Gives you the ability to define one or more elements in a multi-punch list as single-punch (ex: "None of the above").

Table: Answers page

When the question is newly created, the list of answers will be empty. Press the *Add*-button to create a new answer in the list. Click inside the first column in the list. This will activate the first column. The active column is always marked with a green rectangle. When you start typing, the active column will be put in editing mode, and a cursor will appear. If there is previous text in the column, your new text will be appended to the end.

When in editing mode, you can move to the next field by using the tab-button, or by clicking in the column with the mouse. You can directly move to an adjacent row by using the up- and down-arrow buttons. If you are at the bottom of the list, the down-button will create a new row.

When you are not in editing mode (no cursor), you may use the left- and right-buttons to move horizontally around in the list. Press <F2> to enter editing mode.

You can copy rows from the list by pressing <CTRL-C>, and pasting rows by pressing <CTRL-V>. The checkbox *Only copy active button* decides whether you copy and entire row or not, and the checkbox *Overwrite* decides whether you insert new rows when you paste or not.

The button *Add predefined* allows you to add a predefined list or scale to the answer list (see "4.5 Lists"). This is very powerful when reusing for instance brand lists or scales in the survey. In a predefined row, the first column will become a dropdown-list when activated, giving you the opportunity to choose between the different predefined scales and lists.

The scratchpad is explained in "4.4 Scratchpad".

[The list-implementation will be replaced with a spreadsheet-like component in the near future, so the information in this chapter is subject to change quite soon.]

4.2.1.3 Scale

[Only available for grids]

This page defines the *scale* dimension of a grid. The page is functionnally equal to the *Answers* page.

4.2.1.4 Properties

This page lets you define several aspects of the form-object:

Property	Description
Question ID	The project-unique ID of the form. An alphanumeric string with no whitespace, starting with a letter. Constrain length of ID to 8 characters if you want to use the ID as column-name in SPSS when exporting data to this format. For multi-questions and grids, constrain length to 5 or 6 characters, because column names are constructed like this: <ID>_<precode>
Precode mask	If you want to dynamically exclude one or more of the answers in the answer list, use this field to create a JScript-expression resulting in a set of precodes to include. See Section 4.8 for a description of expressions.

Table: Form properties

In the following list of flags, not all flags are applicable to all form types, and none of them are available to the Info-form.

Flag	Description
Randomize	[Available in Single, Multi, Grid] When checked, the list of answers will be randomized [Only available in WI].
Recoded var	[Single, Multi, Open] Turns this form into a recoded variable, which will not show up in the interview. You can predefine complex recoding rules in SQL or Java (see "9.9 Recoding data").
Background var	[Single, Multi, Open] Treat this form as a background variable. The form will not show up in the interview, but will fetch its value from the respondent list. Very useful when you have information on your respondents, which you want to apply in the skipping-logic of your questionnaire, or in the Online reporter. A column with a name equal to the form-ID is required in the respondent list.
Hidden	[Single, Multi, Open] A hidden form is hidden from the interviewer/respondent, but its value may be set on the fly from script-objects, making "on-the-fly"-recoding possible.
Ordered	[Multi] An ordered multi-form will record the order of the respondent's answers [CATI only].
OpenText	[Multi] This flag will turn a multi into a list of text-boxes, making it easy to create open ended ratings [WI only].
Quota var	[Single] Checking this flag will include the variable in the quota-calculations. You define the quotas themselves elsewhere (see "7.2 Quotas ") [CATI only].
Dropdown	[Single, Grid] Singles and grids are normally represented as rows and columns of radioboxes. This flag will generate dropdown-boxes instead [WI only].
Not required	[Single, Grid] Normally, all single-punch questions requires an answer. Checking this flag will turn off this validation [WI only].
Deleted	[All] When you delete an object from the routing, it is not permanently deleted, but the "deleted"-flag is set. If you turn on *Show deleted nodes* (see "4.3 Editing the routing"), you can undelete the object by unchecking this flag.
Validation code	For unlimited flexibility in question-validation, you can write JavaScript-code to validate your answers. See Section 4.9.

Table: Form flags

4.2.1.5 Preview

The *preview* page will give you an idea of the respondent's view of the question. The question is displayed using a standard web-interview template.

4.2.1.6 Permissions

[To be completed]

4.2.1.7 Tracking

[To be completed]

4.2.1.8 Languages

In a multilingual project, you may not want to work with all project-languages at the same time. This page lets you select a subset of languages to work in. Check or uncheck languages, and select another page.

4.2.2 Conditions

Using condition-objects, you can create *skip-logic* in your questionnaire. To create and edit a condition, insert it into the routing, and double click on it. You insert conditions wherever you want to alter the flow of a survey based on a boolean expression, for instance based on the answer of a previous question. If the expression evaluates to *true*, the survey-mechanism will execute the routing-objects beneath the *THEN*-node. If the expression does not evaluate to *true*, the objects beneath the *ELSE*-node will be executed, if the *ELSE*-node exists.

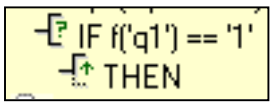

Figure 8: Conditional node

The expressions are written using the JScript-language. Expressions are described in Section 4.8 and to help building expressions, there is a c*ondition builder* (see "4.2.2.2 Condition builder").

4.2.2.1 Description of fields

Field	**Description**
Expression	A JavaScript-expression evaluating to *true* or *false*. Described in detail below (see"4.8 Expressions Involving Survey Variables ")
Branches	Select whether the condition should include an *ELSE*-branch or not.
Deleted	Used the same way as in forms (see "4.2.1.4 Properties")

Table: Conditional fields

4.2.2.2 Condition builder

The condition builder helps you to build JavaScript expressions in no time.

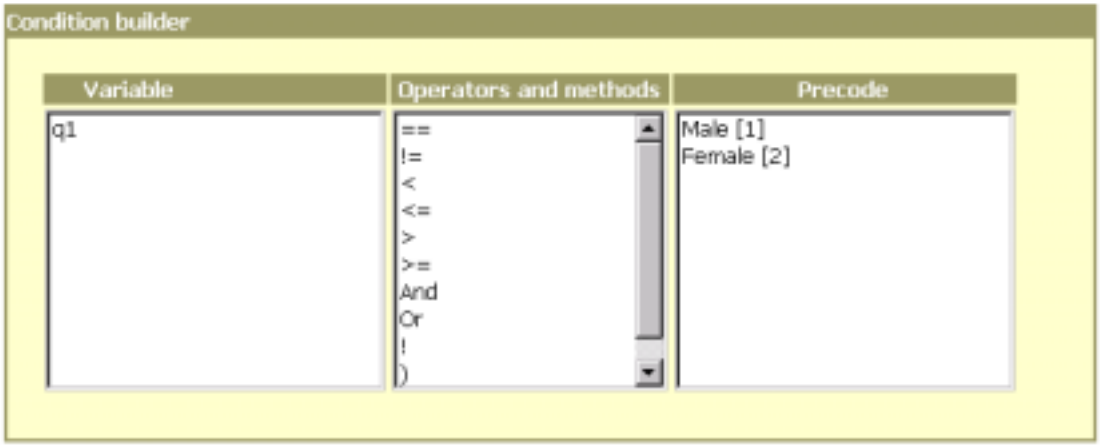

Figure 9: Condition builder

To use the builder, activate a condition-object by double clicking on it. **Single**-click on a form in the routing-tree you want to base your expression on (Double-clicking will activate form-edit mode). This will bring up a builder for the chosen form. The builder consists of three lists:

List	Description
Variable	This list will show the form-ID of the selected form. If the form is a grid, each of the answers in the grid (which are themselves single-punch questions) will be shown. Click on an item in the list to insert a reference to the form in the expression-field.
Operators and methods	This list includes most of the operators and methods you can apply to the form-object in focus. The list will be different for each type of form, because the various form-types support different methods. Click on an item in the list to insert the operator or method into the expression-field.
Precode	This list will show the answers of the selected form, with the precode in square brackets. Click on an item in the list to insert the precode into the expression-field.

Table: Condition builder

In combination, these lists make it easy to build complex expressions without remembering the JavaScript-syntax. You may at any time single-click another form in the routing to expand your expression to be based on more than one form.

4.2.3 Directives

Directives are instructions to the interview-generator that apply from where they are inserted in the routing and forward.

At the moment, there exist two directives, and they are only valid in web-interviews:

Directive	Description
MULTIPLE_QUESTIONS	This directive will instruct the interview-generator to display the succeeding questions as a batch of questions on one page. There will be a break when a condition or a new MULTIPLE_QUESTIONS- or SINGLE_QUESTIONS-directive is encountered. Example: If you want a web-survey to break for each ten forms, insert a MULTIPLE_QUESTIONS after every 10th form.
SINGLE_QUESTIONS	One question per page.

Table: Directives

4.2.4 Stop-objects

A Stop in the questionnaire will end the interview. This is mainly used for screening-purposes, and to end the interview various places in the routing. The status-code you set on the Stop-object is important:

Setting	Description
Complete	The interview should be considered a complete interview. Interview-counts are incremented, and search+interview timings are recorded. Normally, online reporting will only show statistics for complete interviews.
Screened	Use this status-code to mark the interview as incomplete because of screening. Interview-counts are not incremented, and the search-timer is not reset, meaning that interview-time will be part of the search-time for the next complete interview.
Quota full	[Do not use]

Table: Stop node settings

4.2.5 Scripts

For very advanced routing, on-the-fly recoding, or other programming efforts, you can insert script-objects which will run your JavaScript-code when executed. When using script-objects, care must be taken to create codes that will run without errors. JavaScript is an interpreted programming language, which means that the code will only be checked at run-time. You should always perform thorough testing of questionnaires that include script-objects.

4.2.6 Folders

To organize your questionnaire, create folders in the routing. For example, put all demographic-forms into one folder.

Field	Description
Name	The name of the folder (shows up in the routing-tree).
Description	A description of the folder.

Table: Folder fields

Folders make it easy to move a set of related routing-objects around in the questionnaire. Folders are also used in conjunction with the library, and in the creation of wizards.

4.2.7 Loops

[CATI only. Further development during late 1999] A loop is a special type of folder, where the routing inside the loop will be repeated for a number of iterations. Let's say you want to repeat a block of questions for a list of car-brands. Instead of defining the same block of questions for each car-brand, only replacing the brand name, you can simply create a loop iterating over the list of cars, asking the same questions for each brand.

There may be two or more loops in the questionnaire, but not nested loops.

The initial definition of a loop consists of giving the loop a unique ID within the project, and creating a list of loop-iterations (created in much the same way as you create an answerlist in a form (see "4.2.1.2 Answers"). The different parts of the loop-details view are:

- **Loop name:** A unique ID of the loop within the project.
- **Loop members:** A list of loop-iterations, edited the same way as the answer-list in forms. Definition of standard columns:

Column	**Description**
Language columns	The first columns in the list contain the text of the loop-iteration. There is one column for each active language (see "4.2.1.8 Languages").
Precode	Defines the precode of the iteration. If a precode is not supplied, the precode will be equal to the position in the list, starting from 1. Precodes should only contain alphanumeric characters, with no whitespace. The maximum length is 32 characters.
Active	Defines whether the iteration is active or not. For instance, in long-running trackers, you may want to ask the same set of questions once a week, but for different brands/concepts. By adding to the list, and activating/deactivating iterations, you can keep the same questionnaire.

Table: Standard loop columns

An important feature of the loop, is its ability to store background-information for each loop-iteration. For instance, in product or concept testing, you can store important categorized characteristics (color, size, etc.) for each iteration. You do this by first creating one hidden single-form with the necessary categories (red, blue, green) per characteristic inside the loop. After that, you expand the number of columns in the loop-member list by entering the ID of each of these forms into the input-box next to the *Add Element* button in the loop-details view, and pressing *"Add Element" (one ID at a time)*. The list will expand to become a grid where you can specify the different characteristics of each iteration by entering the values for the hidden variables.

One form is required inside the loop: A hidden single-form with the ID *"<loopname>id"*, and with the same precodes as the loop itself. [Subject to change quite soon].

4.3 Editing the routing

You reorganize the routing mainly by means of drag & drop in the routing-tree.

4.3.1 Moving an object

To move an object (and its children, when moving folders, loops and conditions), left click the *text* of a routing object in the tree. This will make the object turn **blue**.

- Keep pressing the button, and start dragging. After moving the mouse a few pixels, a yellow drag-object will appear to the right of the mouse-pointer, giving you an indication of what you are dragging.
- At this point, releasing the mouse-button will drop your object on the object that is beneath the mouse-pointer. This destination object is always **red**.

- If you want to cancel the drop-operation while dragging, press the <ESC>-button. Do not just drop the object “out of sight”.
- If the destination object is out of view because the routing is long, use the arrow-keys or PgUp/PgDown-keys to scroll up and down slow or fast. Keep the mouse-button pressed.
- When you drop the object, it will be moved to the new location, as long it is dropped on a legal destination. An example of an illegal destination, is one of the objects in the *“New objects”*-folder.
- Dropping an object on top of another, usually means to move the object to the position *behind* the destination object. If you drop an object on a folder-type object, it will end up *behind* the folder, not *inside*. If you want the object to end up *inside* the folder, hold the <ALT>-key down before dropping. The object will end up as the first object in the folder.

4.3.2 Duplicating an object

Duplicating behaves exactly like moving, but you create a copy of the source object. To duplicate, hold the <CTRL>-key down before dropping. An alternative is to select an object, and choosing *Duplicate* from the *Item* menu.

4.3.3 Selecting multiple objects

If you want to move or duplicate multiple objects, select the first object, hold <SHIFT> or <CTRL>, and press another object. <SHIFT> will multi-select consecutive objects, and <CTRL> will multi-select specific objects. To drag the selected objects, hold <SHIFT>, click the last selected object, and start dragging. You may release the <SHIFT>-button while dragging, and use the <ALT> and <CTRL> button to move inside or duplicate as described in “4.3.1 Moving an object”.

4.3.4 Deleting objects

To delete one or more objects, select them, and press <DEL>. This will remove the selected objects from the routing, and set the *Deleted* flag on the objects to true.

You can undelete such objects by checking *Show deleted nodes* from the details-view of the top folder in the routing-tree (The root-folder with the name of the project. Double click to access details-view), and refreshing the tree. All deleted objects will appear as grayed-out in the routing-tree. Access the *Properties* page for the deleted object you want to recover (see “4.2.1.4 Properties” for forms), and uncheck the *Deleted* flag.

However, if the objects are already marked as deleted, and you delete the deleted objects, they are **permanently** deleted, and non-recoverable.

4.3.5 Pitfalls

Due to internet-browser limitations, some of the drag & drop operations may not be as intuitive as they are in normal Windows-based application:

- When picking up an object to drag, point to the **text** of a routing object, not the **icon**. You cannot drag the icon.
- If you drag too many objects, the drag-operation will fail with an error-message.
- After dropping an object, there will be a delay because of the round-trip to the server.

These problems will be solved when the system starts taking advantage of functionality specific to Internet Explorer 5.0.

4.4 Scratchpad

The scratchpad is a text editor available in several screens. If you paste text into the scratchpad, for instance from a Word-file, you can easily copy and paste text into textboxes, grids, etc. The content of the scratchpad is stored locally in the web-browser, and is available as long as the web-browser is open, or until the session times out.

Hint: Pasting a Microsoft Word-table into the scratchpad will convert the Word-table into a tab-delimited list. This list can be copied directly into an answer list or a loop-list, if the columns are correct.

4.5 Lists

Quite often, one or more scales, or one or more brand-lists, product-lists, etc. are reused several times in a questionnaire. CONFIRM lets you predefine such lists, and reuse them in forms (see "4.2.1.2 Answers"), or even in other lists (nested lists). Whenever you change the contents of a list, this change will be reflected in all lists that use the changed list.

To create a list, either drag a *Scale /List* object from the *New Objects* folder to the *Scales and lists* folder, or select *Scale /List* from the *Insert* menu. This will create a new list in the *Scales and lists* folder. Double-click the list to edit.

Field	Description
Name	A descriptive name for the list. The list-name is the text that shows up in the drop-down list when inserting the predefined-list in another form or list.
Answers	The answer-list behaves like the list in forms and loops. See "4.2.1.2 Answers" for description of usage.
Scratchpad	See "4.4 Scratchpad"

Table: List fields

4.6 Adding comments

[To be completed]

4.7 Library

[To be completed]

4.8 Expressions Involving Survey Variables

All non-trivial questionnaires contain expressions. Expressions involving survey variables are used:

- In form elements for masking, validation, and text substitution.
- For controlling the flow through the questionnaire.
- In general-purpose code contained in script nodes.

CONFIRM uses Microsoft's JScript[1] scripting engine to evaluate all questionnaire expressions and to execute scripts. The runtime environment of the interview engine (or simply *the runtime*) supplies a number of functions and objects that provide references to and lets you manipulate survey variables.

The first part of this section presents a description of expressions that is rather terse and meant as a reference guide. Examples of how to put expressions to work are given towards the end.

4.8.1 Variable Types

CONFIRM currently provides support for three types of variables:

- *Coded variables* are assigned distinct values from a pre-coded domain of alternatives. The runtime treats the value of a coded variable as a JScript string. In *CONFIRM* users enter the

[1] JScript is Microsoft's adaptation of ECMAScript, a computer language standard based on several originating implementations, most notably Netscape Communication's JavaScript and JScript itself.

responses to coded variables by use of groups of radio buttons or dropdown lists. **Single** forms declare one and **grid** forms a set of coded variables.

- *Dichotomies* may be assigned the integer values 1 or 0, that denote a positive or a negative response respectively. *CONFIRM* uses checkboxes for input of dichotomies. These variables are declared in **multi** forms where they typically appear in groups.
- *Open variables* have an unbounded value domain. In expressions, they are treated as strings. You declare open variables in **open** forms or by checking the OpenText property for a **multi** form.

The runtime defines individual variable objects for each variable type and a *Set* object, which is used to manipulate groups of variables or to describe value domains.

4.8.2 Variable Objects

In the interview system, JScript objects represents each variable. When you reference these objects in expressions, these type conversions apply:

- In Boolean context they are always true.
- In string context, open and coded variables are converted to variable string value if their values have been set. Dichotomies evaluate to their precode.
- In numeric context, an attempt will be made to convert the string value of open and coded variables to a number. Dichotomies evaluate to 0, 1, or NaN (not a number) if their values have not been set.

Each variable object provides two methods for assignment and reference of its value[2]. In the following table `x` denotes the variable object and `v` the result of a JScript expression convertible to a value in `x`'s value domain.

Expression	Result	Semantics/Note
`x.get()`	value of `x`	For open and coded variables, this is a string. For dichotomies an integer value (1 or 0) is returned.
`x.set(v)`	new value of `x`	`x` is equal to the result of converting v to a value in `x`'s value domain

Table: Variable objects expressions

4.8.3 Set Objects

Instances of set objects are primarily used in expressions that mask alternatives of a coded variable or prevent answers to certain variables in a multi-response form. They are also applied in branch expressions, especially for referring to individual member values in groups of dichotomies.

A set object is not a real set in the mathematical sense of the word. They are really maps of objects whose property names can be thought of as the set's members. Contrary to mathematical sets these objects also have an associated value. Most often, you need not pay attention to the distinction. In some contexts you will treat a set as you would have a mathematical one and apply its set operators to obtain set results. In other cases you use the set as an object map in order to access each member values.

When you reference set objects in expressions, these type conversions apply:

- In Boolean context they are always true.

[2] JScript does not allow operators to be overloaded. Because of this, common operations such as assignment are provided in a functional notation with object members rather than by use of the infix operators defined for JScript's primitive types.

- In string context, the string values of all a set's members that evaluate to true are concatenated and separated by commas.
- In numeric context, set objects evaluate to NaN.

The operation supported by sets are summarized in the following table, where `s` denotes a set variable and `v` is a JScript expression:

Expression	Result	Semantics/Note
`s.add(v)`	new value of s	Adds v to the set.
`s.remove(v)`	new value of s	Removes v from the set.
`s.union(v)`	new set which is the union of all members of `s` and `v` that evaluate to `true`.	If `v` is not already of set type, then `v` is converted into a set before the operation is performed.
`s.isect(v)`	new set which is the intersection of `s` and `v` that evaluate to `true`.	
`s.diff(v)`	new set which is the difference between `s` and `v` that evaluate to `true`.	
`s.inc(v)`	`true` if `v` is a member of `s`, otherwise `false`.	This member is used to test for set inclusion. `v` is converted to a string.
`s.size()`	integer value, the size of `s`.	
`s[v]`	set member v	`v` is converted to a string.

Table: Set object expressions

The instances of set that you will use are most often those returned by functions defined by the runtime. Sometimes, however, you will need to create set instances yourself. The `set` function is provided to accomplish this.

`set` function

Description

Creates a new set object.

Declaration

`set([`*mem*$_1$`], …`*mem*$_n$`)`

Remarks

`set` returns a set instance. If provided, the set is populated with the arguments *mem*$_1$ to *mem*$_n$.

4.8.4 Runtime Functions that Access Variables

The runtime provides two functions `f` and `a` that return information about forms and form variable values:

`f` function

Description

Accesses survey variable values.

Declaration

`f(`*formid*`)`

Remarks

You may think of `f` as a mnemonic for *form*, which is place in the questionnaire definition where all variables are declared.

formid is the ID of the form, as specified in the form's properties, where the variable(s) in question are declared. The function returns different types of objects depending on the form type:

single: coded variable with a value domain equal to the "precodes" in the answer list of the form.

open: open variable

multi: set of dichotomy variables with member names corresponding to the "precodes" in the answer list of the form.

multi with OpenText checked**:** set of open variables with member names corresponding to the "precodes" in the answer list of the form.

grid: set of coded variables, each with a value domain equal to the "precodes" in the form's scale list, and with member names corresponding to the "precodes" in its answer list.

`a` function

Description

Returns the complete set of "precodes" defined in a form's answer list.

Declaration

`a(`*formid*`)`

Remarks

`a` returns the complete set of "precodes" defined in the answer list belonging to the form identified by *formid*.

The semantics vary depending on the form type:

single: the returned set corresponds to the value domain of the associated variable.

multi and **grid:** the returned set corresponds to the individual variables' subscripts in the form.

4.8.5 JScript Logical Expressions: A Five Minute Tour

CONFIRM's condition nodes include expressions that evaluate to a Boolean value. The result is used to decide whether a single block (If..Then branch) or one of two blocks (If..Then..Else branches) of questionnaire elements is executed. The Boolean result is either `true` or `false` and depends on which *operators* you use in your expression and their *operands*.

As mentioned, *CONFIRM* relies on the JScript scripting engine when it evaluates expressions. You do not need intimate knowledge of JScript to use the system, but will find a few basics useful. The JScript

expression syntax and semantics has largely been adapted from the C/C++/Java family of languages. Although the language has a full range of operators, only those concerned with evaluation of logical expressions will be treated here. If you crave for more knowledge, Microsoft provides information on JScript at http://msdn.microsoft.com/scripting. The ECMAScript standard itself (which like all standard documents is rather daunting) is available at http://www.ecma.ch/stand/ECMA-262.htm.

A summary of useful operators is given in the table below:

Operator	**Description**
`!`	unary operator, logical NOT
	Relational operators:
`<`	less than
`>`	greater than
`<=`	less or equal
`>=`	greater or equal
	Equality operators:
`==`	equals
`!=`	does not equal
`&&`	logical AND
`||`	logical OR

Table: Useful operators

All operators bind from left to right and the operator precedence decreases from top to bottom. This means that the operators listed first bind stronger to their operands than those below. Sub-expressions can be grouped with parentheses to change the evaluation order.

Below you will find a few examples that illustrate the use of operators and precedence rules. Assume that `f('q1')` returns the string value `'1'` and `f('q2')` the value `'3'`.

Example 1

Expression	*Result*
`f('q1')=='1' && f('q2')=='3'`	`true`

The equality expressions are evaluated first and both are `true`. Then the logical AND is applied to produce a `true` result since both operands are `true`.

Example 2

Expression	*Result*
`f('q1')=='2' || f('q1')=='1' && f('q2')=='3'`	`true`

First the expression `f('q1')==2` is evaluated with a `false` result. Then the sub-expression from example 1 is evaluated because `&&` has higher precedence than `||` and this expression is `true`. Finally, the logical OR operator is applied and yields a `true` result because its second operand is `true`.

Example 3

Expression	*Result*
`(f('q1')=='2' || f('q1')=='1') && f('q2')=='3'`	`true`

In this example the sub-expression `f('q1')=='2' || f('q1')=='1'` is enclosed in parentheses and will be evaluated first. Its result is `true` because `f('q1')=='1'` is `true`. In turn, `f('q2')=='3'` evaluates to `true` and the entire expression becomes `true` when the logical AND is applied.

Example 4

Expression	*Result*
`!(f('q1')=='2' || f('q1')=='1') && f('q2')=='3'`	`false`

As was the case in Example 3, `f('q1')=='2' || f('q1')=='1'` is evaluated first because of the parentheses. Then the unary logical NOT is applied to this expression resulting in a `false` value. Because one of the logical AND's operands are `false`, the entire expression becomes `false`.

4.8.6 Applying Expressions

This section contains an imaginary case that illustrates practical use of expressions in questionnaires:

Mr. Smith wants to elicit people's awareness of Brands A to F within a product category X. First he wishes to ask a "top of mind" question to get some idea of what the most powerful brand name is. If the interviewee is unaware of any brands in the product category, then she is screened and the interview ends. Otherwise, our imaginary researcher would like to find out what other brand names she knows of. In cases where the interview subject knows of at least two of the top three brands A, B, or C, he also wants to acquire knowledge about which brands the respondent buys, and when she last purchased each brand.

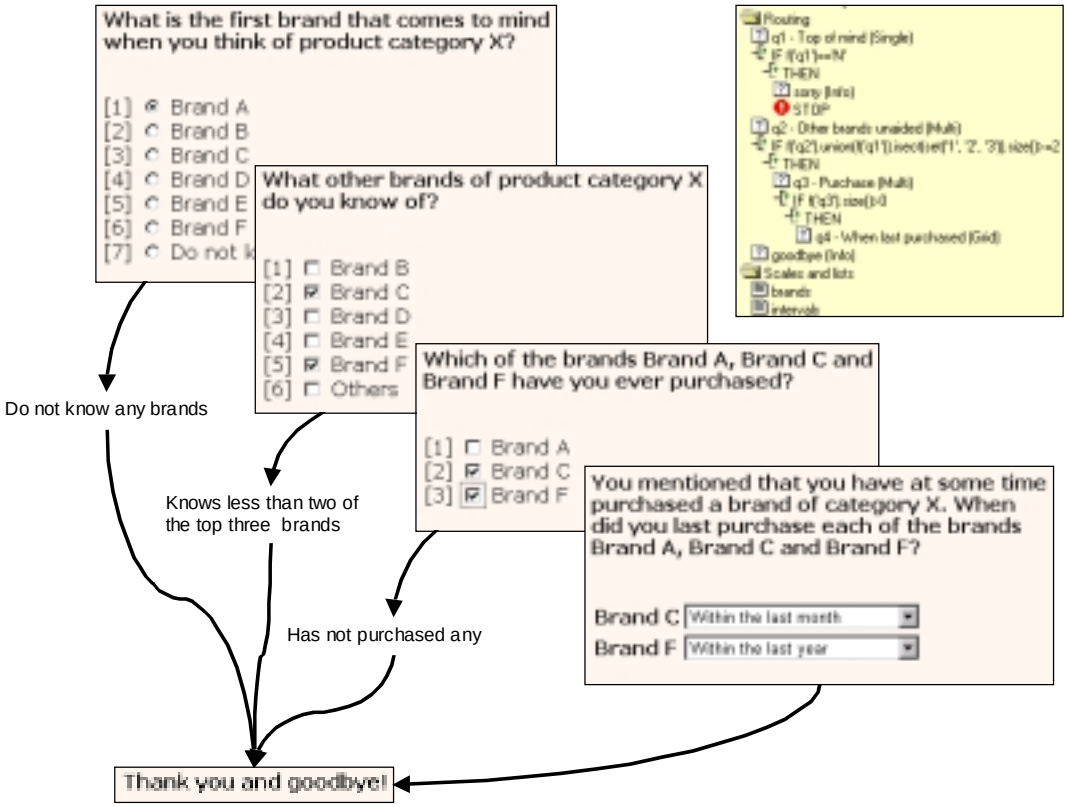

Figure 10: Applying expressions example

The figure above shows the routing and illustrates the forms used to enter responses:

The first form, q1, displays a complete list of brands and a "Do not know any" alternative. If the last alternative is checked then the interview proceeds to the sorry **info** form with the "Thank you and goodbye" message. In other cases the interview moves on to q2.

q2 is a **multi** form on which the remaining brands are displayed. In this example the interviewee knows of Brand C and F in addition to Brand A checked in the first form. Because the user knows two of the top three brands (A and C), the branch leading to q3 is followed.

In q3 the answers of q1 and q2 are combined to produce a list that contains all brands mentioned by the respondent. She is subsequently asked whether she has ever purchased any of these. If one or more of the inputs are checked, then the interview continues with q4.

q4 contains inputs that acquire the last information wanted. A dropdown list is displayed beside the two brand names known to have been purchased by the respondent due to the answer to q3. After q4 is completed the interview falls through to the last **info** form – goodbye – before the it ends.

4.8.6.1 The Branch Expressions

This questionnaire contains three condition elements that evaluate a Boolean expression to determine the sequence of forms shown. The list brands declares the set of brands A to F. The corresponding code for each brand is "1" to "6" respectively.

- The first condition node is placed directly after q1, where the "Do not know any" alternative, pre-coded to "N", is placed below the brand list. If this alternative is checked then the interview program follows the condition's THEN branch leading to the sorry **info** form before the interview eventually ends. Its expression simply test whether the response submitted is equal to "N":

```
f('q1')=='N'
```

- Condition number two is placed after q2 and contains the most complex expression found in this questionnaire:

  ```
  f('q2').union(f('q1')).isect(set('1', '2', '3')).size()>=2
  ```

  The purpose of this rather intimidating expression is to check whether the combined answers to q1 and q2 contain at least two of the brands A, B, or C. Going left to right, we start with the function call `f('q2')` which returns the set of positive answers given in q2. This set is united with the answer to q1 before the intersection between the result of this operation and the "hand-crafted" set with members `'1'`, `'2'`, and `'3'` is computed. At this point, we have found the answers in q1 and q2 that correspond to the top three brands. Finally, we compare the size of this set to two, to check if we have as many answers that identify the top three brands as we required.

- Finally, the questionnaire contains a condition after q3. The condition

  ```
  f('q3').size()>0
  ```

  simply checks whether any of the check boxes in this form where selected.

4.8.6.2 Masking Alternatives

Because of logical relationships between variables in different forms we use masks in order to make the interview easier to fill in and to prevent erroneous input.

- In q2 the answer given in q1 is removed from the list of brands displayed, as we are only interested in brands other than the one at the interviewee's top of mind. The expression that achieves this is

  ```
  a('q2').diff(f('q1'))
  ```

  which computes the difference between all of q2's alternatives and the answer given to q1.

- q3 displays a list which combines the answers to q1 and q2, which is the set of all the brands that the respondent was aware of. In this case the mask expression is

  ```
  f('q2').union(f('q1'))
  ```

  which you may recognize from the condition placed after q2 above.

- q4 displays a list corresponding to the brands checked in q3. Its mask is simply

  ```
  f('q3')
  ```

4.9 Form Validation

When you add a form to a questionnaire you actually add three separate operations to the interview program:

1. **User interaction:** Display one or more forms to the person entering responses and wait for input.
2. **Store:** Store the value of entered responses.
3. **Validate:** Check that answers stored comply with the specified data validation rules. If an error is raised then repeat from 1.

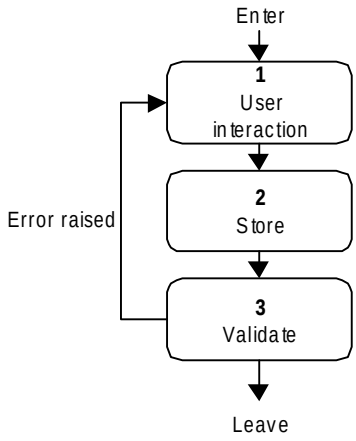

Figure 11: Form validation

This section describes the default validation rules and instructs you on how you can add your own code to a form in order to perform custom validation operations.

4.9.1 Default Validation Rules

The default validation rules in general only check that all questions have been answered:

- At least one alternative must be checked/selected for a coded variable defined in a **single** or **grid** form.
- Textboxes used for input of open responses cannot be left blank.

Multi forms are given special treatment. For these forms, a negative response to all questions (none of the alternatives checked) actually constitutes a valid answer. However, if one or more of the members in the form's answer list has the punch type "single" then a different method of validation is applied. A "single punch" element in a **multi** form declares a variable that is exclusive within the group of variables declared in the form. A positive response to an exclusive value eliminates the possibility that any other variables in its group can have a positive value, e.g. a "None of the above" alternative. In this case, the variables are assumed to exhaust all possible answer combinations and at least one positive answer is required. In consequence, the response will not be deemed valid unless the interviewee gives a positive response to at least one of the none-exclusive or to only one of the exclusive variables.

4.9.2 Adding Your Own Validation Code

One of the form properties is a text box where you may enter arbitrary script code to validate answers. Depending on the complexity of the validation problem, crafting this code may require a more intimate knowledge of JScript than can be presented in this document. This presentation of validation code is limited to a description of the interface provided by the runtime for signaling errors and setting error messages. It also provides a few examples.

4.9.2.1 Runtime Functions for Error Handling

The *CONFIRM* runtime provides two functions for use in validation scripts: `RaiseError` and `SetErrorMessage`. Of course, the full range of functions and objects used to access and modify response data is also available in the validation scripts.

`RaiseError` function

Description

This function is invoked to signal an error.

Declaration

```
RaiseError()
```

Remarks

`RaiseError` signals to the interview runtime that the input entered in a form is invalid and that the form should be redisplayed with an error message.

`SetErrorMessage` function

Description

This function sets the error message for a specified language.

Declaration

`SetErrorMessage(`*langID*`, `*message*`)`

Remarks

Call this function to set the error message (the system default, "Please answer all questions," does not apply in most cases when custom answer validation is used). The function takes two parameters:

- *langID*: A language ID. This argument is provided to support multi-lingual studies. You should provide one message for each of the languages used for interviewing. The runtime provides an object `LangIDs` whose members map language codes to the internal values used to identify each language available in *CONFIRM*. The member names equal the codes specified in the international standard ISO 639 "Code for the representation of names of languages". For example, `LangIDs.en` represents English; `LangIDs.fr`, French; `LangIDs.de`, German; and so on.
- *message*: This argument is simply a string value with whatever message you wish to display.

4.9.2.2 Examples of Custom Validation

In conclusion we will take a look at two examples of script code for validating form input.

Example 1

In this example `q1` is a **multi** form and it is required that at exactly three of its alternatives should be checked.

```
if (f('q1').size()!=3)
{
    RaiseError()
    SetErrorMessage(LangIDs.en,
        "Check exactly three alternatives!")
}
```

`f('q1')` returns the set of variables in `q1` with positive answers and we check if the size of this set is equal to three. If the size differs, we signal an error and set an error message telling the respondent to amend his ways.

Example 2

In this example, q2 is an **open** form whose input should be numeric and in the range 1000 to 9999.

```
var n = parseInt(f('q2'), 10)
if (isNaN(n) || n<1000 || n>9999)
{
    RaiseError()
    SetErrorMessage(LangIDs.en,
        "Enter a number in the range 1000 to 9999")
}
```

`f('q2')` returns a string value containing the answer to q2. The built-in JScript function `parseInt()` is used to convert this value to an integer with radix ten which is stored in the variable n. Another JScript function, `IsNaN()`, is used to check whether it was possible to convert the answer to a number, before we test whether the value lies within the valid range. If we are not satisfied with the answer we raise an error and whine.

5 Preparing for Data Collection

5.1 Generating response databases

When you are done with a questionnaire, or you want to test the questionnaire, you must prepare a response-database for your survey. Every time you make changes in the questionnaire, you have to update the response-database. This process is called compilation. The system keeps two separate databases for your survey, one test-database, and one production-database. You generate the test-database from *Test and verify → Compile test env.*. The production-database is generated from *Administrate project → Compile production env.*.

When clicking *"Compile ..."* for the first time, you will be presented with a dialog, asking you to confirm the creation of the new database. If you click *Yes*, a new database will be created and compiled. This may take some time, depending on the length of your questionnaire.

During compilation, the system performs some checks to ensure the integrity of the questionnaire. For instance, if you have non-unique form ID's in the questionnaire, the system will stop the compilation with an error message.

If the database already exists because of a previous compilation, you are presented with a slightly different dialog. You can choose to create a new database, or update the existing. Most of the time, you should only update the database. This will preserve all data (responses, respondents, quotas, etc.) in the database, and is quicker. In a test-phase, however, it may be useful to start from scratch by recreating the database. **WARNING: This will erase all data in the database. You should NEVER do this in a running production-database.**

5.2 Generating interview files

Before you can start interviewing, interview files have to be generated. WI-files are different from CATI-files. Before generating these files, you have to generate a response database.

5.2.1 Web interview files

The web interview generator produces a file capable of

1. Displaying forms for user input over the web,
2. Storing the respondent's answers, and
3. Controlling questionnaire flow based on background data (if available) and answers received.

A description of the available options is provided below.

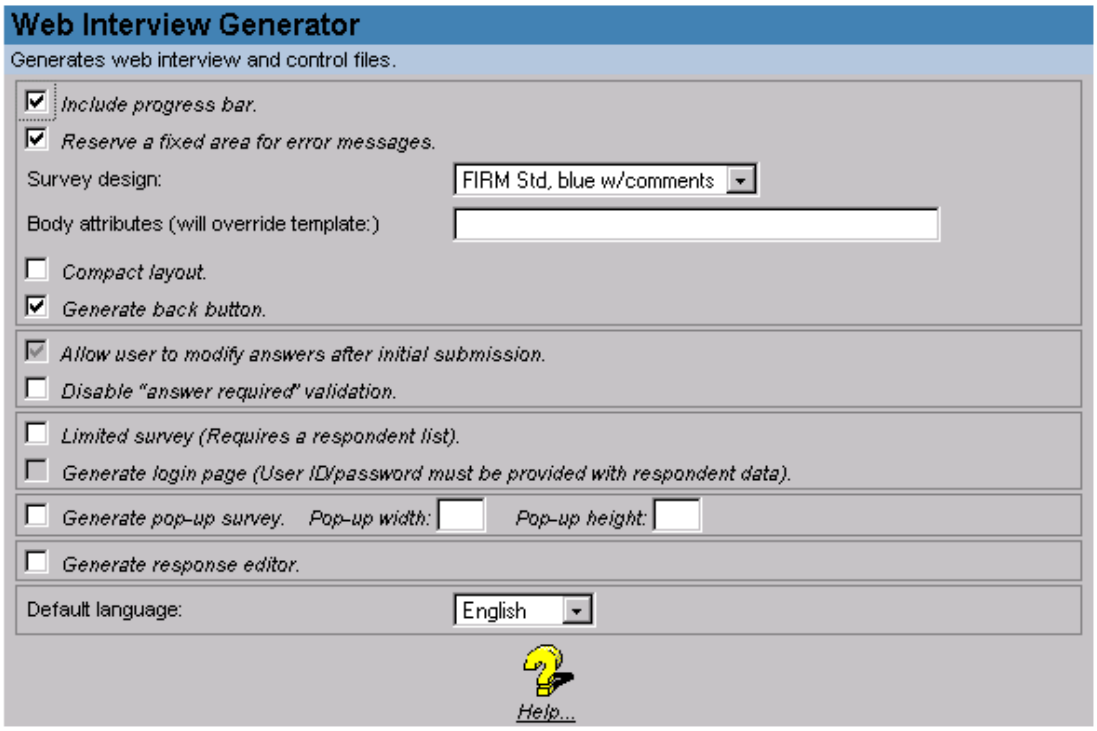

Figure 12: Web interview generator options

5.2.2 Options Controlling Appearance

5.2.2.1 Include Progress Bar

This option instructs the generator to include a progress bar on the interview pages. This bar indicates progress as a percentage of completed forms with respect to the total number of forms in the survey.

5.2.2.2 Reserve a Fixed Area for Error Messages

Select this option if you wish to reserve a small fixed area for error messages just below the page header. This prevents the rest of the page from jumping up and down if the interview program displays an error message. Because this area consumes space, you may want to uncheck this option, especially if you are creating a pop-up survey where the window size is limited.

5.2.2.3 Survey Design

A number of different templates are available that allow you to alter the survey's design. Select the one you desire from the list.

5.2.2.4 Body Attributes...

This option lets you override the HTML body attributes specified in the chosen template (to change the background color etc.)

Use of this option is deprecated. The preferred method of changing the survey's design is to create a new template with the template editor.

5.2.2.5 Compact Layout

This option controls the way in which the form inputs are displayed. Lists of inputs are displayed one after the other using the entire page width instead of one per line.

5.2.2.6 Generate Back Button

Back buttons allow the user to step backwards in the questionnaire and review/modify previous answers. This option is selected as default. It also forces the *Allow user to modify previous answers...* option described below to true.

5.2.3 Options Affecting Interview Behavior

5.2.3.1 Allow Modification of Previous Answers...

This option controls the way in which the generated interview handles reentry to a questionnaire page. Reentry can occur if a respondent uses the browser's back button, or if he/she returns to the questionnaire through using the URL received from FIRM requesting participation in the survey.

If the survey should be “one-shot”, then do not select this option. The generator then modifies the questionnaire flow according to the following scheme:

- If a respondent has completed the questionnaire and attempts to return to it, then the interview program displays a “Survey already completed” message.
- If the respondent returns to a partially completed interview, then a “You have already completed parts of the questionnaire. Press OK to continue” message is displayed.
- If it is detected that the browser's back-button has been used, then a “Previous answers may not be modified. Press OK to continue” message is displayed.

5.2.3.2 Disable “Answer Required” Validation

By default, the generator produces code that ensures that the respondent answers a question unless it has the “Not required” property set. This option is actually a “dirty” shortcut, which turns off validation for the entire interview, without consideration of this form property.

5.2.4 Options for Generating Limited Surveys

5.2.4.1 Limited Survey

The generator supports two kinds of surveys: limited and open. Limited surveys can only be accessed by respondents identified by the researcher in a respondent list. They are typically used for business-to-business or panel surveys. Each respondent is provided with a unique link to the survey, which includes identification credentials. Alternatively, he has to identify himself on a login page (See below).

Open surveys may be answered by anyone who desires to do so. As such, they are suitable for surveys where the respondents' identities cannot be established in advance, e.g. Web site evaluations.

5.2.4.2 Generate login page

As mentioned, *CONFIRM* provides a unique link for each respondent in a limited survey. As an alternative, the survey may include a login page where the user enters a user name and a password to participate. Check this option if you want this mode of operation. Remember to include user name/password information in the respondent lists.

5.2.5 Pop-Ups

5.2.5.1 Generate pop-up survey

Normally, an interview is ended by displaying a “Thank you” box that brings the user to a specified page when the OK button is pressed. This option alters the behavior of the generated interview to close the browser window when the interview is complete.

5.2.5.2 Width and Height

The width and height fields are currently only useful in test mode: they control the size of the preview frame so that you can get an idea of what the pop-up window will look like to the end user if those dimensions are used.

5.2.6 Default language

This option is only displayed for multi-language surveys. The language selected here becomes the default used when no language information is available in the request used to access the interview or in the respondent list.

5.2.7 Response editor

This option is not available in test mode. Checking this option instructs the generator to create a response editor, which is used for quality control and correction of collected data.

5.3 CATI files

[Not currently available from *"Test and verify"*]

To generate CATI-files, click *Administrate project→ Generate CATI files*. The generation-process will start immediately. One set of files for each project language will be generated. After creation of the interview files, the next logical step is to upload respondents (see "6.2 Uploading the respondent list").

6 Handling respondents

For CATI surveys or targeted, non-public web interview surveys, it is necessary to prepare and upload a list of respondents. This chapter describes the necessary steps needed to prepare the respondent list and upload it. Section 6.3 also describes how to automatically send e-mails with invitations and reminders for participants of web surveys.

6.1 Preparing the respondent list

The respondent list must be in the form of a tab delimited text file, with column headings in the first row. This file can for instance be exported from a spreadsheet application such as Microsoft® Excel™ by using the *Save As* function and choosing the format *Text (Tab delimited).*

This data file is prepared on the user's local machine and is subsequently uploaded to the CONFIRMsystem using the web browser.

The respondent data file must contain some or all of the following columns, depending on the type of survey. **Make sure that the spelling is exactly the same as in the table below.**

Column name	Description	CATI	Public WI	Private WI
name	The full name of the respondent	req		
tel	The respondent's telephone number	req		
language	The respondent's preferred language. The survey will be presented in this language			opt
sample_category	The sample category this respondent belongs to. More information on sample categories in section 7.1.1	opt		
email	The respondent's e-mail address			req
userid	The respondent's userid if logon is required			opt
password	The respondent's password if logon is required			opt

req = required opt = optional

In addition to these columns, any background variables defined for the survey must be represented as columns in the respondent data file. **The column names for these background variables must exactly match the form IDs given to these background variables in the questionnaire designer.**

6.2 Uploading the respondent list

Once the respondent data file has been prepared and saved as a tab delimited text file, it is ready to be uploaded to ConFIRM. This is accomplished by selecting the menu *Administrate project | Upload respondents*. Simply press the *Browse…* button, which will invoke the familiar Windows™ File Open dialog, allowing the user to browse through the file system and select the appropriate respondent data file and press the *Open* button in the file dialog.

Once the correct data file has been selected press the *Upload…* button to start uploading the respondent list to the CONFIRMsystem. After the file has been successfully uploaded, the user is presented with three options:

- **Yes, append:** This option will append the newly uploaded respondents to any existing respondents already in the database.

- **Yes, overwrite:** This option will delete any existing respondents in the database before adding the newly uploaded respondents. **Use this option with caution.**
- **No:** This option will abort uploading respondents and return to the *Project overview* page.

If uploading is confirmed, a status screen will appear after a few seconds, summarizing the number of respondents that was uploaded.

6.3 Sending e-mail

ConFIRM provides a powerful wizard for sending e-mails to web survey respondents. On the basis of several selection criteria, e-mails can be sent to all or a subset of the respondents in order to invite them to participate in a web survey or to remind them of a previous invitation. It is even possible to send "thank you" e-mails to respondents that have completed a web interview.

The e-mail wizard is accessed through the *Administrate project* | *Email setup* menu. The wizard will take the user through the following steps:

1. **Respondent selection:** Select which respondents should receive the e-mail and specify the content of the e-mail.
2. **Preview and confirmation:** Preview which respondents will be mailed and how the e-mail will look.
3. **Sending e-mail:** Send the e-mails to the selected respondents.

More on these steps in the following sections.

6.3.1 Step 1: Respondent selection

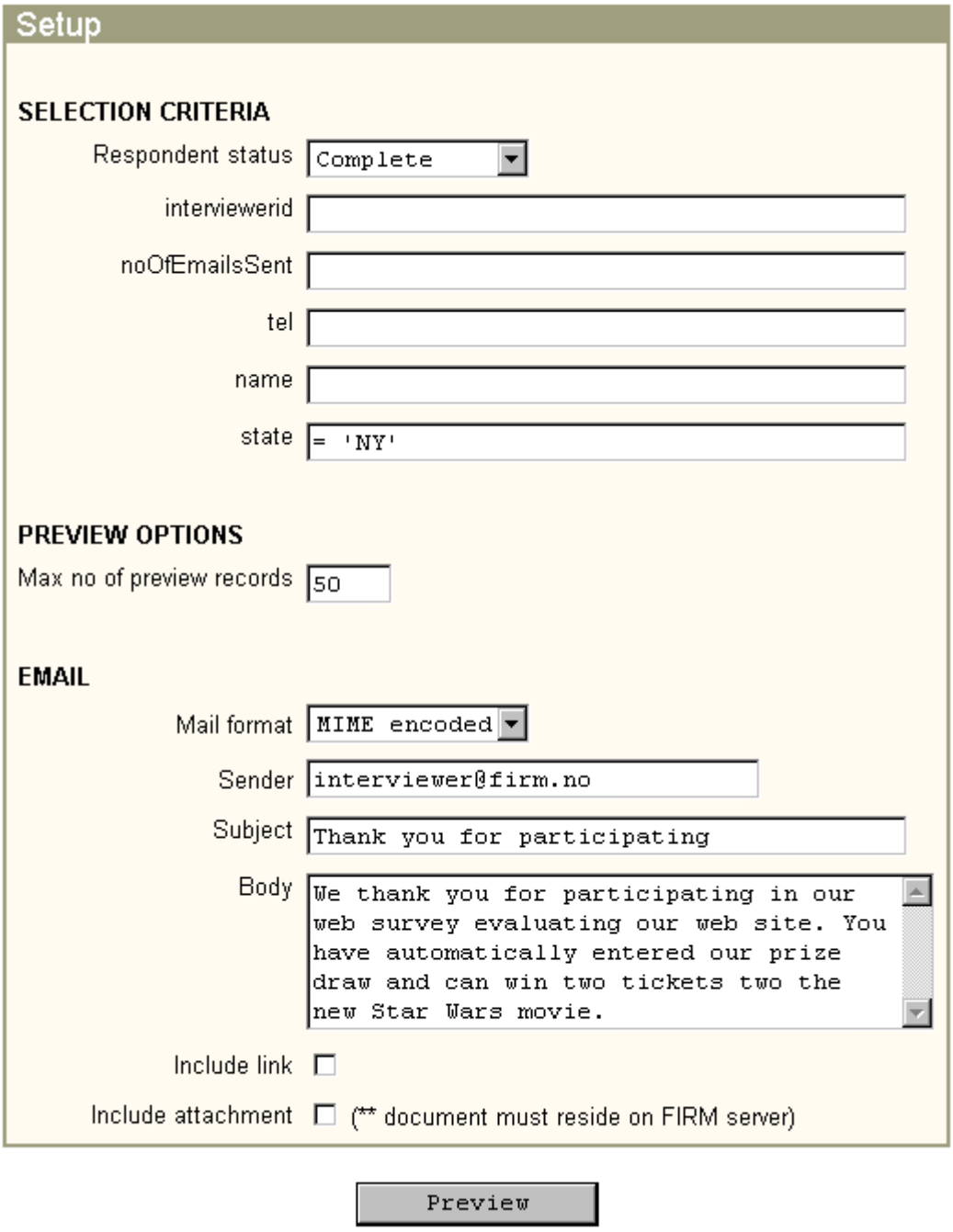

Figure 13: E-mail setup

In this step, the user enters the selection criteria that will determine which respondents will receive the e-mail. The criteria are:

Criterium	Description
Respondent status	Choose the status of the interview, i.e. whether the respondent has answered the survey or not, whether he or she was screened or abandoned the survey in mid-interview. Use for instance the *Completed* status to select respondents for "thank you" e-mails, or the *Not answered* status to send reminder e-mails.
Number of emails sent	The number of e-mails sent is recorded for each respondent, and can be used as a selection criterion.
Other fields	The wizard will display any additional fields that can be used for selection, including sample categories and quota variables.

Table: Respondent selection criteria

To make the selection abilities as flexible as possible, the selection criteria must be input in the *Structured Query Language (SQL)* syntax known from relational database queries.

Examples:

Assume that the respondent data includes gender as a background variable with precodes 'M' and 'F' for male and female respondents, respectively. The e-mail wizard would then list *gender* as one of the possible selection criteria. To select only male respondents one would enter

```
= 'M'
```

in the *gender* input field. Selecting based on a set of values would be done with the *IN* keyword. Selecting both male and female respondents (for the example's sake) is done with the criteria

```
IN ('M', 'F')
```

in the *gender* input field. Selecting respondent that have already received more that 2 e-mails could be accomplished by entering

```
> 2
```

in the *noOfEmailsSent* input field. Similarly, all standard SQL keywords and operators can be used, such as *BETWEEN*, <, <=, >, >=, <>, *AND*, *OR* etc.

Under *Preview option* the user can enter the maximum number of respondents that will be displayed in the preview list in step 2 of the e-mail wizard. If nothing is entered here, no preview respondents will be listed. This is merely a precaution when working with large numbers of respondents.

Under the *E-mail* heading the user can enter details about the e-mail he wishes to send to the selected subset of respondents:

Field	Description
Mail format	Choose between sending a MIME encoded or plain text e-mail.
Sender	Enter the e-mail address that is to appear in the *From* header field in the e-mail.
Subject	Enter the subject header of the e-mail.
Body	Enter the text that is to appear in the body of the e-mail message.
Include link	Select this checkbox to automatically have a link (URL) to the web interview included in the e-mail.
Include attachment	It is possible to send attachments with the mail. For this feature FIRM support is needed.

Table: Email setup fields

By pressing the *Preview* button, the user is taken to step 2 of the e-mail wizard.

6.3.2 Step 2: Preview and confirmation

In this step, a preview list of respondents shows which respondents will receive an e-mail, based on the selection criteria entered in step 1. This list is limited by the maximum number specified under *Preview options* in step 1.

In addition, a preview of the e-mail that is to be sent is shown.

At this point, the user can choose to cancel the e-mail wizard altogether and return to the *Project overview* page; he can choose to go back to step 1 to edit the selection criteria; or he can choose to proceed with sending e-mails and go on to step 3 of the wizard.

6.3.3 Sending e-mails

This is the final step of the e-mail wizard. At this point the e-mails will be queued for immediate delivery to the selected respondents.

In order to avoid unnecessary failure messages and inquiries from respondents when running limited Web-surveys, we strongly recommend to include the following information in the e-mails:

Attention!

Please, read the following before you enter the survey!

- Navigate forth and back by clicking on the >> and << buttons. Do not use the browser`s “back” and “forward” buttons.
- Use the link in the e-mail message when you wish to continue responding to the survey after a pause. Do not bookmark the survey page!
- Cut (Copy) – and – Paste the link into the Internet address window in case you cannot access the survey by simply clicking on the link.
- The link is unique! Please, do not send it to other respondents.
- Your answers are saved only when you navigate forth (or back) from the particular page.
- You complete the survey by clicking on the OK button on the last page.

7 CATI – Computer Aided Telephone Interviewing

7.1 CATI Administration

The CATI administration module is accessed through the URL http://www.firm.no/wi. After logging on, the user is presented with the following options:

- **Sign on this computer as an interview workstation:** Allows an interview manager to sign on the current computer as an interview workstation for his company. This will enable individual interviewers to start interviewing at any time from the current computer without having to log on to the system, simply by accessing the URL http://www.firm.no/wi/select_interviewer.asp. It is recommended that the interviewers have this URL set up as their default start page ("home page") in their web browsers, as this will make the "Select interviewer" page appear when the interviewer starts the browser.
- **Sign off this computer as an interview workstation:** By selecting this option, an interview manager will sign off the current computer as an interview workstation for his company. This means that interviewers will no longer be able to perform interviews from the current computer, and will not gain access to the "Select interviewer" page.
- **Interview management:** Starts the module for managing performed interviews. This module is described in detail in chapter 8.
- **Register interviewers:** Allows the interview manager to register the names and user IDs of their interviewers. More about this below in chapter 7.1.1.
- **Select interviewer:** Selecting this option opens the "Select Interviewer" page, allowing interviewers to select their identity and start interviewing.

In addition, a status bar indicates whether the current computer is signed on as an interviewer workstation.

7.1.1 Setting up interviewers

By selecting the "Register interviewers" option on the "Interviewer management" page, the interview manager is able to maintain information about the individual interviewers, as well as register new interviewers. The user is presented with a list of interviewers, containing the following information:

Field	Description
Username	A short ID uniquely identifying the interviewer.
First name, Last name	The interviewer's full name.
Assigned project	The project ID of the survey the interviewer is assigned to work with. The project ID can be found on the "Project Overview" page in the main CONFIRMapplication.
Language	The interviewer's preferred language. Each interviewer will be given the questions in the language specified here, provided that this is a valid language for the assigned survey.
Sample category	It is possible to assign each interviewer to a specific subset of the respondents. This is useful, e.g. when interviewing in several languages, as each respondent can be assigned to a sample category according to language. Each interviewer may then be assigned to the appropriate sample category according to his or her spoken language. Make sure to double-check the spelling of the sample category entered in this field.

Table: Interviewer setup

The *Assigned project*, *Language* and *Sample category* information can be altered for one or many interviewers, and the changes saved by pressing the *Update info* button.

For the changes to take effect correctly, the interviewer must have quit his current interviewing session before the changes are made. It may also be necessary to press the *Refresh* button in the interviewers web browser before commencing the first interview.

To add a new interviewer, simply fill out the information fields at the end of the list and press the *Add* button.

To delete an existing interviewer, press the *Del interviewer* link at the far right of the appropriate row in the list.

7.2 Quotas

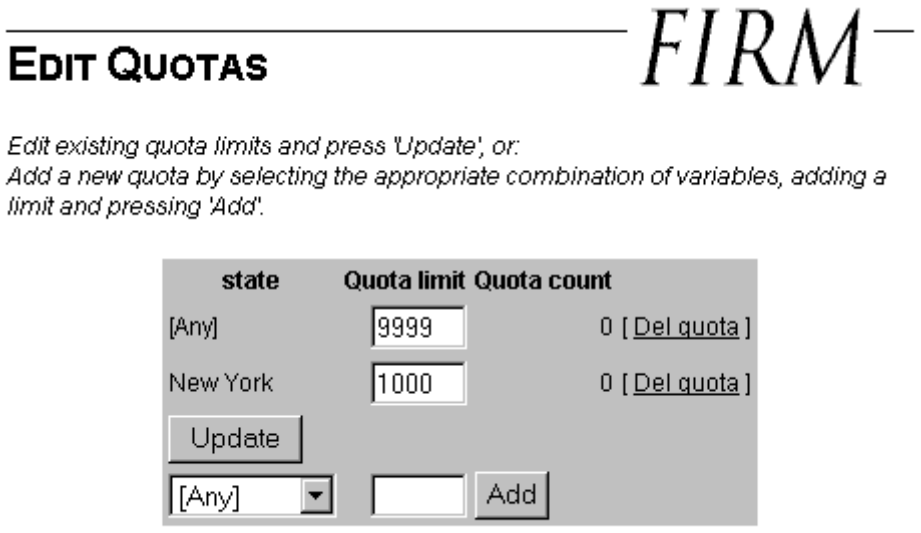

Figure 14: The quota editor

ConFIRM has an advanced quota system that allows a manager to impose quotas on the pool of respondents. The quotas can be defined before the interviewing starts or while interviewing is in progress. They can also be redefined or limits altered at any time.

A typical use of quotas is to specify the number of male and female respondents that are required, ensuring that e.g. 500 men and 500 women are interviewed, although the pool of respondents may be biased in terms of gender.

The quota system is most effective when used in conjunction with background variables only; i.e. the quota variables are also background variables where the values are known beforehand for each respondent. In this case, CONFIRM will remove respondents that belong to filled quotas from the respondent pool.

Quotas may also be defined in terms of variables that are determined by the response to a question in the interview, but at present there is no automatic way of terminating the interview if the respondent turns out to belong to an already filled quota. This is likely to change in a future version of ConFIRM. Until then, interviewer managers will have to manually keep track of quotas, and instruct interviewers of which interviews to abort manually (see section 7.3.3) to avoid oversampling.

A single form is defined as a quota variable by ticking the appropriate checkbox in the *Properties* page in the CONFIRMquestionnaire designer. **Defining a new quota variable has no effect until the production environment has been recompiled and the database has been updated.**

The quota editor is available through the *Administrate project* menu in the main CONFIRMapplication. The editor consists of one column for each defined quota variable, along with a column showing the quota limit and the current quota count, i.e. the number of interviews performed in this quota.

To define a new quota, simply select the appropriate combination of values in the drop-down lists for each quota variable, enter the desired quota limit and press the *Add* button. The *[Any]* option represents a quota that will comprise any value for this quota variable.

It is important to realize that it is necessary to define quotas for all values of the quota variables, either by explicitly defining quotas for each value or by using the *[Any]* option. Otherwise, respondents not meeting any of the quota criteria will be considered to be outside the quotas, and will not be in the respondent pool.

> **Example:**
>
> You have defined the home state of the respondent as a quota variable, and it is also a background variable. You now wish to interview exactly 1000 respondents from the state of New York, but want as many respondents as you can get from the other 49 states until the New York quota has been filled. You would intuitively define a quota for the New York value of the state background variable with a quota limit of 1000. To achieve the desired effect, you would also have to define similar quotas for the rest of the states, setting the quota limit for these quotas high enough so that the New York quota would be filled before any other quota. Alternatively, you could define an *[Any]* state quota in addition to the New York quota. Please notice, however, that New York respondents also would be considered in this quota, so be sure to set the quota limit high enough.

To alter the quota count of any existing quota, simply edit the appropriate field in the quota editor and press the *Update* button.

To delete a quota, select the *Del quota* link to the far right in the quota list.

Any alteration of the quota limits, addition or deletion of quotas will trigger a recalculation of the quota counts and respondent pool to reflect the changes.

7.3 CATI Interviewing

Before CATI interviewing can commence, each interviewer station need to be signed in as a workstation by an interview manager in the fashion described in section 7.1. In addition, each interviewer must be registered with a user ID, an assigned project and a sample category (if applicable) as described in section 7.1.1.

The interviewer starts his interview session by accessing the “Select interviewer” page URL http://www.firm.no/wi/select_interviewer.asp, selects his identity by clicking the link with his name, and thereby enters the CATI system, which will be in *pause mode*.

7.3.1 Pause mode

In pause mode, the interviewer is presented with two choices in the form of two buttons:

- **Start interviewing:** By selecting this option, the CATI system will enter into *Search mode*, allowing the interviewer to start searching for respondents.
- **Quit interviewing:** This option ends the interview sessions, and returns the browser to the “Select interviewer” page, allowing another interviewer to use this interview station.

7.3.2 Search mode

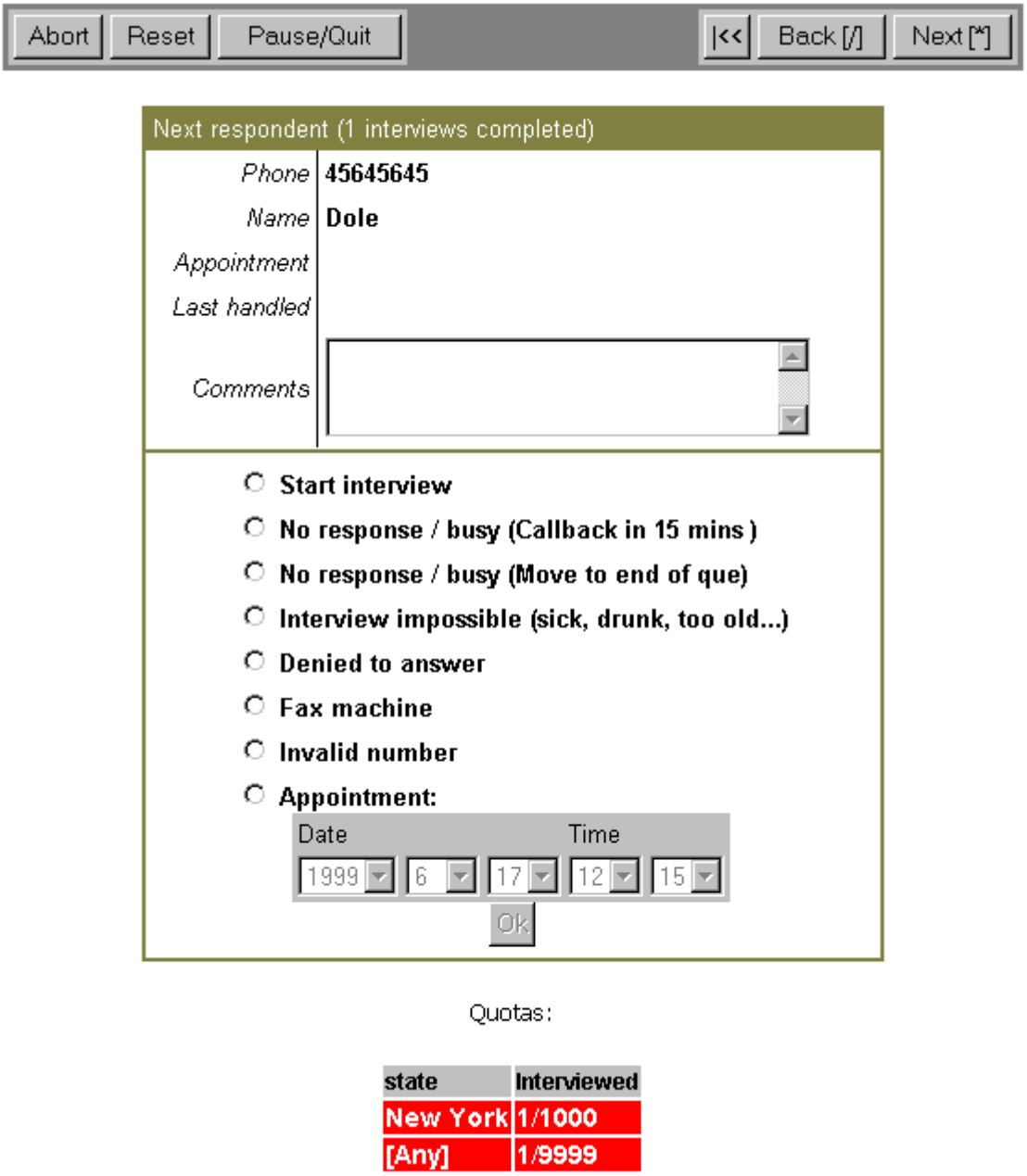

Figure 15: The CATI system in search mode

In search mode, the CATI system pulls a respondent from the respondent poll and presents the interviewer with the following information about the respondent that he is about to call.

Field	Description
Phone	The telephone number the interviewer is supposed to call.
Name	The name of the respondent
Appointment	If an appointment has been made with this respondent to call him back later, the date and time that was agreed is shown here.
Last handled	Shows the date and time that this respondent was last handled, i.e. pulled from the respondent pool and a call attempt was made.
Comments	An open text field where the interviewer can enter comments about the respondent, e.g. more details on an appointment, or the correct phone number.

Table: CATI fields

Below the respondent information area, are several radio button that the interviewer can choose from, depending on the outcome of the call:

Option	Description
Start interview	The respondent answers and agrees to the interview; take the CATI system to *interview mode* and commence interviewing
No response / busy (callback in 15 mins)	No answer; make an "appointment" to call back in 15 minutes.
No response / busy (move to end of queue)	No answer, but only pull this respondent again when all other respondents have been attempted.
Interview impossible (sick, drunk, too old…)	The interviewer judges that the respondent is unable to understand or complete the interview progress. This respondent will removed form the respondent pool and not called again for the duration of the survey.
Denied to answer	The respondent refused to participate in the survey.
Fax machine	The interviewer reached a fax machine on the assigned number.
Invalid number	The number is invalid or non-existing.
Appointment	Make an appointment with the respondent to call back at a later time. The interviewer selects the appropriate date and time before selecting the *OK* button.

Table: CATI interviewer options

After an interview is completed, the CATI system will immediately pull another respondent**. To ensure that search times are correctly calculated, the interviewer should press the *Pause/Quit* button at each break, scheduled or unscheduled, to return the system to pause mode.**

At the end of an interview session, the interviewer should exit the system in a controlled manner, by pressing the *Pause/Quit* button to enter pause mode and selecting *Quit interviewing*.

At the bottom of the screen, the interviewer can keep track of the count for each quota. The quotas that the current respondent are in are shown highlighted with red color.

7.3.3 Interview mode

In interview mode, the CATI system presents one question at the time, showing the question text, any special instructions to the interviewer and the alternative answers. The interviewer simply reads the question text to the respondent and records his answer. If possible, the CATI system moves automatically to the next question.

At the top of the screen is a toolbar with the following buttons:

Button	**Description**
Abort	Aborts the interview. This option is used if the respondent for some reason does not want to complete the interview. Selecting *Abort* marks the interview as aborted, takes the respondent out of the respondent pool, and returns the interviewer back to search mode with another respondent.
Reset	Takes the interviewer back to search mode with the same respondent, resetting all answers given in this interview.
Pause/Quit	Takes the CATI system to pause mode. This option should only be used when in search mode.
|<<	Navigation button. Moves the interviewer back to the first question.
Back	Navigation button. Moves the interviewer back to the previous question. The same effect can be accomplished by pressing the backslash character (/) on the numeric keypad.
Next	Navigation button. Moves the interviewer to the next question. The same effect can be accomplished by pressing the asterisk character (*) on the numeric keypad.

Table: CATI buttons

It is possible to use either the mouse or the keyboard for selecting answers in interview mode. The keyboard shortcut is always listed in square brackets (e.g. [1]) next to the answer alternative.

8 Interview management

The interview management module is designed to provide a versatile and flexible way of handling interview data. It allows an interview manager to run queries to select a subset of interviews based on interviewer data; interview times, status and language; responses to individual questions, as well as combinations of these criteria. Once a subset of interviews has been retrieved, it is possible to alter the responses of a single interview, alter a response for all of the interviews or view various statistics concerning the interviews.

This module is accessed through the CATI Administration module as described in section 7.1. After selecting *Interview management* a list of active CATI projects will appear allowing the user to select a project to work with.

The main interview management screen is divided into three separate frames or panes:

- The topmost frame contains the search criteria and the result list after a query has been performed.
- The frame on the bottom left contains the survey routing represented as a tree structure in the same manner as in the questionnaire and report designers. This is primarily used in conjunction with the *Condition builder*, see section 8.1.3.
- The rightmost bottom frame is the work area, and will contain differing information according to which action is currently performed.

8.1 Retrieving a subset of interviews

In order to make retrieving a subset of interviews as easy as possible, but still providing maximum flexibility, the interview management module provides a simple user interface for selecting simple query criteria. To cater for more complex needs, more advanced options are available that will allow an experienced user to specify criteria that should suffice in the vast majority of cases.

8.1.1 Simple queries

The fastest and easiest way of specifying an interview query is to use the input fields directly below each column header in the query frame. These are:

- **RespID:** The ID of the respondent that answered the interview. CONFIRM will match all RespIDs beginning with what you enter here.
- **Name:** The respondent's name. CONFIRM will match all names beginning with what you enter here.
- **Interviewer:** When selecting this input field, it will expand, revealing a list of all interviewers that have interviewed respondents for the current survey. Select several interviewers by holding the CTRL key down while clicking on each interviewer. CONFIRM will match all interviews performed by the selected interviewer(s).
- **Interview start, Interview end:** Choose the time period to match interviews that were performed, today, last two days, etc. To specify more detailed times, see how to specify an advanced query in section 8.1.2.
- **Status:** This input field operates in the same manner as the *Interviewer* field. CONFIRM will match all interviews having the status selected here.
- **Language:** This input field operates in the same manner as the *Interviewer* field. CONFIRM will match all interviews performed in the specified language.

Press the *Search* button to execute the search and retrieve a list of matching interviews. Pressing *Search* without specifying any further criteria will retrieve a list of all interviews.

Press the *Clear* button to clear all criteria.

8.1.2 Advanced queries

For more advanced search options, select the *Advanced* checkbox, and two new input areas and a button will appear:

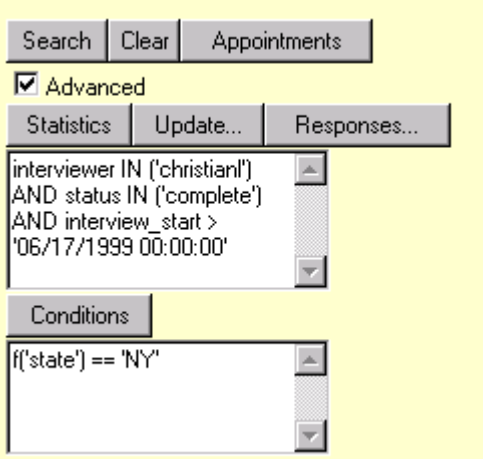

Figure 16: Using advanced query options

- **WHERE area:** This is the topmost input area that becomes visible when selecting *Advanced*. When this area is visible, all selections in the criteria input fields will be translated into the form of a SQL WHERE-clause. This clause is fully editable, and allows an advanced user to specify complex queries using the SQL language.
- **Conditions button:** By pressing this button, the user will invoke the *Condition builder*, making it possible to retrieve interviews based on what the respondent has answered on one or more questions. This is described in detail in section 8.1.3.
- **Condition area:** In this area the conditions entered with the aid of the condition builder will be set for easy editing by an advanced user. It is also possible to enter a condition expression directly here without using the condition builder.

8.1.3 Using the condition builder

The condition builder is based upon the *Expression builder* that should be familiar from the questionnaire designer when specifying logic nodes. The basics of operation are the same.

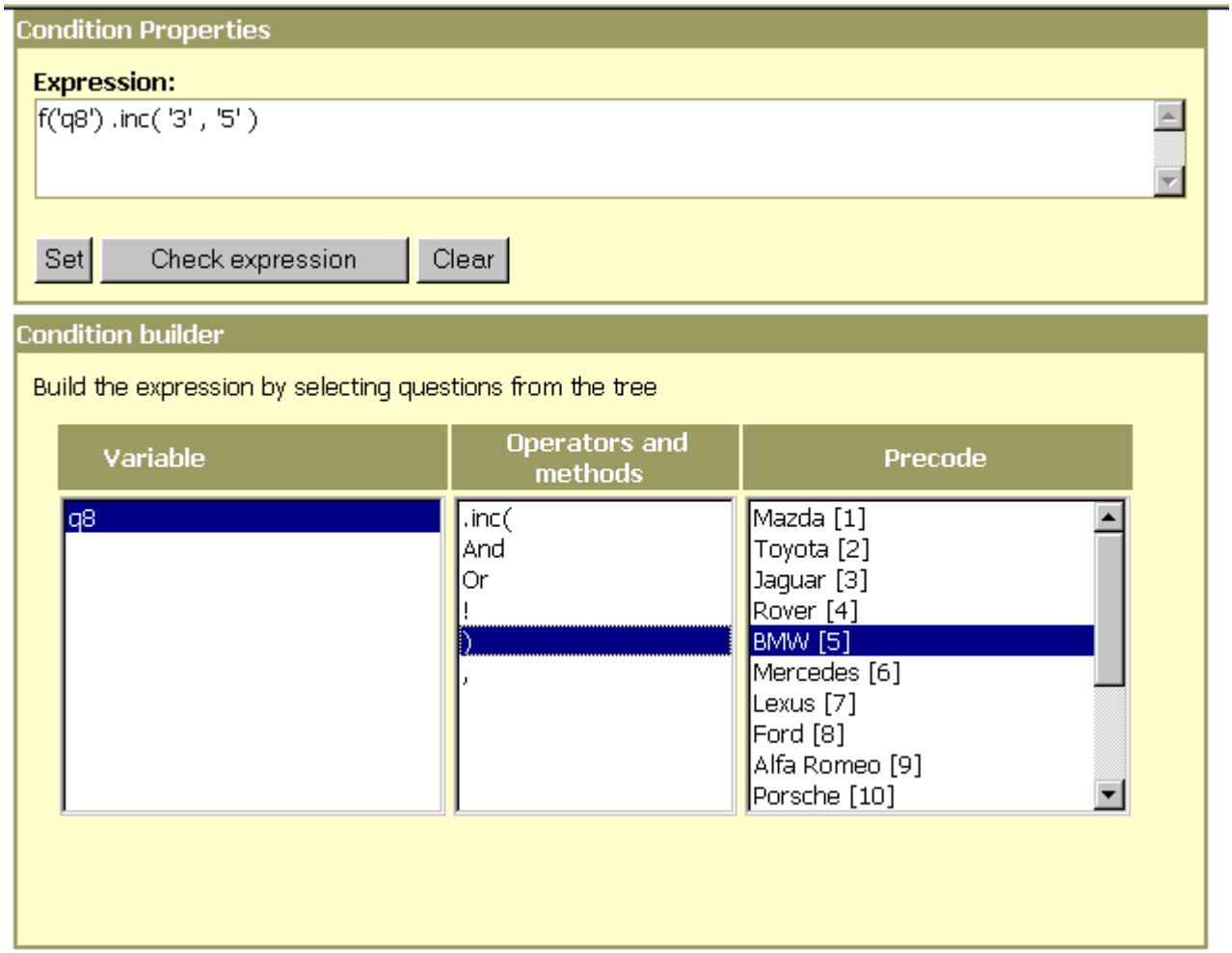

Figure 17: The condition builder

1. Select a form in the routing tree on the left hand by clicking on it with the mouse. This will initialize the condition builder with the valid variables and precode values.
2. Select the correct variable in the leftmost list in the condition builder. In all but grid forms there will only be one option here.
3. Select the appropriate operator or method in the middle list. For an explanation of these, see chapter 4.8.5 on the expression builder.
4. Select the appropriate precode(s) and complete the expression so that it forms a well-formed condition.
5. If applicable, select a conjunctive operator (*And*, *Or*) and expand the expression by repeating the steps above.

Tip: Selecting a logic (branching) node will append the expression for this node to the condition expression.

Press the *Check syntax* button to validate the syntax of the expression.

Press the *Set* button to transfer the condition expression to the condition area in the query frame.

Press the *Clear* button to clear the current expression and start over.

8.2 Updating one respondent's answers

Once a list of interviews has been retrieved in the query frame, the *response editor* can be invoked on an interview by clicking on the appropriate row in the result list. The response editor will appear in the work frame on the bottom right. It is operated as following:

1. Select the desired form in the routing tree on the left.
2. Alter the answer.
3. Press the *Save* button at the bottom to save the changes.

Be careful with background variables, as these will appear in the response editor as "ordinary" single question. A warning text will identify a question as a background variable.

8.3 Updating several respondents' answers

It is possible to update the response to a question for the entire result set of interviews. This is done by following these steps:

1. Retrieve the desired subset of interviews as described above.
2. Press the *Update…* button to invoke a variety of the condition builder that helps build the update clause.
3. Select the form that is to be updated in the routing tree on the left.
4. Build the update expression in the same manner as described above in section 8.1.2.

Press the *Update* button to alter the responses of all the interviews in the current result list. Be extremely cautious when performing this operation. This will alter your production data in an IRREVERSIBLE manner!

9 Reporting

9.1 Introduction

All reporting functionality in CONFIRM is accessed by clicking *Online Reporting* on the main meny. When selected, you enter this screen:

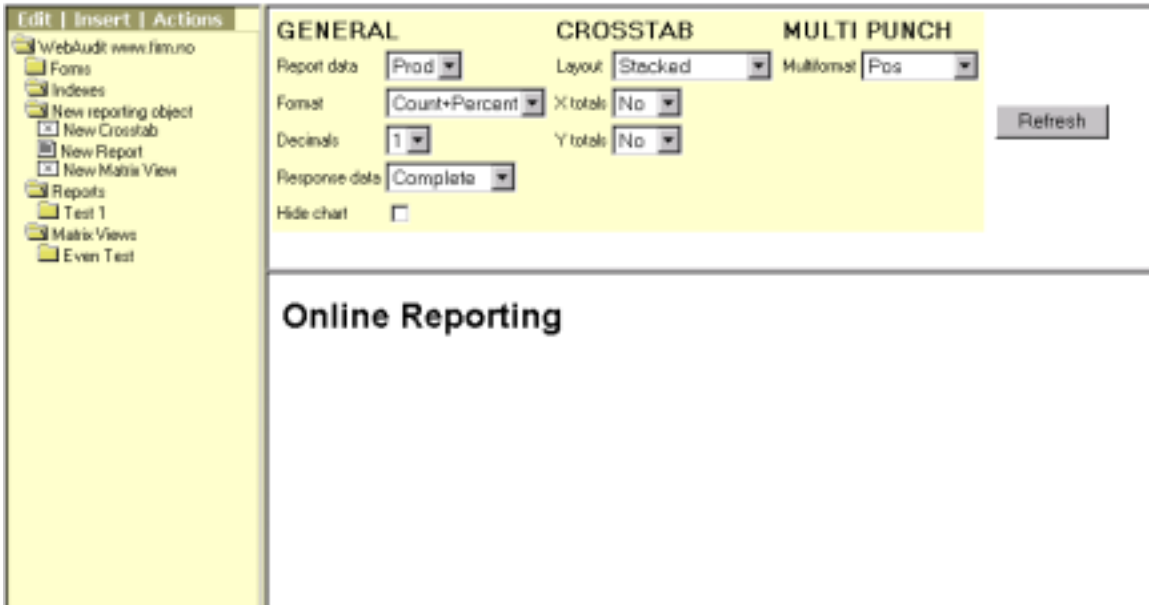

Figure 18: Initial reporting screen

This screen consts of tree frames:

- **The reporting tree.** A tree containing all your reporting objects. That is forms, new reporting objects, reports and matrix views. The menu on top of the tree is called the report tree menu.
- **The settings frame.** A frame with global settings for the reports. These global settings can be overridden by setting specific reportnode properties.
- **The report frame.** The frame where all report tables and charts are displayed.

9.2 Checking a single question

The easiest way of checking the answers of a single question is by entering the *"Form"*-folder in the reporting tree:

1. **Open the *"Form"*-folder.** This will show you the routing the same way as the *"Routing"*-folder in the questionnaire.
2. **Locate the question.** Since the subfolders, condition trees and loops are all initially closed, it is crucial to know something about the questionnaire touring to be able to find the question of interest.
3. **Double click the question.** When the question is double clicked, the result is shown in the bottom right frame as shown in the Figure 19.

Figure 19: Reporting a single question

When checking results directly from the *"Form"*-folder, all settings are either default settings or the settings from the *"Settings"*-frame.

9.3 Building reports

Within a project, all the reports are collected in the report folder. The idea by having more than one report is that you can order PowerPoint and Excel files at the report level. The following actions are possible to take at the report level:

- **Create a new report.** A new report is created by dragging a *New Report* object from the *New reporting objects* folder into the *Reports* folder.
- **Change name.** Select *Edit → Properties* in the report tree menu, and change the report name from the report frame.
- **Add single question to the report.** Drag the form from the *Forms* folder and drop it at the right location within the report.
- **Add crosstabulation to the report.** Drag a *"New crosstab"*-object from the *New reporting objects* folder and drop it at the right location within the report. See chapter 9.4 for details on building crosstabs.
- **Delete.** Select *Edit → Delete* in the report tree menu.
- **Duplicate.** Select *Edit → Duplicate* in the report tree menu.
- **Showing the whole report.** Select *Actions → Show Report* from the report tree menu. This will display the whole report in the report frame.
- **Order report.** Select Actions-Order Report from the report tree menu. By specifying a PowerPoint template and email address in the report frame, Excel sheets and/or PowerPoint slides will be sent to the email recipient.

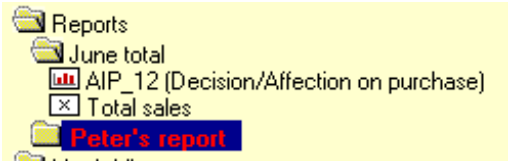

Figure 20: Project with two reports.

Figure 20 shows a report, *June total*, with one crosstab and one single punch question.

9.4 Building crosstabs

When a new crosstab is made within a report, the crosstab object includes got two subfolders:

- The *"Crosstab..."*-folder where the vertical questions of the crosstab can be included. This is done by dropping single questions within the folder.
- The *"With..."*-folder where the horizontal questions of the crosstab can be included. This is done by dropping single questions within the folder.

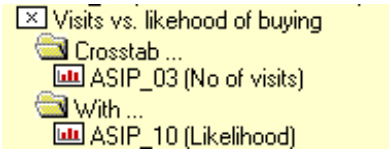

Figure 21: Crosstab: No of visits vs. likelihood

9.5 Settings

9.5.1 Global settings

The settings frame in the reporting screen includes the following global settings for the whole reporting environment.

Setting	Setting description
General	
Report data	Indicates whether test or production data should be reported.
Format	Indicates whether count, percent, count+percent or average should be displayed. Average can only be displayed for weighted single punch or grid questions. See chapter 4.2.1.2 for how to weight a question.
Decimals	Number of decimals to use in average and percent format.
Response data	Normally set to show only complete interviews, but can also show only incomplete interviews or both complete and incomplete.
Hide chart	Use this check box to hide all charts in the report frame.
Crosstab	
Layout	**Stacked:** The questions in the *"With..."*-folder are stacked. **Stack+Nesting:** The questions in the *"With..."*-folder are nested.
X totals	Set this to *"On"* to the totals for a question to the right of the rest of the crosstab table.
Y totals	Set this to *"On"* to the totals for a question under answer alternatives.
Multi Punch	
Multiformat	**Pos:** Show only numbers of positive answers for a multi punch. **Neg:** Show only numbers of negative answers for a multi punch. **Ordered:** Show an ordered multi punch split on order of answering.

Table: Global report settings

Figure 22: Global settings frame

9.5.2 Report elements settings

All of the global settings can be overridden by setting properties for an element within a report. The element would be a crosstab or a simple question within the report. This is done from *Edit* → *Properties* in the report tree menu. From this menu point, you can also set the following properties that are not available from the global settings:

Setting	Description
Chart options	
Type	Defines the chart type to be used. The most normal chart types like bars, pies, lines and areas are supported among some other.
3D	The chart can be view in 3D.
Stacking	The chart is stacked.
Data in rows	Uses data in rows instead of data in columns. This is the terminology of the chart component used. Excel uses the opposite terminology: Series in columns and series in rows.
PowerPoint options	
Header text	The header text in a PowerPoint slide. This text can also be set to give a name to a crosstab in the reporting tree.
Text	The text in text frames in PowerPoint slides.
Object type	Used to choose between table and chart as the PowerPoint object.
Slide layout	**Object left/Text right:** The object (table or chart) is in the left of the slide and the text is in the right of the slide. **Text left/Object right:** The text is in the left of the slide and the object is in the right of the slide. **Only text:** Only text is shown in the PowerPoint slide. **Only object:** Only object is shown in the PowerPoint slide.
Filter options	
Active filter	It is possible to set any of the defined filters active for one report element.

Table: Report item settings

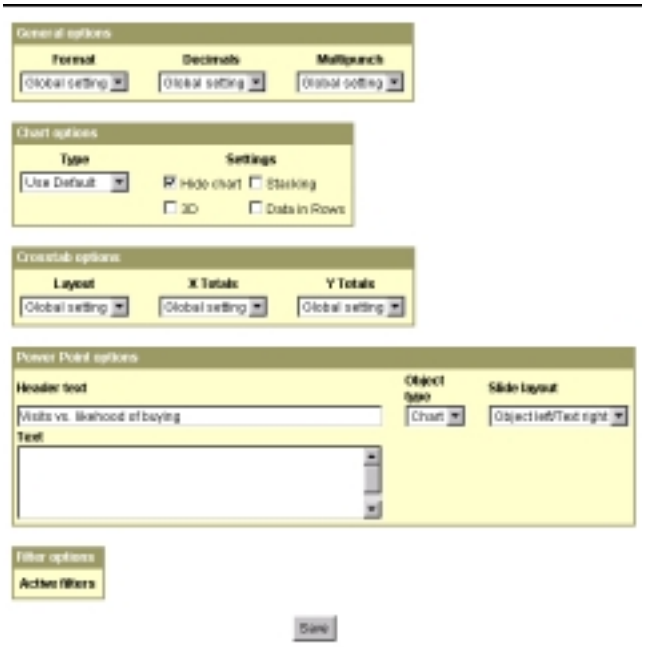

Figure 23: Report element settings

9.6 Filters

9.6.1 Defining filters

Filters are defined like this:

1. Select the question to be filtered within the *"Form"*-folder.
2. *Actions* → *Add Filter* from the reporting tree menu
3. Enter the filter name (e.g. Male).
4. Select the precodes to let through the filter (e. g. the male precode if only male respondents are to be counted).

Figure 24: Filter definition

9.6.2 Applying filters

Filters can be added globally or at the report element level.

- **Global filters:** Select *Online Reporting → Define filters* from the main CONFIRM menu. Then check the correct filter in the list.
- **Element level filters:** See Filter options in 9.5.2.

9.7 Matrix Views

9.7.1 Introduction

A matrix view is mainly an end user interface, but can also be used by research analysts. The concept is to select tree sets of question.

Question set	Description
Indicators	These questions are placed in the left part of the matrix. Only these questions can be reported as single questions within the matrix view.
Demographics	These questions are placed in the bottom part of the matrix. The indicators can be crosstabulated towards the demographics.
Filters	The filter questions will be available in two filter dropdown lists at the top of the matrix. This means that results can be filtered on two questions simultaneously. In one filter you can chose one or more answer alternatives to let through the filter.

Table: Matrix view question sets

9.7.2 Configure matrix views

The matrix views are configured from the reporting tree. The following operations can be done on the matrix view:

- **Create a new matrix view.** A new matrix view is created by dragging a *"New Matrix View"*-object from the *"New reporting objects"*-folder into the *"Matrix Views"*-folder.

- **Change name.** Select *Edit → Properties* in the report tree menu, and change the matrix view name from the report frame.
- **Add question to a question set.** When a new matrix view is created, the matrix view element in the reporting tree includes three subfolders, one for each question set. Questions are added to the subfolders by dragging them from the *"Form"*-folder.

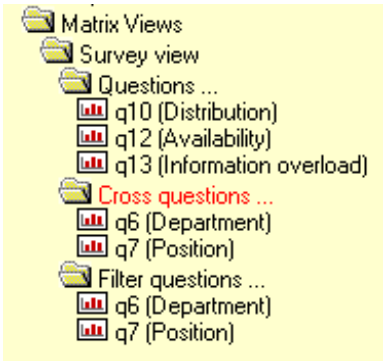

Figure 25: Matrix view configuration

9.7.3 Using matrix views

When the matrix view is initially entered, no charts or tables are displayed. The table below shows what options that are available in the matrix view. The chart or table is refreshed based on the options every time the user presses *"Refresh"*.

Option	Description
Select indicator question	By pressing a question part of the matrix, an indicator question is selected. The question can be selected on and off by pressing them repeatedly. Dark color of the button indicates a selected question
Select a demography question	Demography questions are operated the same way as the indicator questions. When a demography question is selected, the selected indicators are crossed with this demography question.
Make filters available	Select the *"Filters"*-option
Apply a filter	The question to filter on is selected from a dropdown lists in the top of the matrix. When a question is selected, a row of answer alternatives appears right of the dropdown list. The alternatives to let through the filter should be selected.
Change display	The display can be switched between chart and table by changing the *"Display"*-option.
Change format	Format can be switch between count, percent and index (average) by changing the *"Format"*-option.
Change matrix size	The matrix size can be changed by changing the *"Screen"*-option. There are predefined matrix sizes defined to fit the following screen resolutions: 800x600, 1024x768 and 1280x1024
Make timeseries available	Select the *"Timeseries"*-option. This will add a timeseries row below the chart area.
Apply a timeseries	In the timeseries row, periods can be changed between weeks, months or years. When e. g. months are chosen, the months where there have been any respondents are displayed. When the trend button is selected, a line chart of the different weeks is displayed. When selecting a single week, this week is added as a filter.

Table: Matrix View operations

Figure 26: Matrix view

There are some rules and guidelines for what can be done and what cannot be done within a matrix view. These rules and guidelines are collected in the list below:

- To be able to display a chart, at least one question has to be selected in the right question column.
- Index format can only be used with single punch and grid questions.
- Index format can only be used with weighted questions, see chapter 4.2.1.2 for how to weight a question.
- More than one question in the left question list can only be selected when dealing with indexes.
- A filter can not be selected without selecting any of the answer alternatives.
- Maximum 2 questions can be selected from the demography list. The crosstabs generated are nested by the demography questions. Nesting of more than two questions is rarely necessary, and would jeopardize the performance of the reporting server too much.

9.8 SPSS exports

To generate a SPSS export, select *Online Reporting → SPSS Export* from the main menu. The export is generated and sent to the default mail address of the current user.

9.9 Recoding data

Forms in the questionnaire can be turned into recoded variables. This causes an expression column to appear in the answer list of the question. These expressions should be programmed as a Boolean expression evaluating to true or false.

9.9.1 Expression evaluation

The recoded variables are calculated for each respondent. When calculating a single punch recoded variables, one expression at time is calculated until an expression evaluates true. The corresponding answer for this expression is stored as answered for the current respondent. If no expressions evaluate to true, the question will be stored as not answer for this respondent.

When recoding multi punch, the expressions for all the answers are evaluated. If the expression evaluates true, the answer is stored as answered for the current respondent. If the expression evaluates false, the answer is stored as not answer.

9.9.2 Expression syntax

The answers of other questions can be references from within an expression:

- **Single punch and open-ended questions:** ‘Question id’
- **Multi punch and grid questions:** ‘Question id’ + ‘_’ + ‘Answer precode’

The value returned by such a reference will always be a character.

To check the time of an interview, ‘interview_start’ should be used as a question name.

To operate values of other questions or other constants, some methods and operators are available:

Operator/Method	Description
==	Equal
!=	Unequal
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
&&	And
||	Or
!	Not
parseInt(string)	Converts ‘string’ to an integer.
parseFloat(string)	Converts ‘string’ to a floating-point value.
date(string)	Converts ‘string’ to a date. The string format is: ‘YYYYMMDDHHMMSS’. The string could be stopped after any of the letter changes in the format specification.
average(p1, p2, ...)	Returns the averages of the parameters. All the parameters are converted by parseFloat() before added together. Parameters that equal an empty string are not included in the average calculations at all.
.Weight(question)	Returns the corresponding weight of an answer of ‘question’.

Table: Operators and methods

When building the expression, it could be helpful to enter the condition builder. This is done by clicking a question that the recoding would depend upon.

Figure 27: Variable definition with condition builder

APPENDIX A: Paper questionnaire

This appendix shows what an automatically generated paper questionnaire might look like.

This is a demo project which shows beta output of CONFIRM's paper generator. Information about the paper questionnaire generator will be posted in the news section.

1. Which soda is your favorite?

(Check one)

- ☐ Coke
- ☐ Pepsi
- ☐ Moutain Dew
- ☐ Sprite
- ☐ Urge

2. Which sodas, if any, do you drink AT LEAST ONCE A MONTH?

(Check all that apply)

- ☐ Coke
- ☐ Pepsi
- ☐ Moutain Dew
- ☐ Sprite
- ☐ Urge

3. Please rate the following brands:

(Check one alternative per row)

	1	2	3	4	5	Don't know
Coke	☐	☐	☐	☐	☐	☐
Pepsi	☐	☐	☐	☐	☐	☐
Moutain Dew	☐	☐	☐	☐	☐	☐
Sprite	☐	☐	☐	☐	☐	☐
Urge	☐	☐	☐	☐	☐	☐

4. How would you rate your knowledge of the Web?

(Check one)

- ☐ 1 Beginner
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5 Expert

5. Please state your age.

(Check one)

- ☐ Less than 20
- ☐ 20-29
- ☐ 30-39
- ☐ 40-59
- ☐ 60 or older

6. Please specify gender.

(Check one)

- ☐ Male
- ☐ Female

7. Please state you main occupation:

(Check one)

- ☐ Permanent employment in governmental activity
- ☐ Permanent employment in private activity/business
- ☐ Self-employed
- ☐ Part time employee
- ☐ Full time student
- ☐ Part time student
- ☐ Unemployed
- ☐ Senior citizen/Living at home
- ☐ Other

APPENDIX B: CONFIRM LANGUAGE CODES

Language code	Language
0	English (Default)
2	Norwegian
3	Swedish
4	German
5	French
6	Spanish
7	Finnish
8	Danish
9	Icelandic
10	Italian

More languages will be added on demand.