

confirmit
- User Manual -
Version 6.0

September 30, 2000

The information on these pages describes **confirmit** and its features as of September 30th, 2000. New features may be introduced into **confirmit** after this date. Please check "News" for latest updates.

Examples depicted herein are fictitious.

This document is only intended for registered **confirmit** users.
No part of this document may be reproduced, transmitted or stored in any form to other than registered **confirmit** Advanced users without the express written permission of FIRM.

TABLE OF CONTENTS

TABLE OF CONTENTS	2
1 What is confirmit?	9
1.1 ABOUT THIS MANUAL.....	9
1.1.1 Interview Management.....	9
1.2 CONFIRMIT USERS.....	9
1.2.1 Advanced Users	9
1.2.2 Executive Users.....	10
1.3 ABOUT LOG-IN SESSIONS	10
1.4 SECURITY ISSUES.....	10
1.4.1 Secure Sockets Layer (SSL).....	10
1.4.2 Object Security.....	10
1.4.3 Respondent Security and Anonymity.....	10
1.4.4 Back up of Meta Data	11
1.4.5 Back up of Response Data	11
1.5 SYSTEM REQUIREMENTS.....	11
1.5.1 confirmit Users.....	11
1.5.2 Unicode Support for confirmit Server Installations.....	11
1.5.3 Respondent Requirement.....	11
1.5.4 3 rd Party Downloads	12
2 Entering confirmit – a General Overview	13
2.1 CONFIRMIT LOGIN.....	13
2.2 CONFIRMIT MENU STRUCTURE	14
3 General.....	16
3.1 MAIN PAGE.....	16
3.2 USER SETTINGS	17
3.3 PROJECT LIST.....	19
3.4 TASK MANAGEMENT	19
3.5 NEWS PAGE	20
3.6 TEMPLATES.....	21
3.6.1 Web Interview Templates	21
3.6.1.1 The Fields in the WI Template Edit Modus.....	21
3.6.1.2 Published Report Templates	22
3.6.2 MS PowerPoint Templates.....	22
3.7 HELP.....	23
3.8 EXITING CONFIRMIT	23
4 Project Overview.....	24
4.1 OBJECT PERMISSIONS	25
4.2 DUPLICATE PROJECT	26

4.3	DELETE PROJECT	26
5	Building Questionnaires	27
5.1	CREATING A NEW QUESTIONNAIRE	27
5.2	ADDING NEW OBJECTS TO THE QUESTIONNAIRE	27
5.2.1	<i>Forms</i>	27
5.2.1.1	Text.....	28
5.2.1.2	Answers	28
5.2.1.3	Scale	30
5.2.1.4	Properties.....	30
5.2.1.5	Preview	33
5.2.1.6	Tracking.....	33
5.2.1.7	Languages.....	33
5.2.2	<i>Conditions</i>	33
5.2.2.1	Description of Fields	34
5.2.2.2	Condition Builder	34
5.2.3	<i>Directives</i>	36
5.2.4	<i>Stop-objects</i>	36
5.2.5	<i>Scripts</i>	37
5.2.6	<i>Folders</i>	37
5.2.7	<i>Loops</i>	37
5.2.8	<i>Quotas</i>	39
5.2.8.1	Defining quotas and quota forms	39
5.2.8.2	Defining quota counters.....	40
5.2.8.3	Compilation of quotas	44
5.2.8.4	Changing quota limits.....	44
5.2.8.5	Updating quota counters during interviewing.....	44
5.2.8.6	Checking quotas during interviewing.....	44
5.2.8.7	Respondent info for full quotas	44
5.2.8.8	Deleting Quotas	44
5.2.9	<i>3D – Grid</i>	44
5.3	EDITING THE ROUTING	47
5.3.1	<i>Moving an Object</i>	47
5.3.2	<i>Duplicating an Object</i>	47
5.3.3	<i>Selecting Multiple Objects</i>	47
5.3.4	<i>Deleting Objects</i>	47
5.3.5	<i>Pitfalls</i>	48
5.4	SCRATCHPAD.....	48
5.5	LISTS	48
5.6	CONFIRMIT LIBRARY	48
5.6.1	<i>Templates in Library</i>	49
6	CONFIRMIT Scripts.....	51

6.1	INTRODUCTION	51
6.2	ACCESSING SURVEY VARIABLES – THE F FUNCTION	51
6.2.1	<i>f("qID")</i>	52
6.2.1.1	Single.....	52
6.2.1.2	Open Text	52
6.2.1.3	Multi	53
6.2.1.4	Multi with OpenText or Ordered Property Checked.....	53
6.2.1.5	Grid.....	54
6.2.1.6	Other, Specify.....	55
6.2.2	<i>f("qID")["precode"]</i>	56
6.2.2.1	Multi	56
6.2.2.2	Multi with Ordered Property Checked.....	57
6.2.2.3	Multi with OpenText Property Checked.....	57
6.2.2.4	Grids	57
6.2.3	<i>Loops - f("qID","iteration_precode","iteration_precode",....)</i>	57
6.3	CONDITIONS	59
6.3.1	<i>.inc()</i>	59
6.3.2	<i>.size()</i>	60
6.3.3	<i>.toNumber()</i>	60
6.4	PRECODE MASKS	60
6.4.1	<i>a("qID")</i>	60
6.4.2	<i>set(), nset(), nnset()</i>	60
6.4.3	<i>.union()</i>	60
6.4.4	<i>.isect()</i>	61
6.4.5	<i>.diff()</i>	61
6.4.6	<i>Precodes</i>	62
6.5	TEXT SUBSTITUTION/RESPONSE PIPING	62
6.6	VALIDATION CODE	64
6.6.1	<i>Default Validation Rules</i>	64
6.6.2	<i>Adding Your Own Validation Code</i>	65
6.7	SCRIPT NODES	65
7	Preparing for Data Collection.....	67
7.1	GENERATING RESPONSE DATABASES	67
7.2	GENERATING INTERVIEW FILES	67
7.2.1	<i>Web Interview Files</i>	67
7.2.2	<i>Options Controlling Appearance</i>	69
7.2.2.1	Recipient of Error Notifications	69
7.2.2.2	Include Progress Bar.....	69
7.2.2.3	Reserve a Fixed Area for Error Messages	69
7.2.2.4	Survey Design	69
7.2.3	<i>Options Affecting Interview Behavior</i>	70

7.2.3.1	Generate Back Button.....	70
7.2.3.2	Allow user to modify answers after initial submission	70
7.2.3.3	Allow user to modify answers after the interview is complete	70
7.2.3.4	Answer Required Checks	70
7.2.3.5	Exclusivity Tests	70
7.2.3.6	Other – Specify Checking.....	70
7.2.3.7	Rank Order Tests.....	70
7.2.4	<i>Options for Generating Limited Surveys.....</i>	71
7.2.4.1	Limited Survey	71
7.2.4.2	Generate Login Page	71
7.2.5	<i>Pop-Ups</i>	71
7.2.5.1	Generate Pop-up Survey	71
7.2.5.2	An Example of a Script for Pop-up Surveys.....	71
7.2.5.3	.Pop-up Script Wizard	72
7.2.6	<i>Default Language.....</i>	72
7.3	PROJECT STATUS	72
7.4	PROJECT LOG.....	73
7.5	URL SETUP	73
8	Handling Respondents in Limited Surveys.....	75
8.1	PREPARING THE RESPONDENT LIST	75
8.2	UPLOADING THE RESPONDENT LIST	75
8.3	SENDING E-MAIL	76
8.3.1	<i>Step 1: Respondent Selection</i>	77
8.3.2	<i>Mail Merging Functionality.....</i>	79
8.3.3	<i>Avoiding Unnecessary Respondent Inquiries.....</i>	79
8.3.4	<i>Step 2: Preview and Confirmation.....</i>	80
8.3.5	<i>Sending E-mails</i>	80
8.4	HANDLING RESPONDENTS IN MULTILINGUAL PROJECTS	81
8.4.1	<i>Open Multilingual Surveys.....</i>	81
8.4.2	<i>Limited Multilingual Surveys</i>	82
9	Reporting	83
9.1	INTRODUCTION	83
9.2	CHECKING A SINGLE QUESTION.....	83
9.3	BUILDING REPORTS	84
9.4	BUILDING CROSSTABS	86
9.5	REPORTING ON LOOPS	87
9.6	WEIGHT MODEL	88
9.7	TIME SERIES	90
9.8	SETTINGS	90
9.8.1	<i>General Settings.....</i>	90
9.8.2	<i>Report Elements Settings</i>	91

9.8.2.1	Excel Chart Component.....	91
9.8.2.2	PowerPoint Options.....	94
9.8.2.3	Filter Options.....	95
9.8.2.4	Answer Mask/Scale Mask	95
9.8.2.5	Crosstab Options	96
9.8.2.6	Loop Options.....	97
9.9	FILTERS	98
9.9.1	<i>Defining Filters</i>	98
9.9.2	<i>Applying Filters</i>	98
9.10	REPORT PUBLISHER	99
9.10.1	<i>Publishing a Report</i>	99
9.10.2	<i>Changing a Report after Publishing</i>	102
9.10.3	<i>Publishing Different Data from One Survey to Separate Target Groups, or Same Report in Different Languages</i>	102
9.11	MATRIX VIEWS.....	102
9.11.1	<i>Introduction</i>	102
9.11.2	<i>Configure Matrix Views</i>	103
9.11.3	<i>Using Matrix Views</i>	103
10	Recoding Data	106
10.1	EXPRESSION SYNTAX AND EVALUATION	106
10.1.1	SQL	106
10.1.2	JScript.....	107
11	Exports of Data Files	109
11.1	TEMPLATE EDITOR	109
11.2	DATA EXPORT	111
11.3	SPSS EXPORT.....	114
11.4	REPORT EXPORT	115
11.5	MS WORD EXPORT.....	115
12	APPENDIX A: confirmit Script Reference	116
12.1	ACCESSING SURVEY VARIABLES	116
12.1.1	<i>Variable Types</i>	117
12.2	USEFUL FUNCTIONS AND METHODS	118
12.2.1	<i>Basic variable objects</i>	118
12.2.1.1	.get()	118
12.2.1.2	.set()	118
12.2.1.3	.label()	118
12.2.1.4	.value()	118
12.2.1.5	.valueLabel()	118
12.2.1.6	.values().....	118
12.2.1.7	.categories().....	119
12.2.1.8	.categoryLabels().....	119

12.2.1.9	.domainValues()	119
12.2.1.10	.domainLabels()	119
12.2.1.11	Some practical use	119
12.2.2	<i>Arithmetic functions</i>	120
12.2.2.1	Sum()	120
12.2.2.2	Count()	120
12.2.2.3	Average()	120
12.2.2.4	Max()	120
12.2.2.5	Min()	120
12.2.3	<i>Conversion</i>	120
12.2.3.1	.toString()	120
12.2.3.2	.toNumber()	120
12.2.3.3	.toBoolean()	120
12.2.4	<i>Classification functions</i>	121
12.2.4.1	IsNumeric(arg)	121
12.2.4.2	IsInteger(arg)	121
12.2.4.3	IsDateFmt(arg, format)	121
12.2.4.4	IsDate(day, month, year)	122
12.2.4.5	IsEmail(arg)	122
12.2.4.6	IsNet(quadIP, quadNet, quadMask)	122
12.2.5	<i>Context information</i>	122
12.2.5.1	CurrentID()	122
12.2.5.2	CurrentPID()	122
12.2.5.3	CurrentSID()	123
12.2.5.4	CurrentLang()	123
12.2.5.5	InterviewStart()	123
12.2.5.6	InterviewEnd()	123
12.2.5.7	GetStatus()	123
12.2.5.8	SetStatus()	123
12.2.5.9	Forward()	123
12.2.6	<i>Browser information</i>	123
12.2.6.1	BrowserType()	123
12.2.6.2	BrowserVersion()	123
12.2.6.3	RequestIP()	123
12.2.7	<i>Handling sets</i>	124
12.2.7.1	a('qID')	124
12.2.7.2	set()	124
12.2.7.3	nset()	124
12.2.7.4	nnset()	124
12.2.7.5	.members()	124
12.2.7.6	.union()	124
12.2.7.7	.isect()	124

12.2.7.8	.diff()	124
12.2.7.9	.size().....	124
12.2.7.10	.inc()	124
12.2.7.11	.add()	125
12.2.7.12	.remove().....	125
12.2.7.13	Filter(qID,prefix).....	125
12.2.8	<i>General utilities</i>	125
12.2.8.1	SendMail(from, to, subject, body).....	125
12.3	FORM VALIDATION.....	126
12.3.1	<i>Adding Your Own Validation Code</i>	126
12.3.1.1	A typical Validation Code.....	126
12.3.1.2	RaiseError().....	126
12.3.1.3	Error Messages.....	126
12.3.1.4	Some default functions for Conditions in Validation Code.....	127
12.3.1.5	Template Based Error Messages	128
13	APPENDIX B: Online Coding Tool	132
13.1	WHY HAVE A CODING SYSTEM?.....	132
13.2	SPECIFICATION SUMMARY	133
13.3	CHANGES/ADDITIONS TO CONFIRMIT.....	134
13.3.1	<i>General Additions</i>	134
13.3.1.1	New Menu Item - Coding.....	134
13.3.1.2	User Settings - 2 new items	134
13.4	ONE MULTI FOR EACH OPEN-ENDED.....	135
13.5	LOCKING THE QUESTIONS FOR CODING	137
13.6	CODING WINDOW	138
13.6.1	<i>Elements of the Coding Window</i>	139
13.6.1.1	Heading	139
13.6.1.2	Searching and Sorting	139
13.6.1.3	Working Set and Coding	140
13.6.1.4	Navigation.....	141
13.6.1.5	Replace Function.....	141
13.6.1.6	Show or Hide Codes.....	141
13.7	FINAL COMMENTS	142
13.7.1	<i>Step-by-Step Guide for Online Coding:</i>	142
14	APPENDIX C: confirmit LANGUAGE CODES	144
15	APPENDIX D: RESERVED KEYWORDS	149
	INDEX	150

1 What is confirmit?

Future Information Research Management (FIRM)'s vision is to help customers to derive tangible knowledge from intangible information.

The **confirmit** software is FIRM's answer to that vision. **confirmit** is a comprehensive Web-based information retrieval and presenting software.

1.1 About this Manual

confirmit software is available either as a server installation or "log-in". They do not differ in a substantial way, but in the instances they do, "log-in" specifics are marked with a gray background like this:

Login only:

The "log-in"-version also includes CATI (Computer Assisted Telephone Interviewing) functionality. This functionality is described in a separate manual.

Throughout this manual we refer to different URLs, e.g. <http://www.yourconfirmitserver.com/confirmit/firm.asp>. You will have to substitute yourconfirmitserver.com with your actual server address, or www.confirmit.com for login users.

Please note that the terms "question" and "form" are to be interpreted in the same way.

1.1.1 Interview Management

The interview management module and more detailed information on its use will be available after the revision the functionality is undergoing at the moment.

1.2 confirmit Users

confirmit users may have either Advanced or Executive rights. Your **confirmit** administrator is entitled to define this user level.

Login only:

Contact support@confirmit.com to change user level.

1.2.1 Advanced Users

Advanced users have to acquire the most complete knowledge of the **confirmit** web application. They are typically responsible for these areas of a survey:

- Initiating a new project
- Programming questions and skip logic
- Setting up project properties (permissions, public survey/limited survey etc)
- Handling respondents in limited surveys
- Compiling/generating interview files to make the project go "live"
- Extend the library of interview layout templates
- Extend the library of questions and forms
- Create and publish reports

1.2.2 Executive Users

These users are set up with permissions to view and comment on questionnaires and response data. They may view and order generated reports. They may NOT build questionnaires or create reports themselves.

1.3 About Log-in Sessions

Whenever a **confirmit** 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 information. 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 **confirmit** 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 20 minutes, an automatic *time out* occurs. **confirmit** 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.

1.4 Security Issues

Login:

1.4.1 Secure Sockets Layer (SSL)

Regardless of what many people say, the Internet is a secure communications medium. Network traffic travels 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. As long as you trust the server on the other end of the connection, very few things can go wrong; the possibility of someone listening in on your traffic is very small indeed.

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

1.4.2 Object Security

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

A project, a question, a report—these are all *objects* inside **confirmit**'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 **confirmit**, the project is by default only visible to you, because no one else has read or write permissions. However, you may specify that other users or groups of users should have these permissions.

1.4.3 Respondent Security and Anonymity

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

confirmit 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 does not have to enter information about age or gender because this information already exists in his or her profile.

1.4.4 Back up of Meta Data

Login only:

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

For server installations, your company has to set up its own routines for backup.

1.4.5 Back up of Response Data

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

It is the responsibility of the **confirmit** user to make sure that this does not happen.

Login only:

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 exist.

For server installations, your company has to set up its own routines for backup.

1.5 System Requirements

1.5.1 confirmit Users

confirmit is designed to function correctly in Microsoft Internet Explorer 4.0 or 5.0 in Microsoft Windows™ 95/98/NT/2000. 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 **confirmit**. We therefore recommend our customers to apply the latest Microsoft Internet Explorer patch for their country. (Internet Explorer 5.0 or Internet Explorer 4.01 Service Pack 2).

1.5.2 Unicode Support for **confirmit** Server Installations

To support more than the "Western European" character set on a **confirmit** server installation *will* require that Windows 2000 is installed on the authoring and deployment servers. This is due to limitations in MS Internet Information Server 4.0. This issue has been addressed in MS IIS 5.0, which will only install under Windows 2000. We are investigating the possibilities for solving this by implementing our own character set conversion components.

Please note that system generated error messages are translated into English, Norwegian, Swedish, German, French, Spanish, Finnish, Danish, Icelandic, Italian, Dutch, Flemish, Portuguese and German. For other languages, error messages will be defaulted into the main category (eg: Venezuela Spanish into Spanish), or into English if that language is not available.

Server installation customers: See Technical Overview for information on how to add system generated messages in other languages or change those in already existing languages.

1.5.3 Respondent Requirement

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

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

1.5.4 3rd Party Downloads

The **confirmit** application does not rely heavily on 3rd party components. Currently three 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/
Reporting	Excel chart component	Microsoft	http://www.microsoft.com

Table 1: 3rd party downloads

2 Entering confirmit – a General Overview

2.1 confirmit Login

Users log in to CONFIRMIT from this URL:

<http://www.yourconfirmitserver.com/confirmit/firm.asp>

Login only: You can also log in to **confirmit** by choosing “Login” on FIRM’s home page, <http://www.confirmit.com/>.

Accessing this link will make your browser request a user name and a password, see Figure 1: Login box. Enter your personal user name and password, and you will be logged on. If the browser who tries to log on to **confirmit** does not meet the browser requirements, an error message will be displayed.

A screenshot of a Windows-style dialog box titled "Enter Network Password". The dialog has a blue title bar with a question mark icon and a close button. The main area is light gray. On the left, there is a yellow key icon. To the right of the icon, the text "Please type your user name and password." is displayed. Below this, there are two labels: "Site:" followed by the text "www.confirmit.com", and "Realm:" followed by the text "CONFIRM". Under these, there are two text input fields. The first is labeled "User Name" and the second is labeled "Password". Below the input fields, there is a checkbox with the label "Save this password in your password list". At the bottom right, there are two buttons: "OK" and "Cancel".

Figure 1: Login box

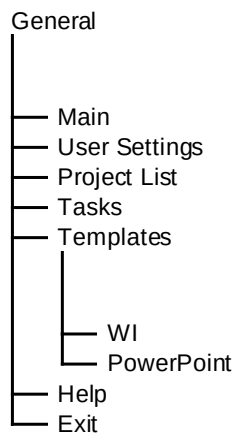
2.2 *confirmit* Menu Structure

When you have logged in to **confirmit**, you will be brought to the first page of **confirmit**. In the left-hand margin you will find a menu structure that is divided into two main sections: *General* and *Project*. These menus navigate you through both, the main and project specific, features of **confirmit**.

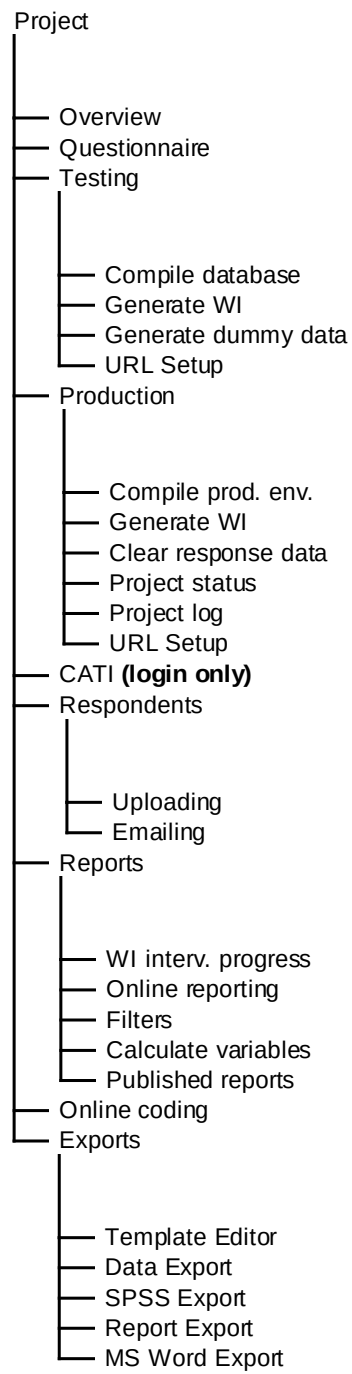
The following chapters offer a more detailed description of every choice in the menu.



Figure 2: *confirmit* menu



This and the following figure offer you an overview of all the menu links with their subsections.



3 General

This section describes the functionality under “General” in the menu.

3.1 Main Page

The first page of the application indicates some natural starting points for your session.

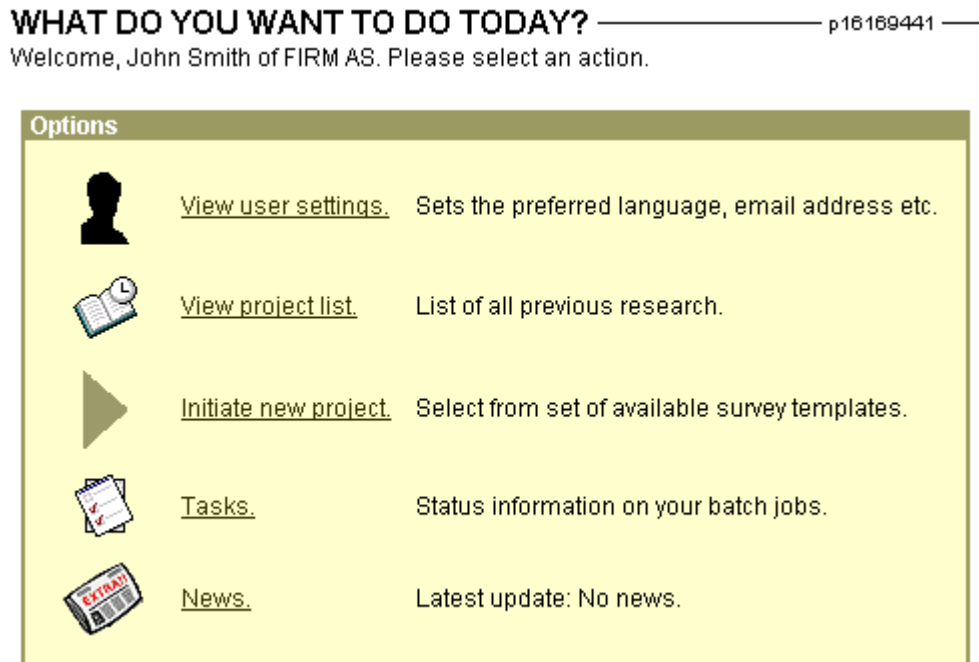


Figure 3: Welcome page

3.2 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.

USER SETTINGS ————— confirmit —

Personal settings for johns	
User name:	johns
User key:	VMQGCCLAFBWURPQORHTWWIDNG
First name:	John
Last name:	Smith
E-mail:	john.smith@firm.no
Language:	English
Display:	1024 x 768
Microsoft Office version:	Newest version
Chart component:	Excel
Coding Screen Size:	0
Working set size:	0
Answerlist font:	
Enable unicode:	☐
Requires NT or Windows2000	
Update settings	

Figure 4: User settings page

Setting	Changeable	Description
User name	-	User login name. In combination with a password, this user name will log the user on to the confirmit application.
User key	-	Unique to each user. To make yourself “visible” to confirmit users outside your organization, you will have to supply them with your user key. (This is really a security issue.)
First name, last name	X	Name of the user. Used mainly for interaction with other users.
Language	X	User’s language of preference. This is the language used when previewing questions and reports. Unicode that enables the software to handle Asian languages has been introduced into confirmit . The user will be able to choose any language when designing a questionnaire.
Display	X	Use this for optimizing your screen resolution with the confirmit software. After having made changes to your display settings, refresh the browser (F5).
Email	X	Email address of user. This address is used by the confirmit message system to send messages to the confirmit users. This is also the default email address where requested report and export files are sent.
User level	(X)	Some users are set up with permissions to change their user level between Advanced and Executive. See 1.2.
Microsoft Office Version	X	Users can choose what Office version to use for PPT/Excel exports from their user settings. By selecting 'Newest version', the newest version installed on the export server will always be used. Currently this is Office 97. Excel exports will never be made in Excel 4.0 and older, though, since these old versions do not support more than one worksheet for each workbook.
Chart Component	X	Use this to choose between First Impression or Excel components in Online Reporting. Excel component is recommended, especially if you are planning to publish the report (see 9.10 Report Publisher).
Coding Screen Size	X	Here you specify the number of answers that will be shown on the Coding Window screen. This number will also be specified on the back and forth buttons, e.g. prev 10, next 10.
Working set size	X	Here you specify the number of answers you wish to work with at a time. If you e.g. have 50 000 answers, you may wish to work in smaller batches, as e.g. 1000 at a time. Thus Online Coding Tool will process the information faster than if you worked with all 50 000 answers simultaneously.
Answer list font	X	Here you can specify the font of the entries in Answers list. Usually it will remain empty, but in cases it is necessary to use e.g. Chinese, you will have to specify a font that supports the Chinese character set, e.g. SimSun or PMingLiu.
Enable unicode	-	Check this box to be able to use Unicode supported characters. Applies only for users of NT and Windows 2000.

Table 2: User settings fields

3.3 Project List

The project list is the main project organizer. It consists of *Favorite projects*, i.e. projects you have been working on recently. 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). You only have to enter the first letters of the criteria on which the search is to be based. The system will return all matches. Example: Specifying “te” in the *name* column will return all projects starting with the letters “te”.

If you want all projects to be listed, enter a ‘*’ in the *name* column under *search projects*.

You may enter a project directly from this list by clicking the project name or the project ID.

When the list of your Favorite projects contains a considerable number of projects, you may want to remove the ones that you currently do not work on by clicking the *Rem* (=remove) link. This will not delete the projects, but move them from the Favorite list.

In case you wish to access the removed projects, you can use the *Search projects* fields. All projects the user has been granted access to will be listed according to search criteria. You may move these projects to your Favorite list by clicking the *Add* link.

PROJECT LIST ————— p0793814 ———

Project list				
Name	ID	Customer	Creator	Created
te				Last week
Favorite projects TestProject p0793814 FIRM AS Smith, John Search projects Results				
				Last week Last month Last two months Last half year last year Before last week Before last month Before last two months Before last half year Before last year
				Filter [Rem] Search

Figure 5: Filtered project list

3.4 Task Management

Task management lets you track the status of your **confirmit** batch jobs, including MS PowerPoint exports, SPSS exports etc.

When placing requests for exports (i.e. SPSS), users may select a starting time for the execution of the task: Immediately, at midnight or at a specified hour.

The task management page has five tabs:

1. Queued tasks: A list of the user's pending tasks.
2. Executing tasks: Task currently being run at the batch server.
3. Aborted tasks: Tasks that for some reason did not finish within the specified timeframe.
4. Completed tasks: Tasks that did finish within the specified timeframe. Does not guarantee that they are successful. If not, users should get an error notification in their inbox.
5. Recurring tasks: Tasks that have been scheduled to recur at specified time intervals.

6. Search tasks: Here you can search for your batch job to check their status. You can search by various parameters:
- Task ID: the ID number given to the batch job by the system.
 - User ID: the username of the confirmit user
 - Project ID: the system generated project number.
 - Status: Any, in queue, executing, completed, aborted by user, and aborted by system.
 - Task type: SPSS Export, MS Word Export, MS Office Export (PPT/Excel), Data Export, Batch Emailing.

TASK MANAGEMENT

p16169441

Global management screen for batch tasks.

Queue
Exec
Compl
Abort
Recurr
Search

Search tasks

Task ID
Project ID
Status
Task type

Task ID	Project ID	Schedule	Task type	User comment
513	p16169441	9/20/2000 3:54:39 PM	MS Office Export (PPT/Excel)	

Search

Figure 6: Task management

If you click on the *Task ID* number, you will be brought to the *Task instance properties* overview where you get information about the particular jobs Schedule, Status, percentage completed, Start time and End time, and Time usage. You can get a complete system log overview for batch job in *Log entries* section of the Task settings. By clicking on *Task Definition* button, you can get an overview of the Recurrence pattern of the particular job, and change it if necessary by clicking *Change* button on the Recurrence pattern page.

NB: **confirmit** sets a time limitation for how long a task can run before it is killed. This is done to avoid unnecessary strain on the server if there is an error with the file. However, this may also result in heavy tasks (e.g. based on long questionnaires) being killed before they are completed. Ask your **confirmit** administrator to manually adjust the time limitation, should it become necessary.

Login only:

Contact FIRM Customer Relations and Support for help

3.5 News Page

Click *News* to learn about the latest updates to **confirmit**. The page is divided into six separate categories. The most recent changes are to be found on top. The lists are searchable. Click the subject links or the red arrows for details.

[All categories](#)
[General](#)
[Q Design](#)
[Reporting](#)
[File export](#)
[Server maintenance](#)
[CATI](#)

All categories

Search

2000-01-17	Report Publisher	A new feature <i>Report Publisher</i> is now available in <i>Online Reporting</i> . ▶
2000-01-06	FAQ	A CONFIRM <i>FAQ</i> is now in place. ▶

Figure 7: News

3.6 Templates

From the *General* menu, a *Templates* link leads to two subsections, MS PowerPoint templates and Web Interview (WI) templates. **confirmit** templates are always bound to a specific organization.

3.6.1 Web Interview Templates

Web interview templates define the layout of web interview files and published reports. 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. It is important that you push the Save button after having made changes *before* you go to the Preview.

The parameters you can use in the templates have to be prefixed and suffixed by caret (^):

SNAME Survey name

HELP Help link

LOGO Your company logo

NAVCTL Navigation bar (back/fwd)

FTITLE Form title

FNAME Form name

FTEXT Form text

FCOMMENT Form instruction

INPUTS Form inputs

QNUM Question number

3.6.1.1 The Fields in the WI Template Edit Modus

Name: The name of the template

Body attributes: Controls the design of the whole page: background colors, picture (background picture), margins etc.

Error font attributes: Controls the font size/color of the error messages in a WEB survey.

Grid color: Controls the color of the gridlines in a grid question.

Horizontal label font attributes: Controls the “scale” text in a grid question.

Vertical label font attributes: Controls the “answers” text in all questions.

Page header: This field controls the layout of the top of the pages. You may e.g. insert and edit logos and the name of the survey in this field, e.g. ^LOGO^, ^SNAME^.

Page footer: Controls the bottom of the pages, e.g. company’s name and the navigation buttons, e.g. ^NAVCTL^.

Information form header: Controls the design of the upper part of an information form, both background color and text (font size, color etc.), e.g. ^FTITLE^.

Information form footer: Controls the area immediately under the information form (similar to Page footer).

Information form body: Controls the design of the information form body, background color, text of the information form, and comments, e.g. ^FTEXT^, ^FCOMMENT^.

Data form header: Controls the design of the upper part of an open, grid, single, multi question form; background color and text (font size, color, etc.), e.g. ^FTITLE^.

Data form footer: Controls the area right under the question (similar to Page footer).

Data form body: Controls the designs of the body of the grid, single, multi question form; background color, text of the question (^FTEXT^), as well as comments (^FCOMMENT^) and answer alternatives (^INPUTS^).

Progress bar font attributes: Controls the design/font size of the percent numbers under the bar.

Progress bar background color: Controls the background of the progress bar.

Progress bar image: Controls the moving progress bar that shows the progress of the survey.

3.6.1.2 Published Report Templates

The last four fields at the bottom of the WI Template Editor control the layout of Published Reports (See 9.10). No preview of these templates is available at the moment. In order to see the edited template, you will have to publish the report.

Report header: This field controls the layout of the top of the pages. You may e.g. insert and edit logos and the name of the survey in this field.

Report footer: Controls the layout of the bottom of the pages, e.g. company’s name.

Report light/dark color: The index and tables in the published reports are by default in the light and the dark green colors that are used in **confirmit**. If you wish to apply other colors to your report, you may specify HTML color codes in these fields.

There are three parameters that are used only in report templates:

RNAME Report name

RGEN Report generation time

RPER Report data period

3.6.2 MS PowerPoint Templates

Predefined MS PowerPoint templates may be uploaded to **confirmit**. All templates uploaded are visible to **confirmit** users within the same organization. A user may select from these templates when ordering MS PowerPoint presentations. Extension: .pot.

3.7 Help

In the General menu you find a Help button. When one clicks on the button, a new browser window named **confirmit** Help opens offering **confirmit** users four links with various help information:

A **Quick card** is a one-page text based help file dedicated to a certain topic/area in **confirmit**. It has short explanations of all functions available within the topic/area.

Step-By-Step includes demonstrations with screen shots and text from all the main areas in **confirmit**.

User Manual is identical to this paper edition. The document is indexed, searchable and has internal linking (e.g. by clicking on a topic in Table of Contents, you get directly to the topic).

Layout includes a comprehensive Template guide, as well as useful HTML codes.

Login customers will be directly redirected to the extranet after having clicked the Help link.

Customers with server installations may choose whether they want to have access to our extranet, or to have the last updated Help page in **confirmit** on their server.

3.8 Exiting CONFIRMIT

Log out of **confirmit** by clicking *Exit* in the left margin. To prevent other users from accessing **confirmit**, please close the browser e.g. by clicking the *Exit Internet Explorer*-button.

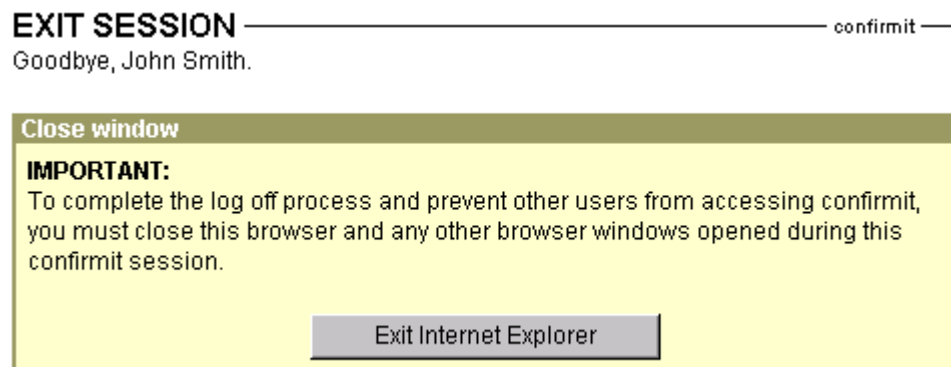


Figure 8: Exiting confirmit

4 Project Overview

A new blank project may be initiated by clicking, *Initiate new project* from the main page, and then clicking *New*. You can also choose whole questionnaire templates from this page (for more information on this subject see 5.6.1 Templates in Library). This will automatically take you to the *Project list*. Your initiated project will be positioned at the very bottom of the project list, named *New blank project*. Click on it in order to come to the *Overview* page of this project.

This is the page you are brought to when you click on existing projects in the project list.

PROJECT OVERVIEW ————— p16169441 ———

Project details

Project ID: p16169441

Project name: Nye funksjonaliteter 2

Company: FIRM AS

Description:

Languages:

Available

Afrikaans
Albanian
Arabic
Arabic (Algeria)
Arabic (Bahrain)

Selected

English
Norwegian (Bokmål)

Status:

Deployment: WI (public)

Save

[[Object permissions](#)] [[Duplicate project](#)] [[Delete project](#)]

Figure 9: Project overview page

Field	Description
Project ID	System generated unique identification number.
Project name	A title you choose for the project. Be aware that this will be the title of the browser window in a live WI survey.
Company	Your company's name.
Description	Here you may enter a more detailed description of the project.
Languages	Here you can select the working language(s) of your project. The <i>Available</i> field contains all the languages that are supported in confirmit . The <i>Selected</i> field will

	contain the working languages of your project. You highlight a language in the <i>Available</i> field and click the arrow button to add it to the <i>Selected</i> field.
Status	Project status. (See 7.3)
Deployment	Survey type (Public/Limited/Pop-Up) Web Interview. Login only: CATI

Table 3: Project overview

4.1 Object Permissions

When you click on the link *Object permissions*, you are brought to the following page:

OBJECT PERMISSIONS p16169441

Nye funksjonaliteter 2

Users in organization

Name	-	R	W	D	Administrate project	Coder of opentext questions
all	☒	☐	☐	☐	☐	☐
firm	☒	☐	☐	☐	☐	☐

Figure 10: Object Permissions

Field	Description
Name	The userid and the full name of the user (if entered in confirmit) will appear in this field.
-	The user does not have access to the project.
R	The user has only Read permission, i.e. he/she is not allowed to add new or delete existing elements in the questionnaire.
W	The user has Write permission, i.e. he/she is allowed to add questions to the questionnaire.
D	The user has Delete permission, i.e. he/she is fully allowed to work on the questionnaire including the permission to delete it.
Administrate project	The user is the administrator and owner of the project. He/she may alter the questionnaire, set the project live, check response status, etc. This is usually the person who initiates the project.
Coder of opentext questions	The user has the permission to code open-ended questions.

Table 4: Object permissions

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 **confirmit** users.

All **confirmit** users in your organization will be listed in *Object permissions*. The project administrator may set up other **confirmit** accounts with permissions to view the new project.

Login only:

Personnel at FIRM may be granted access for support purposes.

Initially, only users and groups within your organization are listed on the “object permissions” page. However, more users may be added by supplying **confirmit** 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 **confirmit** by clicking the *Add users* button at the bottom of the *Object permissions* page.

4.2 Duplicate Project

It is possible to duplicate whole projects in **confirmit** by clicking this link under *Project Overview*. The system duplicates the questionnaire structure with both skip-logics and, if used, advanced scripting. It duplicates the constructed reports as well.

You may afterwards apply changes to the questionnaire.

The data from the previous survey are not duplicated to the copy of the project.

4.3 Delete Project

You may delete projects by clicking this link under *Project Overview*.

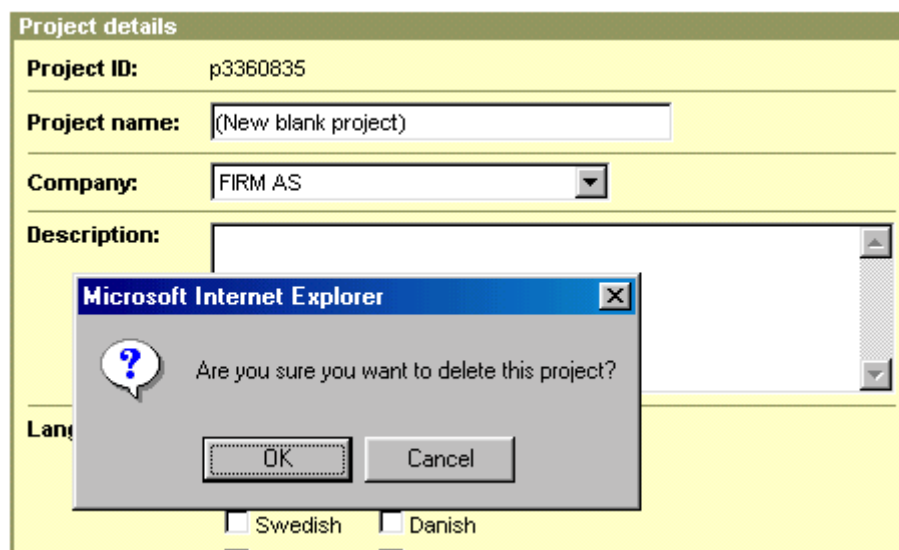


Figure 11: Deleting a project

Before actually deleting, **confirmit** will generate the message above, warning you that you are about to delete a project.

5 Building Questionnaires

5.1 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 **Figure 12**. Note the three folders, *Routing*, *Scales and lists* and *New objects*.

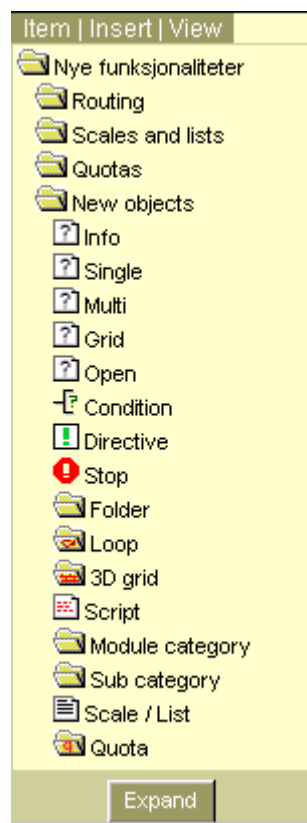


Figure 12: Questionnaire view

5.2 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 preferred destination of the object, then selecting 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 routing-objects available.

5.2.1 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 5: Standard form types

To create and edit a form, insert it into the routing, and 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.

5.2.1.1 Text

On this page, you supply the text of the form, along with other related information. 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 5.2.1.7)

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. The length of the title can be maximum 255 characters. When a title is not provided, Text will be used as a title. (In templates, you may use the primitive ^FTITLE^ where you want to insert this text. See 3.6.1.1)
Text	The form text. (In templates, you may use the primitive ^FTEXT^ where you want to insert this text. See 3.6.1.1)
Instruction	Use this field to give descriptive instructions or comments. (In templates, you may use the primitive ^FCOMMENT^ where you want to insert this text. See 3.6.1.1)
Scratchpad	See 5.4

Table 6: Text page

5.2.1.2 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 are 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.

q2 - Gender

Text **Answers** Properties Preview Tracking Languages Save

Answers

English	Precode	Weight	ColWidth	BgColor	Punch	KeepPos	Other
Male							
Female							

Add Add predefined Clear Delete rows Save

Figure 13: 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 5.2.1.7). 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. If you change the answer list while a survey is running, make sure that the precodes follow the original answers. Precodes should only contain alphanumeric characters, with no white space. The maximum length is 32 characters.
ColWidth	[Only relevant for grids] 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 value for the answer. Use this field to define the individual value/weight of each category. You need to define values/weights to be able to calculate averages for the question in Online reporting. The average will only be calculated from answers with assigned values/weights. If you e.g. have "Do not know" as a possible answer, but want the averages to be calculated only from the other answers, you may leave the value/weight of "Do not know" blank.
Punch	[Applies to Multi questions only] Gives you the ability to define one or more elements in a multi-punch list as single-punch (e.g.: "None of the above").
Background color property	This property makes it possible to have color variations inside a grid, i.e. a red to green scale. The list elements take a standard color string as input. Example: White may be specified as "white" or "#FFFFFF". A list of color codes is to be found under Help/Quick cards/HTML Colors

KeepPos	Set this property to “yes” if you want a variable keep its position when randomizing a list (e.g. “ <i>none of the above</i> ” always last in a list).
Other	Set this property to "yes" if you want an open-ended text box at the end. It is not possible to combine dropdown option (see 5.2.1.4 Properties) and Other in Grid questions. The Other option overrides the Dropdown option. If you combine Dropdown and Other in Single questions, the text box will not appear when Other is chosen. If you have checked off this option for several elements of a Grid question, these elements will automatically be marked as Not required (see 5.2.1.4 Properties), i.e. it will be sufficient to answer the other elements in the Grid.

Table 7: 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. When you start typing, the active column will be put in editing mode, and a cursor will appear. If there already is 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 move directly 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 paste rows by pressing <CTRL-V>. You can copy entire answer lists from the scratchpad and paste them into answers: copy the list in the scratchpad, go to the *Answers* in your question, click *Add*, single-click in the cell to activate it, and then paste the list.

The button *Add predefined* allows you to add a predefined list or scale to the answer list (see 5.5 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 5.4 Scratchpad.

5.2.1.3 Scale

[Only available for grids]

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

5.2.1.4 Properties

This page lets you define several aspects of the form-object. In brackets we specify which kind of questions the properties apply for.

Property	Description
Question ID	The project-unique ID of the form (questions). An alphanumeric string with no white space, starting with a letter. This is automatically generated. The data are stored in the database with variable names constructed in the following fashion: Single One column with a tag identical to question id, and with answer precodes as contents Open Similar to that of Single, but the answer text is the content. Multi A column per answer alternative. Tag is equal to: question id_precode of the answer alternative. Contents are either 0 (not selected) or 1 (selected). Grid A column for each sub question (answers). Tag is equal to: question id_precode of the answer alternative. Contents are the values answers have in Scales. Other “Other: specify” in single, multi or grid questions is represented in data files in the following way: Tag is equal to question id_precode_other. Contents will be the answer text. NB: You do not have to change the default ID of a form, even if you introduce the form between other existing forms (e.g. at a late stage of the questionnaire construction). If you anyway do change the ID, be very careful to give it a unique ID! So if you introduce a new form between forms 5 and 6, being 5 a multiple question, do not label the new one 5_2, as this is the code for the second alternative in the original question 5! These variable names will be used as column titles when survey data is exported to e.g. SPSS or Excel. 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, you should try to constrain length to 5 or 6 characters, because column names are constructed like this: question 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 12 for a description of expressions.
Field width	[All] Number of characters assigned to this question in the database and when exporting to fixed width ASCII files. In one-row text boxes, it will not be possible to enter more characters than specified in Field width. If the text box has several rows, an error message will be generated, and the answer will be truncated.
Rows/Columns	[Open, Multi] You can control the size of Open question boxes. The open-ended questions may appear as text areas or as single line input boxes. The "open text" object has got two properties: "Rows" and "Cols". Rows = Height (number of lines),

	Cols = Width (number of characters). To produce a single line input, set "Rows" to 1. In Multi questions with the OpenText option, you can specify the field width in Columns (the default is 30 characters). The Columns option does not appear until you select OpenText and after you save.
List Rows/Columns	[Single, Multi, Grids] Divides the answer list in several columns and/or rows. If you specify number of rows only, the answers will be distributed evenly across the columns and vice versa. These options have no effect on Single dropdown questions. If you have specified both options, Columns will be preferred as an option in cases where the number of elements is higher than specified. In Grids, if you have chosen Dropdown as an option, List Rows/Columns may be used to specify the number of rows and columns for the dropdowns.
Numeric	[Open, Multi with OpenText selected] If "Numeric" is selected and you click save, then input fields for "Precision" (integer $0 < p \leq 28$. Specifies the maximum total number of decimal digits that can be stored, both to the left and to the right of the decimal point), "Scale" (integer $0 \leq s \leq p$. Specifies the maximum number of decimal digits that can be stored to the right of the decimal point.), "Lower limit" (float) and "Upper Limit" (float) are available. For "Lower limit", there is a dropdown containing the choices ">" and ">=". For "Upper limit", there is a dropdown containing "<" and "<=". "Field width" is now not available. For a Multi with the OpenText selected, the same "Precision", "Scale", "Lower limit" and "Upper Limit" will be applied for all fields. Recoded variables are currently not supported.
Validation code	For unlimited flexibility in question-validation, you can write JScript-code to validate your answers. See 6 confirmit Scripts.
Coding of	[Multi with Opentext coding selected] Here you specify the Question ID of the Open-Text question that will be coded in the Multi question.

Table 8: Form properties

Flag	Description
Randomize	[Available in Single, Multi, Grid] When checked, the list of answers will be randomized.
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 JScript (see 10 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, e.g., "on-the-fly"-recoding possible.
Ordered	[Multi, Grid] Check this flag if you wish the generator to produce code that ensures that the inputs in Multi Ordered questions and Grid questions are unique and

	consecutive for each alternative in the list. In Multi questions, you can combine this choice with OpenText option. The data input boxes will be placed after the text.
OpenText	[Multi] This flag will turn a multi into a list of text-boxes, making it easy to create open-ended ratings.
Dropdown	[Single, Grid] Singles and grids are normally represented as rows and columns of radio buttons. This flag will generate dropdown-boxes instead.
Not required	[All] Normally, all questions require an answer if you have checked off the option Answer Required in WI Generator (see 7.2.1). Checking this flag will turn off this validation for the particular question.
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 <i>Show deleted nodes</i> (see 5.3 Editing the Routing), you can undelete the object by unchecking this flag.
Password	[Open] Makes it possible to collect data from the screen in a password style. The letters will appear as **** on screen. You may then want some validation of the password, or screening: q1 (<i>the password question</i>) <code>IF (f('q1') == "secret_password") THEN . . .</code> For organizations that want to do internal surveys but don't want to give out their employees' email addresses, this may be a solution. The survey password must then be distributed within the organization. Rows are always forced to 1, no matter what you specify in rows.
Opentext coding	[Multi] Check this flag to specify that this Multi question will be used for coding of an Open-Text question. Remember to check hidden, too!

Table 9: Form flags

5.2.1.5 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. Some of the features are not shown in preview yet, though, e.g. Dropdown Grid, Background Color, Size of text boxes, etc.

5.2.1.6 Tracking

The tracking functionality allows you to trace where and when a particular question has been used. This applies to questions that are saved in **confirmit** Library and reused in several projects over time. Questions are traceable as long as you do not change their Question ID. You may change titles and texts, though.

5.2.1.7 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.

5.2.2 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 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.

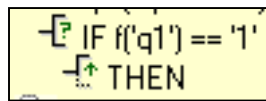


Figure 14: Conditional node

The expressions are written using the JScript-language. Expressions are described in Section 12 and to help building expressions, there is a *condition builder* (see 5.2.2.2 Condition Builder).

5.2.2.1 Description of Fields

Field	Description
Expression	A JScript-expression evaluating to <i>true</i> or <i>false</i> . Described in detail in 12 confirmit Scripts.
Branches	Select whether the condition should include an <i>ELSE</i> -branch or not.
Deleted	Used the same way as in forms (see 5.2.1.4 Properties)

Table 10: Conditional fields

5.2.2.2 Condition Builder

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

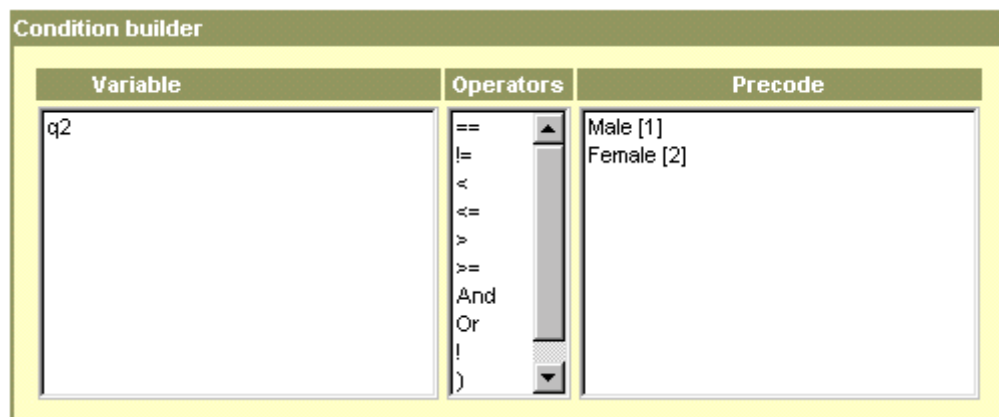


Figure 15: Condition builder

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 11: Condition builder

In combination, these lists make it easy to build complex expressions without remembering the JScript-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.

Description	Symbol
Logical NOT	!
Less than	<
Greater than	>
Less than or equal to	<=
Greater than or equal to	>=
Equality	==
Inequality	!=
Logical AND	&&
Logical OR	

Table 12: Single/grid operators

Description	Method/operator
Includes	.inc(
Union	.union(
Intersection	.isect(
Difference	.diff(
Logical AND	&&
Logical OR	
Logical NOT	!
End parenthesis	)
Comma	,

Table 13: Multi operators/methods

Remember that each part in a condition has to be a complete statement.

Example: You have an age question with 4 answer alternatives. You want to include a condition that selects only respondents who have answered '1' or '2'.

This is *not* the correct condition:

IF f('age') == '1' || '2' THEN...

Here '2' is *not* a statement!

This is the *correct* syntax:

IF f('age') == '1' || f('age') == '2' THEN...

See 6 **confirmit** Scripts for a more thorough description of Jscript syntax in conditions.

To check if your expressions are correct there is a Check Expression button. The Check Expression might however sometimes give you an error message, even if your expression is correct:

Error in expression

Object doesn't support this property or method

confirmit allows all valid Jscript expressions whereas the Check Expression is built only to accept the most basic expressions. You will also get a similar error message when you compile your project (see 7.1.).

When you test your project the condition will work properly.

5.2.3 Directives

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

At the moment, two directives exist :

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, precode mask 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 14: Directives

5.2.4 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 web interview should be considered a complete interview. Status is set to complete and the end time of the interview is registered. Normally, online reporting will only show statistics for complete interviews.
Screened	Use this status-code to mark the interview as incomplete because of screening. Status is set to "screened" and end time is set.
Quota full	The interview is stopped for a certain segment of respondents due to a full quota.

Table 15: Stop node settings

5.2.5 Scripts

For very advanced routing, on-the-fly recoding, or other programming efforts, you can insert script-objects that will run your JScript-code when executed. When using script-objects, care must be taken to create codes that will run without errors. JScript 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. (See 6 **confirmit** Scripts.)

5.2.6 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 16: 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 (see 5.6 CONFIRMIT Library).

5.2.7 Loops

[Login only: CATI has different loop construct.]

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. You can also use nested loops (loops in loops).

Loop ID

Field width

Precode mask

Randomize
☐

Deleted
☐

Loop members

English	Precode	KeepPos	Active
cars			

Add
Add predefined
Clear
Delete rows
Save

CATI Specific:
Add Column
Del Column

Figure 16: 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 the same way as you create an answer list in a form (see 5.2.1.2 Answers). The different parts of the loop-details view are:

- **Loop ID:** A unique ID of the loop within the project.
- **Field width:** Number of characters assigned to the iteration precode in the database and when exporting to fixed width ASCII files.
- **Precode mask:** If you want to dynamically exclude one or more of the iterations in the list, use this field to create a JScript-expression resulting in a set of precodes to include. See APPENDIX A: **confirmit** Scripts for a description of expressions.
- **Randomized:** When checked, the list of iterations will be randomized.
- **Deleted:** When you delete a loop from the routing, it is not permanently deleted, but the “deleted”-flag is set. If you turn on *Show deleted nodes* (see 5.3

Editing the Routing), you can undelete the object by unchecking this flag.

- **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 5.2.1.7 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 white space. The maximum length is 32 characters.
KeepPos	Set this property to “yes” if you want an item to keep its position when randomizing the list of iterations.
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 17: Standard loop columns

- **CATI Specific (login only)**

5.2.8 Quotas

Within a survey, quotas are used to predefine the number of responses you would like for different sub samples (target groups), or for the overall survey. When a quota is reached, the survey can automatically be closed for respondents belonging to that sub sample by using conditional expressions, while it remains open for other respondents belonging to other quotas that have not been filled up.

confirmit allows for full quota control based on both demographical or attitudinal questions. Quota control can also be achieved on compound quotas. **confirmit** may also inform you by mail warning about quotas being completed.

Example:

- You have a pop up survey running, and you know that mostly young people visit the site (your estimate: 5% 40 years or older). However, you are interested in opinions from 40+ as well, and you would like to have 100 responses from that sub sample. Without quota management you would have to “wait” for 2000 respondents ($2000 \cdot 5\% = 100$). But if you only need e.g. 200 responses from the young ones, you can avoid receiving “too many” responses by setting quotas, closing the survey for the young ones after 200, and just letting the others in until their quota is filled up as well.

5.2.8.1 Defining quotas and quota forms

The questionnaire author from within the Questionnaire tree defines quotas. Quotas are created by following these steps:

1. Drop a *Quota* object from the *New objects* folder into the *Quotas* folder.

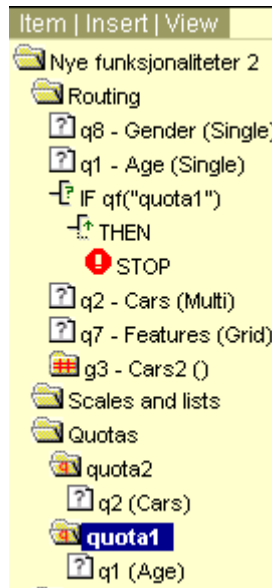


Figure 17: Adding quotas to a survey

2. Drop the questions that are relevant for the quota into the new quota folder. Only single punch, multi punch and grid questions can be used for quotas. **Forms from within loops cannot be used for quotas. It is not customary to use multi punch and grid questions as quotas.**

5.2.8.2 Defining quota counters

Once a quota is defined in the questionnaire tree, the actual quota limits with corresponding quota counters have to be defined. **Remember that you have to recompile the database before you can edit the quota details.**

Initially a quota contains no counters. The counters are defined by adding a combination of precodes from zero or more of the forms within the quota. You double-click on the quota folder in order to edit quota counters.

If you click the *Add all* button, all answer alternatives will be added for quota editing. E.g. in Figure 18: Quota details for a single punch question, all age groups would be added for quota editing.

QUOTA DETAILS p16169441

Quota 'quota1'

Quota name:

Quota full email:

Age	Quota limit	Quota count	Email
Save Update counters			
[Any] 15-24 25-34 35-44 45-54 55-64 65 and older		Add	

Figure 18: Quota details for a single punch question

If you wish to define a quota limit only on one or several of the variables, you highlight it and click the *Add* button.

QUOTA DETAILS p16169441

Quota 'quota1'

Quota name:

Quota full email:

Age	Quota limit	Quota count	Email
25-34	100	3 ☒	[Del quota]
Save Update counters			
[Any] 15-24 25-34 35-44 45-54 55-64 65 and older		Add	
Add all			

Figure 19: Defining quota limit for one variable

Field	Description
Quota name	Allows to name your quota
Quota full email	The email address you wish to receive a message to once the quota is full.
Quota limit	Here you specify the quota limit.

5.2.8.3 Compilation of quotas

Once quotas with all quota limits are defined, the quotas are compiled to the SQL database (interview environment) during the compilation of the survey. During this compilation all quota counters are calculated based on completed interviews if the quota counter has not been compiled before.

Once a quota counter is compiled to the interview environment, the interview engine only updates the quota counter when new interviews are completed.

If you add a quota after a survey has started running, you will have to press Update counters button in order to register the respondents qualifying for the quota sample, but who have answered the survey before the quota was added.

5.2.8.4 Changing quota limits

Once the quota is compiled, any changes to the quota limits will be updated directly to the interview environment. It is not necessary to recompile the survey in order to update the quota counters. It is enough to click on the “Update counters” button. If some interviews are removed from the database manually, it may be necessary to recalculate all the quotas. Selecting Update counters from the quota details screen can do this.

5.2.8.5 Updating quota counters during interviewing

When an interview is completed the first time, quota counters are incremented by 1. Only counters satisfying the answers are incremented. If a respondent re-enters an already completed interview, the quota counter is not modified.

In the single punch question quota example above, the counter for respondents in the age group 25 - 34 will be incremented if a respondent in this age group completes an interview.

5.2.8.6 Checking quotas during interviewing

All quota-checking needs to be programmed into the interview as indicated in the Questionnaire tree above, see Figure 17: Adding quotas to a survey. The function `qf(“quotaname”)` returns true if the specified quota is full.

5.2.8.7 Respondent info for full quotas

The users will receive a message telling that there are enough responses of a particular category.

As shown in Figure 17: Adding quotas to a survey, you will have to either stop the survey for the respondents qualifying for an already full quota, or reroute them to a different group of questions, all depending on your survey.

If you are terminating the survey by adding the STOP node to the THEN branch, the respondents will receive the standard **confirmit** message: Thank you very much for your time! If you wish to present the respondents with a customized message, add an Info object before the STOP node and enter your end text, e.g. “Thank you for your cooperation, but we have gathered enough responses from the target group you belong to.....”.

5.2.8.8 Deleting Quotas

In order to delete a quota from the database, you highlight the quota, press *Delete*, and then recompile the database.

5.2.9 3D – Grid

A 3D grid is a new questionnaire folder object. This object gives you the opportunity to combine various question types into one object with grid layout thus enabling a more compact rendering. The 3D grid allows all the question types to share the same list of answers.

The 3D grid object can contain the following question types:

- Single punch question
- Multi punch question
- Grid question

To add a 3D grid object to your routing, you drag a 3D grid object from the *New objects* folder:

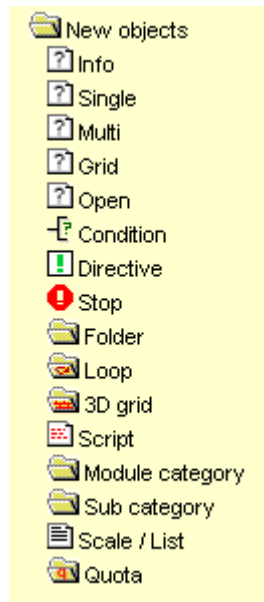


Figure 23: 3D grid object in the New objects folder:

and drop it in your questionnaire routing, e.g. Cars2 ().

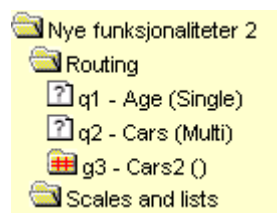


Figure 24: 3D grid in the questionnaire routing

Once you have added a 3D grid object to your routing, you can edit it by double-clicking it. In *Answers* you enter the answer list that will be shared by all the questions that will be added to the 3D grid. In *Text* you may enter the question text that also will be common for all the elements in the 3D grid. The menu and properties are the same as for the other types of questions, and are explained in detail in Chapter 5.2.1 Forms.

Now you can drag questions from the *New objects* folder and drop them into the 3D grid. Remember that **if you want the object to end up inside the folder, hold the <ALT>-key down before dropping. The object will appear as the first object in the folder.** The next object you can drop on the first object in the folder.

You edit the questions in the 3D grid by double-clicking them. You will notice that there is no *Answers* option for Single punch, Multi punch and Grid questions in the menu. The text you wish to appear above each of the questions in the 3D grid, e.g. *Like, Own, Which best* (see Figure 27: A 3D grid object in a survey), has to be entered in the Text field of these questions.

q4 - Liking					
Text	Properties	Preview	Tracking	Languages	Save

Properties	
Question type	Multi

Figure 25: Multi punch question menu in a 3D grid

However you can add *Scale* to the Grid element in the 3D grid.

q6 - Rating						
Text	Scale	Properties	Preview	Tracking	Languages	Save

Properties	
Question type	Grid

Figure 26: Grid question menu in a 3D grid

You will not be able to see the *Preview* of the whole 3D grid object. For this purpose you will have to Generate WI in the Test mode. You can preview all the elements in the grid alone, though.

The following picture shows how a 3D grid consisting of a Multi punch question, a Single punch question, and a Grid question with a dropdown property chosen, will look like in an interview.

Cars2			
Answer the following questions about cars:			
	Like:	Own:	Which best:
Vovlo	☐	☐	Please select your answer ▼
BMW	☐	☐	Please select your answer ▼
Audi	☐	☐	Please select your answer ▼
Mercedes	☐	☐	Please select your answer ▼
Saab	☐	☐	Please select your answer ▼
Ferrari	☐	☐	Please select your answer ▼

Figure 27: A 3D grid object in a survey

5.3 Editing the Routing

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

5.3.1 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 an allowed destination. An example of an un-allowed 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.**

NB: If you should wish to change default form IDs, please read carefully the Question ID section in Table 8, 5.2.1.4 Properties.

5.3.2 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.

5.3.3 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 5.3.1 Moving an Object.

5.3.4 Deleting Objects

To delete one or more objects, select them, and press . 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 5.2.1.4 Properties), 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.

5.3.5 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.

5.4 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.

5.5 Lists

Quite often, one or more scales, or one or more brand-lists, product-lists, etc. are reused several times in a questionnaire. **confirmit** lets you predefine such lists, and reuse them in forms (see 5.2.1.2 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 5.2.1.2 Answers for description of usage.
Scratchpad	See 5.4 Scratchpad.

Table 19: List fields

5.6 CONFIRMIT Library

The CONFIRMIT 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.

- 1) Open the Questionnaire you are working on.
- 2) To enter the Library you go to the upper menu *View > Library*. Another frame beneath the questionnaire will open.
- 3) You go to the new objects folder in the questionnaire tree and drag a folder named *Module category* into the folder named *Modules* in the Library. You can double-click on it to give it a name.
- 4) You mark your module category, then you go to upper menu *Insert > Module* and a tree structure alike node will appear in your folder. You can have as many of them there as you wish.
- 5) When you double-click on the module, a third frame, looking exactly like the Questionnaire frame, will open. You double-click on the first folder, and the following window will open:

Module properties

Type: Template

Image: Module Template

English

Name: Test Age

Description:

Save

Figure 28: Module properties

Field	Description
Type	In this field you can choose whether you wish the question/questions/questionnaire in the module to be saved only as a module in the library, or also as a template in the <i>Main menu > Initiate a new project</i> . See 5.6.1 Templates in Library.
Image	In this field you can refer to an image that will appear as an icon on the <i>Initiate a new project</i> page next to the title of the template. You simply enter the URL of the image.
Name	In this field you enter the name of the folder, which will also be the title of the module.
Description	You can enter a more detailed description of the module.

Table 20: Module properties

In those modules you can simply drag and drop the questions or whole questionnaires you plan to reuse. You just drag the questions into the Routing folder.

The same procedure goes for Forms folder.

When you start working on a new project and wish to use some of the questions from the library, you simply open the library and the module where the particular question or questionnaire is stored, and by the drag and drop method move them upwards to the questionnaire. The items are copied in this process, so a copy will always remain in the library.

5.6.1 Templates in Library

When you work on a module you wish to store in the **confirmit** Library, you are given the option to store it either as a module or as a template. Ordinary modules are accessible only through the **confirmit** Library. If you store a module as a template, it will be possible to access it through the Library, and as a link on the *Initiate New Project* page, as shown below. When you click a template on the Initiate New Project page, you will be brought straight to the Questionnaire builder containing the questions from the template you chose.

INITIATE NEW PROJECT

p0796346

Please select a project type.

Initiate new project

<u>New</u>	Creates a blank project.
<u>Template Test</u>	(Template)
 <u>Cancel</u>	Back to welcome page.

Figure 29: Templates in Initiate New Project

6 CONFIRMIT Scripts

6.1 Introduction

Most questionnaires contain some amount of script code. Scripts are used:

- In conditionals for controlling the flow through the questionnaire,
- For filtering the lists displayed in questions and the iterations in loops based on previous answers (precode masks)
- In form elements for text substitution (piping),
- For custom validation of user input and
- In general-purpose code contained in script nodes.

confirmit uses Microsoft's JScript¹ 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 let you manipulate survey variables.

JScript is an interpreted programming language, which means that the code will only be checked at run-time. So when you use scripts you should always test your questionnaires thoroughly before going "live".

This chapter describes briefly how to refer to and manipulate survey variables.

NB! The functions used here are not described in detail. They are described in the context they are used most often, (e.g. `union()` in the precode masks section), but that does not mean that they are in any way restricted to being used in just that context.

For a more thorough reference manual to scripting in **confirmit**, see Appendix A where the functions are described in detail.

6.2 Accessing Survey Variables – the *f* Function

The *f* function is used to access survey variable values, i.e. the answers given to a question. It will return values or sets depending of the type of variable it is used on. It will return *null* if the question has not been asked (e.g. because of skipping logic).

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.

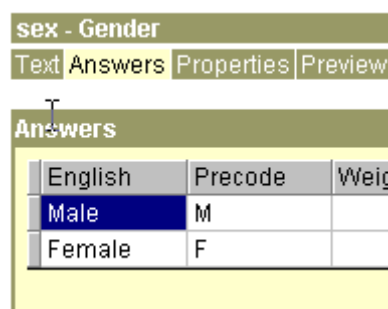
6.2.1 f("qID")

Its basic syntax is `f("qID")`, where qID is the question ID specified in Properties (see 5.2.1.4 Properties). The values returned are different for different question types, but they are always strings or sets of strings:

Question type	Values returned
Single	The precode of the answer.
Open	A string with the text answered.
Multi	A set with the precodes of the answers selected.
Multi with Ordered or OpenText checked (see 5.2.1.4 Properties)	A set with the precodes of the elements in the answer list that are answered.
Grid	A set with the precodes of the elements in the answer list that are answered.
"Other" from answer lists	A string with the text answered.

Table 21: Values returned from `f("qID")` for different question types

6.2.1.1 Single



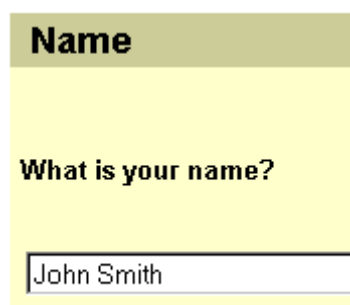
The screenshot shows a survey question titled "sex - Gender". Below the title are four tabs: "Text", "Answers", "Properties", and "Preview". The "Answers" tab is active, displaying a table of possible answers. The table has three columns: "English", "Precode", and "Weight". The "English" column lists "Male" and "Female". The "Precode" column lists "M" and "F". The "Weight" column is empty. The "Male" row is highlighted in blue.

English	Precode	Weight
Male	M	
Female	F	

Figure 30: Answer list of single question "sex"

If the respondent answers Male, `f("sex")` will return the value *M*.

6.2.1.2 Open Text



The screenshot shows a survey question titled "Name". Below the title is a text input field with the question "What is your name?". The input field contains the text "John Smith".

Figure 31: Open Text question "name"

`f("name")` will return *John Smith*

6.2.1.3 Multi

TVchannels - TV channels

Text **Answers** Properties Preview Tracking Languages Save

Answers

English	Precode	Weight	ColWidth	BgColor	Punch
ABC	1				
CBS	2				
CNN	3				
FOX	4				
NBC	5				
None of them	99				Single

Add Add predefined Clear Delete rows Save

Figure 32: Answer list of multi question "TVchannels"

If the respondent answers ABC, CNN and NBC, `f("TVchannels")` will return a set containing 1,3,5.

This answer list (and its precodes) will be used in the next examples as well.

6.2.1.4 Multi with OpenText or Ordered Property Checked

Rank

Please rank the channels you watch on a regular basis, 1 being the one you like best.

2 ABC

3 CNN

1 NBC

Figure 33: Multi question "rank"

For an Ordered multi question the respondent has to provide consecutive answers in the range from 1 to the number of items in the list (3). `f("rank")` will return the precodes of the items answered in the list, 1,3,5. When used in a script node you have to use `f("rank").get()` (see 12 APPENDIX A: confirmit Script Reference for more details).

Favorite program

What is your favorite program on these channels?

ABC	NYPD Blue
CNN	
NBC	Friends

Figure 34: Multi question "program"

A multi question with the OpenText property checked will present the answer list. If "Not required" is checked as well, the respondent will not have to answer the question. If the question is answered like this, $f("program")$ will return a set containing 1,5.

(Notice that the answer lists of these two questions as well as the grid in next question are filtered based on the answers given on the multi question *TVchannels*. More on filtering with "precode masks" in 6.4)

6.2.1.5 Grid

Hours watching



How many hours per week do you watch these channels?

	Less than 1 hour	1-2 hours	3-5 hours	6-10 hours	More than 10 hours
ABC	☐	☒	☐	☐	☐
CNN	☒	☐	☐	☐	☐
NBC	☐	☐	☒	☐	☐

Figure 35: Grid "hours"

$f("hours")$ will return a set of precodes from the answer list that has an answer, here it would be 1,3,5.

6.2.1.6 Other, Specify

favorite - Favorite TVchannel							
Text	Answers	Properties	Preview	Tracking	Languages	Save	
Answers							
English	Precode	Weight	ColWidth	BgColor	Punch	KeepPos	Other
ABC	1						
CBS	2						
CNN	3						
FOX	4						
NBC	5						
Other, specify	6						Yes
Don't know	99						

Figure 36: Answer list of single question "favorite"

Favorite TVchannel

What is your favorite TV-channel?

☐ ABC
☐ CBS
☐ CNN
☐ FOX
☐ NBC
☒ Other, specify
☐ Don't know

Figure 37: Single question "favorite"

An Other text box will have "question id"+_+"precode"+_other as variable name, here it would be `favorite_6_other`. `f("favorite_6_other")` will return the string `MTV`.

This syntax is useful when you want to include the value entered in the Other text box in answer alternatives in later questions or in loops.

Loop ID

Field width

Precode mask

Randomize
☐

Deleted
☐

Loop members

English	Precode	KeepPos	Active
ABC	1		
CBS	2		
CNN	3		
FOX	4		
NBC	5		
^f(favorite_6_other)^	6		

Figure 38: Other, specify for in loops

In this case *MTV* will appear in the loop, not *Other, specify*.

6.2.2 `f("qID")["precodes"]`

Multi and grids group one or more variables in the same form. The answers are stored in consecutive columns in the database (see 5.2.1.4 Properties for an explanation of how they are stored). To get the answer of an item of a Multi or a Grid question, use the precodes of the item you want as identifier:
`f("qID")["precodes"]`

Question type	Values returned
Multi	1 if the item with this <i>precodes</i> is checked off, 0 if it is not.
Multi with Ordered checked	A string with the value (rank) given to the item with this <i>precodes</i> .
Multi with OpenText checked	A string with the text answered for the item with this <i>precodes</i>
Grid	The precodes (from scale) of the answer to the item with this <i>precodes</i> .

Table 22: Values returned from `f("qID")["precodes"]` for different question types

6.2.2.1 Multi

If we use the same example as in 6.2.1.3, the multi question *TVchannels* where ABC, CNN and NBC are selected, `f("TVchannels")["1"]` will return 1 (since ABC is checked),
`f("TVchannels")["2"]` will return 0 (since CBS is not selected).

6.2.2.2 Multi with Ordered Property Checked

Using the same example as in 6.2.1.4, the multi question *rank* where ABC was ranked as number 2, CNN as 3 and NBC as number 1, `f("rank")["1"]` will return 2 (since ABC was ranked as number 2), `f("rank")["3"]` will return 3 (CNN) and `f("rank")["5"]` will return 1 (NBC).

6.2.2.3 Multi with OpenText Property Checked

Using the multi question *program* from 6.2.1.4, `f("program")["1"]` will return *NYPD Blue*, `f("program")["3"]` will return an empty string (since the respondent did not answer anything for CNN) and `f("program")["5"]` will return *Friends*.

6.2.2.4 Grids

See the example in 6.2.1.5, the grid *hours*. Using the answers from that example, `f("hours")["1"]` will return 2, `f("hours")["3"]` will return 1 and `f("hours")["5"]` will return 3 provided that the scale of the grid is defined as in Figure 39.

hours - Hours watching			
Text	Answers	Scale	Properties
Answers			
English	Precode	Weight	
Less than 1 hour	1		
1-2 hours	2		
3-5 hours	3		
6-10 hours	4		
More than 10 hours	5		
Add Add predefined Clear Delete row			

Figure 39: Scale of grid "hours"

6.2.3 Loops - `f("qID","iteration_precode","iteration_precode",.....)`

You refer to questions inside loops as described above, but you may want to be able to refer to the value in a specific iteration within a loop.

Say we have a loop that iterates through the list of channels in the question *TVchannels* from 6.2.1.3. For each of them, a multi question *types* is asked and the respondent checks off different types of programs he or she likes to watch on the channel. Then there is a loop inside the other loop iterating through the types of programs the respondent has checked off, and inside that loop the respondent is asked to rate the channel's programs of that type, and if he/she gives it a score 4 or 5 he/she is asked to give the reasons for giving that score.

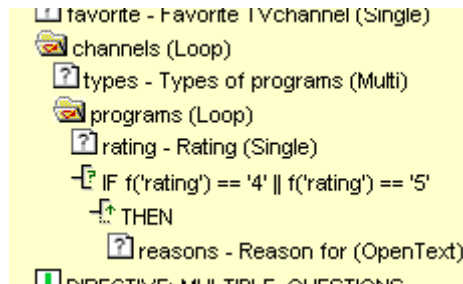


Figure 40: Nested loops

As you see from the IF-condition, within the innermost loop, *rating* may be referenced as a normal single question, using `f('rating')`. This expression will give the value of *rating* within the current iteration. But if you wanted the value of *rating* in another iteration, the "precode" of that iteration would have to be specified. `f("rating", "1")` would be the value of *rating* of "Sit-coms" (see Figure 41).

Loop ID

Field width

Precode mask

Randomize ☐

Deleted ☐

Loop members

English	Precode	KeepPos	Active
Sit-coms	1		Yes
Soaps	2		Yes
Talkshows	3		Yes
Childrens entertainer	4		Yes
Education	5		Yes
Sport	6		Yes
News	7		Yes
Movies	8		Yes
Nature	9		Yes
Documentary	10		Yes

Figure 41: The inner loop, "programs"

This expression may be used within both loops. For the outer loop the current value for "channels" will be used. But you may specify the iteration of the outer loop as well (Outside the loops you would have too!). `f("rating", "1", "5")` would give the rating of NBC's "Sit-coms" (see Figure 42). Notice the order: The innermost loop first. Outside the loops an expression like `f("rating")` would not make much sense.

Loop ID

Field width

Precode mask

Randomize
☐

Deleted
☐

Loop members

English	Precode	KeepPos	Active
ABC	1		Yes
CBS	2		Yes
CNN	3		Yes
FOX	4		Yes
NBC	5		Yes

Figure 42: The outer loop, "channels"

Inside the loops you may refer to the current iteration with the syntax `f("loopID")`, e.g. `f("channels")` or `f("programs")`. The value returned will be the precode of the current iteration.

6.3 Conditions

When making conditions, the following standard Jscript logical operators are useful:

Description	Symbol
Logical NOT	!
Less than	<
Greater than	>
Less than or equal to	<=
Greater than or equal to	>=
Equality	==
Inequality	!=
Logical AND	&&
Logical OR	

Table 23: Jscript logical operators

6.3.1 .inc()

As described in 6.2.1, `f("qID")` of multi question returns a set of the precodes answered to question `qID`. If you want to test for a particular value, use `.inc` (includes). `l.inc(r)` returns TRUE if `r` is a member of the set `l`. E.g. `f("qID").inc("precode")` will return TRUE if "precode" is one of the answers to `qID`.

6.3.2 .size()

If you need a test on how many answers were selected on a multi question, you may use .size().

`l.size()` will return the number of items in the set `l`. `f("qID").size() > 1` will return TRUE if `qID` has more than one answer.

6.3.3 .toNumber()

The .toNumber() method converts the variable to a number.

Example: The `f` function returns strings or sets of strings. If you e.g. had a single question with an answer consisting of 11 elements with the precodes "1", "2", ..., "11" the expression

`f("qID") <= 4` would evaluate to TRUE for the following answers: "1", "10", "11", "2", "3" and "4".

.toNumber() converts the variable to a number. So `f("qID").toNumber() <= 4` would evaluate to TRUE for the following answers: "1", "2", "3" and "4".

6.4 Precode Masks

Precode masks are used to filter answer lists of grids, single and multi questions and to select a set of iterations for a loop. In the precode mask field in Properties of a question (see 5.2.1.4) or in the loop construct (see 5.2.7) is a Jscript expression that evaluates to a set of precodes. The answer list (or loop member list) will be filtered based on the set of precodes in the Precode mask field.

We have already seen how the `f` function evaluates to a set of precodes when used on multi questions. These are other functions that are useful when constructing sets of precodes:

6.4.1 a("qID")

`a("qID")` returns the complete set of precodes defined in the answer list belonging to `qID`, no matter if they are answered or not.

6.4.2 set(), nset(), nnset()

Sometimes you need to create sets yourself. These three functions simplify this.

`set("precode", "precode", ...)` returns a set consisting of the precodes you specify.

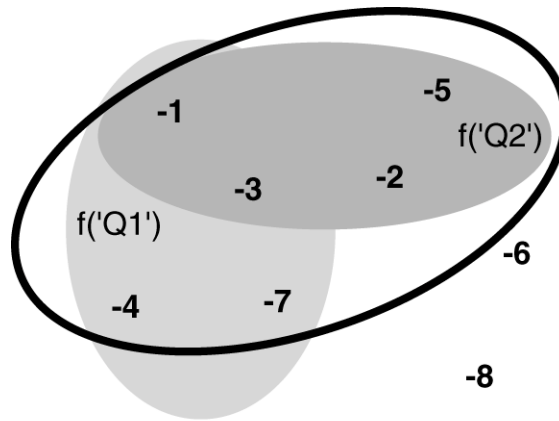
`nset(n)` returns a set populated with the members 1, 2, 3,...,n.

`nnset(m, n)` returns a set with values from `m` to `n`: `m`, `m+1`, ..., `n-1`, `n`.

Example: You use a general predefined list as your answer alternatives in several questions, but in a particular question you only want to return answer alternatives 8,9,10,11 and 12. Then all you have to do is to put `nnset(8,12)` in the Precode mask property of that question.

6.4.3 .union()

`l.union(r)` returns a new set that is the union of `l` and `r`, that is a set consisting of all precodes in either `l` or `r`. If you want to add precodes to a set in a precode mask, then use .union()

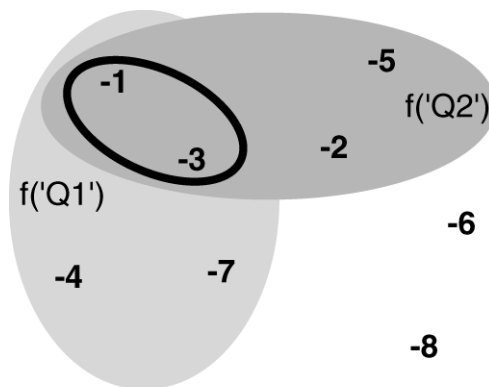


`f('Q1').union(f('Q2'))`

Example: `f('qid').union(set('99'))` Here you will always include the answer alternative with precode '99'

6.4.4 `.isect()`

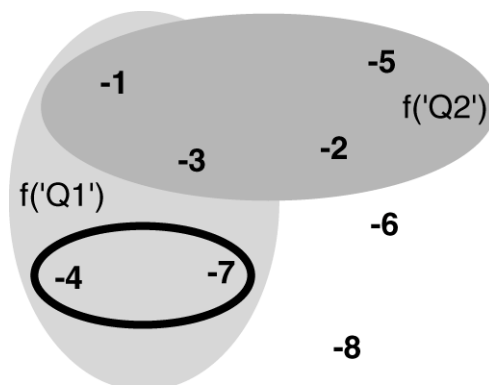
`l.isect(r)` returns a new set that is the intersection of `l` and `r`, that is a set consisting of all precodes both in `l` and `r`.



`f('Q1').isect(f('Q2'))`

6.4.5 `.diff()`

`l.diff(r)` returns a new set that is the difference between `l` and `r`, that is a set consisting of all precodes in `l` and not in `r`. If you want to remove precodes from a set in a precode mask, then use `.diff()`



`f('Q1').diff(f('Q2'))`

Example: `f('qID').diff(set('1'))` Here '1' is never included.

6.4.6 Precodes

Remember that the precode mask gives a set of precodes that are used to filter the answer list/iterations. Make sure that you use exactly the same precodes on the items of the answer lists in the different answer lists, or else the masks will be wrong. Tip: Use Predefined scales/lists (see 5.2) as much as possible when using precode masks, because then you are sure that the precodes are exactly the same.

6.5 Text Substitution/Response Piping

In order to retrieve text or values from a question and insert it in the question wording of another question, you may use caret (^) in front of and after a Jscript expression.

Example:

In the *TVchannels* question from 6.2.1.3 `^f("name")^` is inserted in the text field. This is referring to the *name* question of 6.2.1.2.

The screenshot shows a web-based question editor for a question titled "TVchannels - TV channels". The interface has a top navigation bar with tabs: "Text" (selected), "Answers", "Properties", "Preview", "Tracking", "Languages", and "Save". Below this is a section for the "English" language. It contains three main input areas: "Title" with the text "TV channels", "Text" with the content "Hello, ^f('name')^.
 Which TV channels do you watch on a regular basis?", and "Instruction" which is currently empty. Each input area has a vertical scrollbar on its right side.

Figure 43: TVchannels

When displayed to the respondent, the question will look like this (of course provided that the respondent answered "John Smith" to the *name* question):

TV channels

Hello, John Smith.
Which TV channels do you watch on a regular basis?

- ☒ ABC
- ☐ CBS
- ☒ CNN
- ☐ FOX
- ☒ NBC
- ☐ None of them

Figure 44: "TVchannels" question with text substitution.

Question type	Returned from $\wedge f("qID")\wedge$
Open text	The text answered in the text box.
Single	The text from the answer list corresponding to the precode returned from $f("qID")$ in the current language.
Multi	The text(s) from the answer list corresponding to the precode(s) returned from $f("qID")$ in the current language. If more than one, the answer will be separated by comma and with "and" (in the current language) between the last two items.
Other property of answer lists	The text answered in the text box. (The question ID is <code>qID_precode_other</code> (see 5.2.1.4))
Loop	The text of the loop member corresponding to the iteration with precode <code>qID</code> in the current language.

Table 24: Texts returned from $\wedge f("qID")\wedge$ for different question types

Question type	Returned from $\wedge f("qID")["precode"]\wedge$
Multi with OpenText or Ordered property	The text answered in the text box.
Grid	The answer from the scale corresponding to the precode returned from $f("qID")["precode"]$ in the current language.

Table 25: Texts returned from $\wedge f("qID")["precode"]\wedge$ for different question types.

The same rules will apply to loop questions, but there you may need to specify iteration(s) as well (see 6.2.3).

6.6 Validation Code

When you add a form to a questionnaire you actually add four stages of operations to the interview program:

1. **User interaction:** Display one or more forms to the person entering responses and wait for input.
2. **Check size:** Verify that open-ended answers comply with the field width. If not, truncate answers and raise error flag. Note that for performance reasons, no such checks are added for coded variables. The system assumes that the survey designer makes sure that the code lengths do not exceed the field width.
3. **Store:** Store the value of entered responses.
4. **Validate:** Check that answers stored comply with the specified data validation rules. If an error has been raised then repeat from 1.

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.

6.6.1 Default Validation Rules

The system provides these types of built-in error checking:

1. Required answer testing.
2. Exclusivity checks.
3. Other-Specify verification.
4. Rank order testing.
5. Answer size tests for fixed-width fields.

If you do not like the default handling of 1–4, or prefer to use a less stringent quality scheme for the data you collect, each type of error handling can be turned off individually when generating WI (see 7.2.1). The system always performs size checks, because failing to do so will cause database errors. A description of each test is provided below:

Required answer testing

Required answer testing exists in order to ensure that you get complete responses to your surveys.

- 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.

Note that for plain **multi** forms, an answer where no alternatives are checked are considered a valid answer, unless it contains “Single punch” alternatives which are subject to the exclusivity testing described next. Required answer testing may be turned off for separate questions by selecting “Not Required” in the question's properties (see 5.2.1.4).

Exclusivity testing

Exclusivity testing is used for **multi** questions that have one or more “Single punch” alternatives. 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.

The exclusivity test assumes that the alternatives given exhaust all possible answer combinations and at least one positive answer is required.

In consequence, the response will only be considered valid if the respondent checks at least one of the none-exclusive or exactly one of the exclusive options.

Other-Specify validation

Other-Specify validation verifies that:

- The respondent provides a specification to an alternative that has a specify-field if that alternative is checked.
- That the alternative is checked/answered if a specification is provided.

Rank order testing

The system applies rank order testing when the “Ordered” option is selected for a form. **Multi** and **grid** answers must then constitute a set of consecutive integers starting at 1.

6.6.2 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.

6.7 Script Nodes

Script nodes typically contain code for

- Internal programming purposes
Example: Say for instance, you have the multi question the *Tvchannels* question from 6.2.1.3. You want a single question *most* after that one where you ask something like "And which of these channels do you watch most?" If the respondent answers just one in *TVchannels*, you would not want them to get this question. That can be done by adding a condition (see 5.2.2) with the code `f('TVchannels').size() > 1`, and move the single question *most* to the then-branch of the condition. But you would want to have the one channel they answered in *TVchannels* in the data of the *most*. This can be accomplished by making an ELSE-branch to the condition, and add a script node in it.

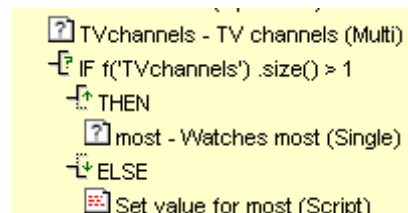


Figure 45: If-then-else construction

The script node should contain this simple code:

```
f("most").set(f("TVchannels"))
```

- Defining functions used in precode masks, conditions, other script nodes or validation code
Example: You have a grid in one question and want the next question only to include the answer alternatives which have been given the highest score, i.e “6” on a scale. You then write a function in a script node:

```
function onlyinclude6(){
    var s = new Set()
    var precodes=f("qID").domainValues();
    for(var i=0; i
```

In the precode mask property of the next question you call this function by using it's name. In this case `onlyinclude6()`

- Setting the values to hidden variables (see 5.2.1.4)
Example: Instead of asking a respondent which browser they use, you may include a hidden question Browser and set it's value by using the `BrowserType()` and `BrowserVersion()` functions in a script node.

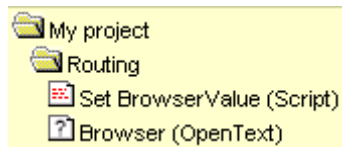


Figure 46: Setting a value for a hidden question

The only code in the BrowserValue script should be:

```
f('Browser').set(BrowserType()+BrowserValue())
```

7 Preparing for Data Collection

7.1 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.

NB: The system keeps two separate databases for your survey, **one test-database**, and **one production-database**. You generate the test-database from *Testing* → *Compile database*. The production-database is generated from *Production* → *Compile prod. env*.

When clicking “*Compile ...*” for the first time, you will be presented with a dropdown dialog, asking you to confirm the creation of the initial database. If you click *OK*, 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. The system also performs a check on whether the precodes of the Single and Grid questions do not occupy more space than allotted to them in Field Width. (See 5.2.1.4 Properties).

If the database already exists because of a previous compilation, you are presented with a slightly different dropdown dialog. You can choose between:

- Update existing database (incremental) – the system will update only the recently made changes.
- Update existing database (rebuild) – the system will update the whole database. You would e.g. use "rebuild" on your projects after a system upgrade or after some bug fixing has been carried out.
- Create new database – the system will create a completely new database. All existing data will be deleted.

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 delete all data in the database. You should NEVER do this in a running production-database.**

Login only:

To avoid unnecessary storing, FIRM does not guarantee that test databases are saved on the server for more than two weeks.

7.2 Generating Interview Files

Before you can start interviewing, you must have generated a production response database as described above.

Login only:

For generation procedure for CATI files, check separate manual.

7.2.1 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.

The URL to an open survey will always be

<http://www.yourconfirmitserver.com/wi/pXXXXXXX/i.asp>

Login only: The URL to the survey will always be

<http://survey.confirmit.com/wi/pXXXXXXX/i.asp>

WI Generators

You now have 3 choices for the WI Generators under the Test and Production menus. The *old* and *confirmit5.0* alternatives are available to provide support for legacy surveys that over time have used generator specific workarounds to accomplish functionality not directly supported by confirmit. Note that many of those workarounds were introduced to get around limitations that later confirmit versions have removed, so updating a legacy project can be relatively easy.

- **Generate WI (old):** Use this for projects created specifically for the first generation interview generator. Workarounds include use of `<%= %>` to do answer piping. This version of the generator was replaced by the 5.0 version on the 21st of February, 2000. If you have not updated your surveys yet, then now is a good time to do it!

- **Generate WI (confirmit5.0):** As for the "old" generator above, use this for projects with 5.0 specific workarounds. Surveys that rely on scripts which use non-published features of the interview runtime fall into this category.

- **Generate WI (default):** This is the latest-generation Web interview generator. It supports new features, such as 3D grids and international character sets, and produces precompiled code that makes interviews run significantly faster than what was the case earlier.

All projects, new and old, should be run on the new default generator unless they rely on any of the generator specific workarounds mentioned above. If your survey only uses documented confirmit features then the transition should be trouble free. If in doubt, please ask us!

NOTE! As a general rule, you should always test your projects before going live.

Web Interview Generator
Generates web interview files.

Access existing test interview:

- [English](#)
- [Norwegian \(Bokmål\)](#)

Recipient of error notifications:

☐ Include progress bar.
☐ Reserve a fixed area for error messages.

Survey design:

☒ Generate back button.

☒ Allow user to modify answers after initial submission.
☒ Allow user to modify answers after the interview is complete.

Use system's default validation for:

☒ Answer required checks.

☒ Exclusivity tests.

☒ Other-Specify checking.

☒ Rank order tests.

☐ Limited survey (Requires a respondent list).

☐ Generate login page (User ID/password must be provided with respondent data).

☐ Generate pop-up survey.

Default language:

Figure 47: Web Interview Generator options

Once you have generated the survey in Test mode, you can access it again by going through the *Project menu > Test > Generate WI*. In the field *Access existing test interview* you will find the link to the test survey.

7.2.2 Options Controlling Appearance

7.2.2.1 Recipient of Error Notifications

In this field you can enter an email address you wish to receive survey error notifications to.

Whenever there is an error in the survey, the respondent will get a general error message:

“Sorry, but the system has encountered an internal error and cannot go on with the interview at this stage. Please check back later.”

The addressee of the email address entered in the WI Generator, on the other hand, will receive a detailed description of the error and its whereabouts, e.g.:

**** Error info ****

Date : Fri Apr 28 01:06:58 UTC+0200 2000

Project : pXXXXXXX (The Name of the Project)

Respondent: 1

**** Error description ****

Problem encountered while processing question 'q22':

Error in mask:

f("q21").add("99")

Object doesn't support this property or method

The error message specifies the project ID, the title of the project, the question, and the erroneous script, precode mask, condition, and etc. expressions.

7.2.2.2 Include Progress Bar

This option instructs the generator to include a progress bar on the interview pages. This bar shows the respondent how much of the questionnaire is remaining (in percentage).

7.2.2.3 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.

7.2.2.4 Survey Design

In this field you may choose among WI templates you have predefined (see 3.6 Templates) and apply them to your project.

7.2.3 Options Affecting Interview Behavior

7.2.3.1 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.

7.2.3.2 Allow user to modify answers after initial submission

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.

7.2.3.3 Allow user to modify answers after the interview is complete

This option controls whether the respondents can enter the survey after they have responded to the whole interview and submitted it by pushing OK button on the last page.

7.2.3.4 Answer Required Checks

The generator produces code that ensures that the respondent answers a question unless it has the “Not required” property set. This option applies to all Open, Single and Grid questions, and Multi questions with an exclusive answer alternative defined as single punch.

7.2.3.5 Exclusivity Tests

The generator produces code that ensures that the respondent has answered the exclusive answer alternative defined as single punch in a Multi question, without checking additional answer alternatives in a list.

7.2.3.6 Other – Specify Checking

The generator produces code that confirms the consistency of user input, i.e. that the respondent does both, checks off for Other and writes in the text box,- and vice versa.

7.2.3.7 Rank Order Tests

The generator produces code that ensures that the inputs in Multi Ordered questions and Grid questions are unique for each alternative in the list and consecutive. The questions must be marked as Ordered in Properties (see 5.2.1.4 Properties).

7.2.4 Options for Generating Limited Surveys

7.2.4.1 Limited Survey

The generator supports two kinds of surveys: limited and open. Limited surveys can only be accessed by respondents identified in a respondent list. Each respondent may automatically be provided with a unique link to the survey, which includes identification credentials. Alternatively, he has to identify himself on a login page.

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.

7.2.4.2 Generate Login Page

As mentioned, **confirmit** can provide 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. See 8.1 Preparing the Respondent List.

7.2.5 Pop-Ups

7.2.5.1 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.

7.2.5.2 An Example of a Script for Pop-up Surveys

If you want to do a pop-up on a website, a script will have to be included in the code of the page where you want the pop-up to be opened from. There are many different solutions for this, depending on when you want the pop-up to come up etc. Below is an example of a script you could use.

```
<script>
if (document.cookie.indexOf ("FIRMpXXXXXXX=true")==-1){
    if ( Math.random() < 0.2){
        document.cookie = "FIRMpXXXXXXX=true;path=/;Expires=Mon,
1 Jan 2001 12:00:00 UTC"
        window.open("http://www.yourconfirmitserver.com/wi/pXXXXXXX/i.a
sp", "windowname",
"toolbar=no,location=no,directories=no,status=no,menubar=no,resizable
=yes,copyhistory=no,scrollbars=yes,height=360,width=300")
    }
} </script>
```

Change pXXXXXXX to the actual project number.

The script first checks if the respondent has a cookie already (then he/she has already gotten the pop-up and it should not be opened again). Math.random gives a random number between 0 and 1. Math.random() < 0.2 will mean that 20% of those who open the web page (for the first time) will get the pop-up. document.cookie will send a cookie. Expires: Lifetime of the cookie. Set this to a date some time after the survey is finished.

Window.open opens the pop-up window. Change the url to the url of your survey. windowname may be changed into a name of a specific window if there already is a window opened that you want the pop-up in, or you may use it to refer to the window from other scripts. You may leave this empty ("").

The other parameters control the pop-up window:

toolbar=no- allow the window to have or not have a toolbar.

location=no - allow the window to display its location or not.

directories=no - allow the window to display directory buttons or not.

status=no - allow the window to have a status bar at the bottom or not.

menubar=no - allow the window to display a menu bar or not.

resizable=yes - make the window resizable, borders are draggable.

copyhistory=no - allow for tracking of links activates the back or forward buttons.

scrollbars=yes - automatically include scrollbars on the window if needed.

height=360 - height of the window in pixels.

width=300 - width of the window in pixels

7.2.5.3 .Pop-up Script Wizard

In Test modus, after having generated a web interview, you are presented a link, e.g. English, to the test survey. When you enter the link to a survey that has been generated as a Pop-up, you will immediately be presented with a Pop-up Script Wizard. The Wizard includes an example of a Pop-up script, very similar to the one included in 7.2.5.2, as well as it gives the user an opportunity to adjust several of the Pop-up parameters: window appearance, display frequency, and cookie.

The changes you make to the Pop-up script will be applied the next time you generate web interview in the Test mode.

7.2.6 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.

7.3 Project Status

You can set the project status on this page.

PROJECT STATUS _____ p0793814 _____

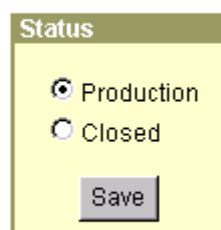


Figure 48: Project status

Production: The survey is running.

Closed: The survey has been closed.

7.4 Project Log

The system records every action executed on a project basis. In *Production > Project Log* you will have an overview of all changes that have been made on the project. The system records the exact time when changes are made, the user who makes the changes, and specifies every change.

PROJECT LOG p0793814

Project Log		
Date	User	Event
Log Entries		
08.feb.00 16:23:56	Smith, John	Project created.
08.feb.00 17:13:04	Smith, John	ID of node "q1_1" changed: "q1_1"->"q2_1"
08.feb.00 17:13:04	Smith, John	Property "randomize" of node "q2_1" changed: ""->"false"
08.feb.00 17:13:04	Smith, John	Property "recodevar" of node "q2_1" changed: ""->"false"
08.feb.00 17:13:04	Smith, John	Property "backgroundvar" of node "q2_1" changed: ""->"false"

Figure 49: Project log

7.5 URL Setup

You have the opportunity to control which website the respondents will come to, when they have finished the survey. When you click *Project > Production > URL Setup*, you will be brought to this page.

URL SETUP (PROD DATABASE) p0793814

Applies to web interview projects only.

URL specifications		
English		
	URL	Text
Help link:		
End link:		
Save		

Figure 50: URL setup

Field	Description
Help link	In this field you enter a URL address to a website you wish to use either for additional information or for another purposes in the questionnaire. In this case you have to have included primitive ^ HELP^ in the WI template you will apply to this survey.
End link	In this field you enter the URL address of the website you wish the respondents to be brought to after they have finished the survey.
Text	In this field you may control the appearance of the link, i.e. you may substitute the address with another text.
Link preview	In this column you will get a preview of the link.

Table 26: URL setup

Usually, the End link must be defined before you compile the survey database: *URL Setup > Compile prod env.*

8 Handling Respondents in Limited Surveys

For 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. Chapter 8.3 also describes how to automatically send e-mails with invitations and reminders for participants of web surveys.

8.1 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 - .txt)*.

This data file is prepared on the user's computer and is subsequently uploaded into **confirmit**.

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	Limited WI
name	The full name of the respondent	opt
language (for multilingual projects)	The respondent's preferred language. The survey will be presented in this language (see appendix for language codes)	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

Table 27: Fields in respondent lists

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. See 5.2.1.4 Properties.**

8.2 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 into **confirmit**. This is accomplished by selecting the menu *Respondents/Uploading*. 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 **confirmit** system. 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 were uploaded.

8.3 Sending E-mail

confirmit 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 *Respondents > Emailing* 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.

8.3.1 Step 1: Respondent Selection

Setup

SELECTION CRITERIA

Respondent status

noOfEmailsSent

name

email

PREVIEW OPTIONS

Max no of preview records

EMAIL

Mail format

Send as HTML ☐

Sender

Subject

Body

Include link ☒

Next >>

Figure 51: E-mail setup

Please note that this screen may vary depending on the fields you have included in the respondent file that you are using. All fields will be presented under *selection criteria*.

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

Criterion	Description
Respondent status	Chooses respondents based on the status of the interview, see table below.
Number of emails sent	The number of e-mails sent is recorded for each respondent, and can be used as a selection criterion.
<i>Other fields:</i> eg “email”	The wizard will display any additional fields from the respondent file that may be used for selection, including sample categories and quota variables.
Status	Description
(Any)	All respondents
Not answered	All respondents who have not entered the survey
Complete	All respondents who have completed the survey. (Useful for sending “Thank you” e-mails).
Incomplete	All respondents who have started answering the survey, but have not completed it.
Screened	All respondents who have screened status (See 5.2.4 Stop-objects).
Quota full	The interview is stopped for a certain segment of respondents due to a full quota.

Table 28: Respondent selection criteria/status

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 available 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 than 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*, *LIKE* etc. For instance, to find all respondents whose names start with ‘B’ (assuming the respondent list includes name as a background variable):

LIKE ‘B%’

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. 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 (default) or plain text e-mail. Normally MIME will work best on most e-mail programs.
Send as HTML	Emails may be sent as HTML codes. Check the "Send as HTML" box to send HTML coded email invitations. Comment: Only a few mail systems will display such emails correctly. Use with caution. When used, "MIME" format should also be selected.
Sender	Enter the e-mail address that is to appear in the <i>From</i> 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 at the bottom of the e-mail (default).

Table 29: Email setup fields

8.3.2 Mail Merging Functionality

To merge a background data field into the body text, for instance the respondents' e-mail address, write:

`^email^`, referring to the uploaded email column (must be written exactly as the column label).

It is possible to display the link to the survey within the e-mail text by simply referring to it in the following way:

- `^slink^`

Make sure that:

- "slink" is not used as background data;

The reference `^slink^` should always be written on a separate line in order to avoid breaking the link over two lines.

8.3.3 Avoiding Unnecessary Respondent Inquiries

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

Please, read the following before you enter the survey!

- Cut (Copy) – and – Paste the link into the Internet address window in case you cannot access the survey by simply clicking on the link. Make sure it's on one single line and without any spacing.
- Navigate back and forth by clicking on the >> and << buttons, or just >> if the back button is disabled. Do not use the browser's "back" and "forward" buttons, or your answers may not be saved!
- If you are not able to switch to the next page, you might have forgot to answer all questions on the page. Look for an error message.
- 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!
- The link is unique! Please, do not send it to other respondents.

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

If you are including the unique links to the survey at the bottom of the email message by checking *Include link* flag, make sure that the mouse cursor is at the beginning of a new line. Thus you can prevent that the link to the survey merges with the body of your email message.

8.3.4 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. The maximum number specified under Preview options in step 1 limits this list.

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.

8.3.5 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.

When sending emails, a standard check will be run on the email list. The message will not be sent to invalid addresses. The **confirmit** user will receive a mail from the system on how many emails have been sent, and how many mails it has failed to send.

After having previewed your email message and the respondent list, you can either click *Back* to make changes if necessary or, if you are ready to send your survey out, *Next*. Clicking on *Next* will bring you to the screen shown in Figure 52: Batch Emailing.

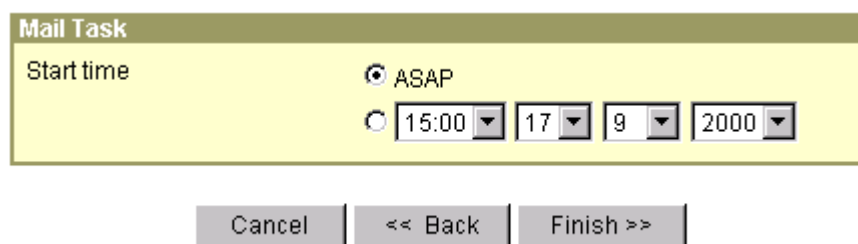


Figure 52: Batch Emailing

Here you can schedule when you wish your survey to be sent out. Clicking on *Finish* will bring you to the *Task Manager* where you can monitor the progress of your batch job.

Properties

Log

Task instance properties

Property	Value
Task type	 Batch Emailing
ID	490
Schedule	2000-09-17 15:18:05
Status	In queue
Approx % completed	0%
Start time	-
End time	-
Time usage	00:00:00
Assigned server	

Task definition

Figure 53: Task properties for batch emailing

A confirmation from the system will be sent to the address you have specified in your own User Settings, even if you have specified another sender address in the *sender* field.

8.4 Handling Respondents in Multilingual Projects

In multilingual projects it is necessary to ensure that the respondents get the survey in the correct language. There are several methods of controlling this for Limited and Open surveys.

8.4.1 Open Multilingual Surveys

There are two ways of securing that respondents get the survey in a particular language:

- In open multilingual surveys you may have a Single question with the question ID **I** (as in language) as the first question of the survey. You will have to use the actual languages as answers, and the language codes (see Appendix B: **confirmit** Language Codes) as precodes.

The respondents will get the next question in the language they chose in the Language question.

- Another option is to distribute survey links specifying language, to the countries the survey will be conducted in, e.g.:

<http://www.yourconfirmitserver.com/wi/pXXXXXXXX/i.asp?l=0> (for English)

<http://www.yourconfirmitserver.com/wi/pXXXXXXXX/i.asp?l=6> (for Spanish)

etc.

8.4.2 Limited Multilingual Surveys

When it comes to limited surveys, you may:

- Include a column with a tag *language*, which specifies the language each respondent will get the survey in (see 8.1 Preparing the Respondent List). The contents of the column must be **confirmit** language codes (see Appendix B: **confirmit** Language Codes).
- The other option is to include a Single question with the question ID *l* in the survey, as mentioned above (see 8.4.1 Open Multilingual Surveys).
- In multilingual surveys with login-page (see 7.2.4.2) the respondents are presented with a drop-down where they can choose language.

9 Reporting

9.1 Introduction

All reporting functionality in **confirmit** is accessed by clicking *Reports* in the main menu. Click on the *Online Reporting* and you will enter this screen:

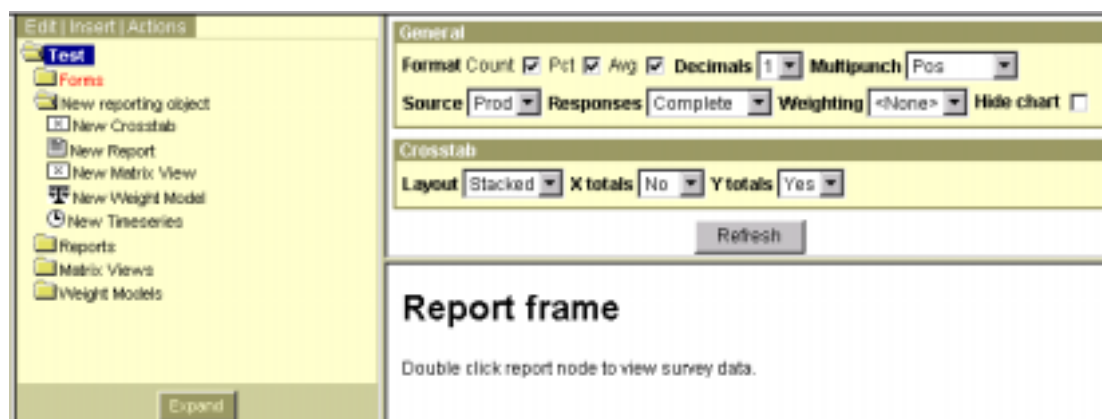


Figure 54: Initial reporting screen

This screen consists of three 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 General settings for the reports. These General settings can be overridden by setting specific properties for the report nodes. To get more space on the screen for tables and charts, the general option frame can be toggled. Toggling is done from the reporting tree menu: 'Edit->Toggle options'.
- **The report frame.** The frame where all report tables and charts are displayed, or the properties of the report node when chosen.

You may expand all trees by clicking the *Expand* button.

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.** Single-click on the folder icon. 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 routing to be able to find the question of interest.
3. **Double click the question.** When the question is double clicked, the results are shown in the bottom right frame as shown in Figure 55: Reporting a single question with Excel Chart Component.
4. **Expanding the Other-specify fields.** By clicking the icon of a form containing 'Other' answer alternatives, the form will be expanded. The sub-tree of the form will contain the open-text fields. These can be reported and dragged into reports just as any other open-text forms.

5. **Expanding Grids.** Single-click on the icon of a grid question to expand it. All the grid elements will be shown as single questions.

Age

"Age-How old are you?"

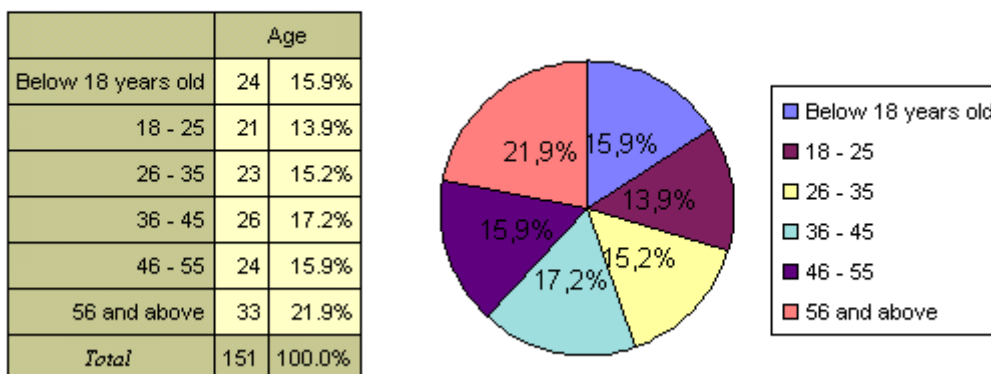


Figure 55: Reporting a single question with Excel Chart Component

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 of having more than one report is that you can prepare reports for different target groups and order exports (ppt/Excel files) separately. 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 in the report frame.
- **Answer mask for reports.** You can select the answers (for a single question) and the answers and scales (for a grid question) you want to be shown in the report. This masking can be executed for specified question forms in a report. You just select a question in a report you want to mask and go to *Edit* → *Properties* in the report tree menu. In the report settings in *Answer mask/Scale mask* field you may select the answers. See table *Report elements settings* in chapter 9.8.2 for details.
- **Adding questions to the report.** Drag the form from the *Forms* folder and drop it at the right location within the report.
- **Add cross tabulation 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.
- **Add time series to the report.** Drag a *New timeseries* object into the report, e.g. within a crosstab. Select the *New timeseries*, and then go to *Edit* → *Properties* from the report tree menu. Afterwards configure the time series with granularity and start/end points.
- **Delete.** Select *Edit* → *Delete* in the report tree menu. Can be applied to both, individual questions/crosstabs and to whole report folders.
- **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, and you will be presented with the following screen:

Export properties

Project ID: p16169441

Report name: Test

PPT template: confirmit 01

Export files: MS Excel and MS Powerpoint

Database: Production

Language: English

Recipient email: john@anywhere.com

Comment:

Start time: Schedule for later execution

ASAP

Schedule for later execution

Figure 56: Report Export

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. In User Settings you may specify in which MS Office version you wish the report to be sent.

Field	Description
Project ID	The system generated project number will be displayed.
Report name	The report name will be displayed.
PPT template	Here you can choose a PPT template you wish to apply to your report.
Export files	Here you can choose whether you wish to export to either MS Excel, MS PowerPoint or to both.
Database	Here you can specify what kind of data you wish to export: Test or Production data.
Language	Here you can choose the language you wish the report to be exported to. Remember that this choice has to correspond with the language in your User Settings.
Recipient email	Here you can specify the recipients email address.
Comment	Here you can enter a comment for your batch job.
Start time	You can either schedule the report to arrive immediately, or at a later point. After having submitted the request you are brought to Task settings, see Figure 57: Task settings for an ordered report.

Table 30: Report Export

 The request has been added, see below.

Properties **Log**

Task instance properties

Property	Value
Task type	 MS Office Export (PPT/Excel)
ID	492
Schedule	2000-09-17 17:26:32
Status	In queue
Approx % completed	0%
Start time	-
End time	-
Time usage	00:00:00
Assigned server	

Task definition

Figure 57: Task settings for an ordered report



Figure 58: Project with two reports.

The charts in PowerPoint reports are saved as pictures. This means that it is not possible to make any changes to ordered PowerPoint reports. If you wish to make changes to the charts in ordered reports, we recommend that you order Excel report, make your changes and prepare a PowerPoint presentation afterwards.

Figure 59: Crosstab: No of visits vs. likelihood shows a report, *June total*, with one crosstab and one single punch question.

9.4 Building Crosstabs

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

- The “Crosstab...”-folder where the vertical questions of the crosstab can be included. This is done by dropping questions within the folder.

- The “With...”-folder where the horizontal questions of the crosstab can be included. This is done by dropping questions within the folder.

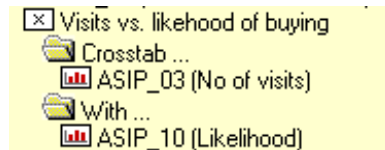


Figure 59: Crosstab: No of visits vs. likelihood

9.5 Reporting on Loops

The easiest way of checking the answers in a Loop is by entering the “Form”-folder in the reporting tree:

- **Open the “Form”-folder.** Single-click on the folder icon. This will show you the routing the same way as the “Routing”-folder in the questionnaire.
- **Locate the Loop.** Since the subfolders, condition trees and loops are all initially closed, you should know something about the questionnaire routing to be able to find the question of interest.
- **Single-click the Loop folder.** When the Loop is single-clicked, the questions the Loop contains will appear. If you single-click on the form node, all iterations for the loop question will appear. When you double-click on each iteration, the results will be shown for only the particular iteration (see Figure 60).

Satisfaction

How satisfied were you with ^f('cars')^?

	Satisfaction	
Alfa Romeo		
1	0	0.0%
2	0	0.0%
3	0	0.0%
4	0	0.0%
5	0	0.0%
Total	0	0.0%

Figure 60: Showing results for single iteration

- **Adding loops to reports.** You can add a loop question to the report by drag and drop method. You single-click the loop folder in *Forms*, and all the questions in the loop will appear. Select the question you wish to report on, and drop it into the *Report* folder.
- **Loops in crosstabs.** You may cross tabulate the questions in the loop with the loop. Insert a *New crosstab* into the *Report*. Insert the loop questions into the *Crosstab...* folder if you wish

them to appear vertically, and the loop folder into the *With...* folder if you wish the iterations to appear horizontally; or you may do it vice versa (see Figure 46).

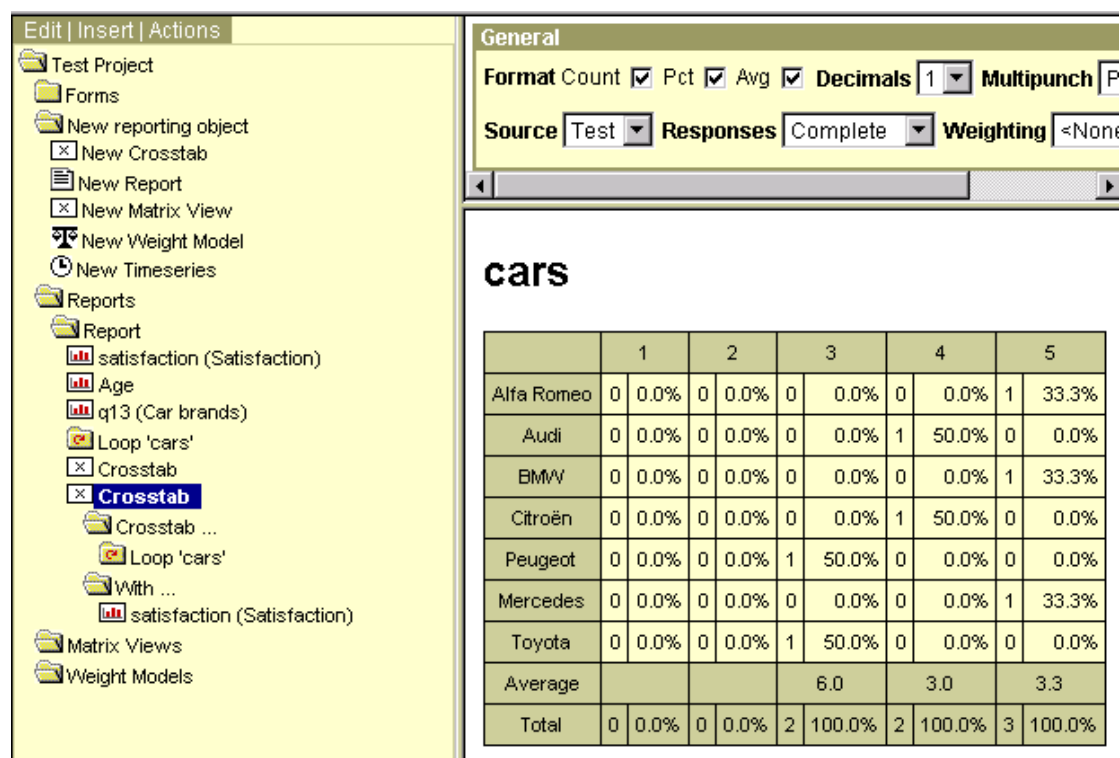


Figure 61: Loop in crosstab

For more detailed information on how to edit reporting elements of loops, see chapter 9.8.2 Report Elements Settings.

9.6 Weight Model

CONFIRMIT allows weight models to be applied to your reports. Three separate types of weight models may be used.

Weight Model Type	Description
Sample balancing	This method is used for weighing along multiple dimensions when only the marginal reference percentages along each dimension are known. E.g: It is known that the reference population is 30% male and 70% female and that 60% of the population own a car while 40% do not. From these facts the actual weights are calculated according to the standard sample balancing algorithm. Max iterations and deviation limit for this algorithm decide how accurate calculations are required.
Cell matching - percentage	This method is used for weighing along multiple dimensions when all the matrix cell percentages are known. E.g:

	It is known that: • 20% are male with cars • 10% are male without cars • 40% are female with cars • 30% are female without cars From these facts, the weights can be calculated by dividing the sample percentages by the reference percentages.
Cell matching – weight	With this method all the weights are entered directly and not calculated by comparing a reference with the sample. The method may for example be used if the weights are calculated by another system.

Table 31: Weight Model

Create a new weight model by dragging the "New weight model" object into the "Weight models" folder. Double click the new object to edit its properties. The percentages for the cell matching methods are entered from this property screen.

Drag and drop the associated questions into the new weight model. For "sample balancing" (default), double click the individual questions in the weight model folder to set their percentages.

Properties	
Weight Model Name:	Female/male
Type:	Sample balancing
Max Iterations::	99
Deviation Limit:	0.1
Status:	Generated
Last generated:	Thu Nov 18 13:04:56 UTC+0100 1999
Last log	

Figure 62: Weight model properties

Marginal percentages for 'q2'	
Male[1]:	49
Female[2]:	51

Figure 63: Percentages

Respondent weights are calculated when the user selects [Action => Calculate weights] after first having activated the weight model folder. Weight models may be applied to report folders and/or as a general report option. Report folder model overrides the generally selected model.

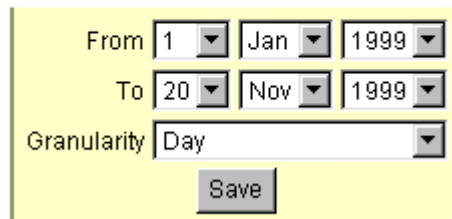
It is not possible to apply weighing on questions that were answered only by part of the respondents (filtered questions).

9.7 Time Series

Use time series to create trend reports.

1. Drag a 'New timeseries' object into the report, e.g. within a crosstab.
2. Select the new time series and select 'Edit->Properties' from the report menu.
3. Configure the time series with granularity and start/end points (day, week, month, quarter, year).

TIME SERIES DEFINITION —



The dialog box is titled 'TIME SERIES DEFINITION'. It contains three rows of controls. The first row is labeled 'From' and has three dropdown menus with values '1', 'Jan', and '1999'. The second row is labeled 'To' and has three dropdown menus with values '20', 'Nov', and '1999'. The third row is labeled 'Granularity' and has a dropdown menu with the value 'Day'. Below these controls is a 'Save' button.

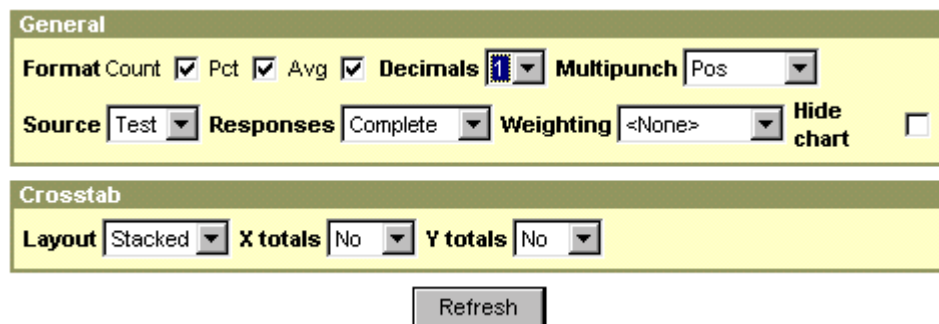
Figure 64: Time series properties

Time series may be used in cross tabulations or as a stand-alone report item.

9.8 Settings

9.8.1 General Settings

The settings frame in the reporting screen includes the following general settings for the whole reporting environment. Every time you make any changes in the General Settings you have to click the *Refresh* button in order to update the settings.



The settings frame is divided into two sections: 'General' and 'Crosstab'. The 'General' section contains the following controls: 'Format' with checkboxes for 'Count', 'Pct', 'Avg', and 'Decimals' (set to 2), and a 'Multipunch' dropdown set to 'Pos'. Below this are 'Source' (set to 'Test'), 'Responses' (set to 'Complete'), 'Weighting' (set to '<None>'), and a 'Hide chart' checkbox. The 'Crosstab' section contains 'Layout' (set to 'Stacked'), 'X totals' (set to 'No'), and 'Y totals' (set to 'No'). A 'Refresh' button is located at the bottom of the frame.

Figure 65: General settings frame

Setting	Setting description
General	
Source	Indicates whether test or production data should be reported.
Format	Indicates whether count, percent, and/or average should be displayed. Average can only be displayed for weighted single punch or grid questions. See chapter 5.2.1.2 for how to give values/weights to a question.
Decimals	Number of decimals to use in average and percent format.
Responses	By default set to show only complete interviews, but can also show only incomplete interviews or both complete and incomplete.
Multipunch	Pos: Shows only numbers of positive answers for a multi punch. Pos + Neg: Show numbers of positive and negative answers for a Multi question (answered/not answered) Ordered: Shows an ordered multi punch split in ranking.
Weighing	<None> or one of the defined Weight models
Hide chart	Use this check box to show only tables in the report frame.
Crosstab	
Layout	Stacked: The questions in the “With...”-folder are stacked. Nested: The questions in the “With...”-folder are nested.
X totals	Set this to ” Yes” to get the totals for a question to the right of the rest of the crosstab.
Y totals	Set this to ” Yes” to get the totals for a question under answer alternatives.

Table 32: General settings options

9.8.2 Report Elements Settings

All of the general settings can be overridden by setting properties for each single element within a report. The element would be a crosstab or a simple question within the report. This is done by selecting a report element, single-clicking on it, and then choosing *Edit* → *Properties* in the above the tree structure menu. From this menu point, you can also set the following properties that are not available from the general settings:

9.8.2.1 Excel Chart Component

The following screenshots will be available for you if you choose Excel as the chart component in user settings (see 3.2 User Settings).

General options

Format ☒ Global setting ☐ Count ☐ Percent ☐ Average

Averages ☒ Collapse in tables ☒ Include base

Decimals

Multipunch

Deleted ☐

Chart options

Chart type:

Chart subtype: (3D only available in exports)

	Border Color	Border Linestyle	Interior Color
Chart	Automatic	Automatic	Automatic
Plot Area		Automatic	
Legend	Automatic	Automatic	Automatic

Chart content

Min Scale

Max Scale

☐ Series in rows
 ☒ Series has values
 ☒ Show Legend

☐ Excel chart has table
 ☐ Hide chart (Online reporting)

☐ Categories in reverse order
 ☐ Values in reverse order

Figure 66: Chart options for Excel Chart Component

Chart Options Excel Component	
Setting	Description
Type	Defines the chart type to be used. The most normal chart types like bars, pies, lines and areas are supported among some other.
Subtype:	
Border Color	(chart and legend) The default border color of charts and legends is set to be black. The changes are invisible in confirmit . Nevertheless, they will be activated when ordering the report. Double-click on the fields in order to activate the Color Selection.
Border Linestyle	(chart, plot area, legend) The changes are invisible in confirmit . Nevertheless, they will be activated when ordering the report.
Interior Color	(chart and legend) Double-click on the fields in order to activate the Color Selection.
Chart content	Indicates whether count, percent, and/or average should be displayed. Average can only be displayed for weighted single punch or grid questions. See chapter 5.2.1.2 for how to weigh a question.
Min Scale	Indicates the minimum value on a scale.
Max Scale	Indicates the maximum value on a scale.
Series in rows	The graphical representations of the categories, e.g. bars or columns are placed next to each other, each category having its own legend.
Excel chart has table	The changes are invisible in confirmit . Nevertheless, they will be activated when ordering the report.
Categories in reverse order	If you check off for this option, the categories in the chart will be shown in reverse order as compared to the table, e.g. male – female in the table, but female – male in the chart.
Series have values	Table values of each category are explicitly specified in the chart.
Hide chart (Online reporting)	Use this check box to hide the charts in the report frame.
Values in reverse order	The values in the chart, i.e. count, percent or average, will be shown downwards and not upwards.
Show legend	You may choose whether you wish the legend to be displayed in the chart.

Table 33: Excel chart component options

9.8.2.2 PowerPoint Options

Power Point options can be edited in all project languages simultaneously. When you order a report, it will arrive in the language you have specified in your *User Settings*.

If you have edited Report Publisher options prior to editing Power Point options, you can easily copy Publisher Header texts and Texts into your Power Point presentation by clicking *Copy Publisher texts* button.

The screenshot shows a software interface with a tabbed menu at the top: General, Chart, Ppt, Publisher, Filter, Masks, Save. The 'Ppt' tab is selected. Below the tabs is a section titled 'Power Point options'. Inside this section, there are two dropdown menus: 'Object type' set to 'Chart' and 'Slide layout' set to 'Only object'. Below these are two language sections. The 'English' section has a 'Header' label followed by a single-line text input field, and a 'Text' label followed by a multi-line text input field with a vertical scrollbar. The 'Norwegian (Bokmål)' section has a 'Header' label followed by a single-line text input field, and a 'Text' label followed by a multi-line text input field with a vertical scrollbar. At the bottom center of the dialog is a button labeled 'Copy Publisher texts'.

Figure 67: PowerPoint options

PowerPoint options	
Settings	Description
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.

Table 34: PowerPoint options

9.8.2.3 Filter Options

See 9.9 Filters.

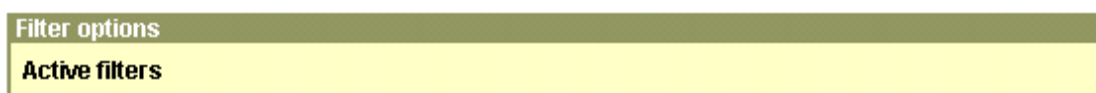


Figure 68: Filter options

Filter options	
Setting	Description
Active filter	It is possible to set any of the defined filters active for one report element.

Table 35: Filter options

9.8.2.4 Answer Mask/Scale Mask

Answer mask

☒ Show all answers

Show these answers:

☐ Advertising on TV
☐ Special offer in store (e.g. discounted 3-pack)
☐ See friends drink it/tried it at friends`
☐ Curiosity about new brands seen in the store

Scale mask

☐ Show average
☐ Show best
☐ Show worst

☒ Show all answers
Show these answers:

☐ Not important at all
☐
☐
☐
☐ Very important

1
2
3
4
5

Figure 69: Answer Mask/Scale Mask

Answer mask/Scale mask	
Setting	Description
Show average	Choose this to show the average values of the variables. The answers/scales have to be assigned values to as described in 5.2.1.2.
Show best	The best average value is shown.
Show worst	The worst average value is shown.
Show all answers	All answers/scales are shown in the table/chart.
Show these answers	You may check off specific answers/scales you want to be shown in the table/chart.

Table 36: Answer mask/Scale mask options

9.8.2.5 Crosstab Options

Figure 70: Crosstab options

Crosstab options	
Setting	Description
Layout	Stacked: The questions in the “With...”-folder are stacked. Nested: The questions in the “With...”-folder are nested.
X totals	Set this to ” Yes” to get the totals for a question to the right of the rest of the crosstab table.
Y totals	Set this to ” Yes” to get the totals for a question under answer alternatives.
Table content is average of first node	The contents are the average values of the first question in the <i>Crosstab...</i> folder. The variable must be weighed to get average values.

Table 37: Crosstab options

9.8.2.6 Loop Options

Loop options

☐ Show average
☐ Show best
☐ Show worst
☒ Show all iterations

Show these iterations:

☐ Alfa Romeo
☐ Audi
☐ BMW
☐ Citroën
☐ Peugeot
☐ Mercedes
☐ Toyota

Figure 71: Loop options

Loop options

Loop id: cars
☐ Ignore 'cars'
☒ Show all iterations

Show these iterations:

☐ Alfa Romeo
☐ Audi
☐ BMW
☐ Citroën
☐ Peugeot
☐ Mercedes
☐ Toyota

Figure 72: Loop Question options

Setting	Description
Show average	Choose this to show the average values of the variables. The answers/scales have to be assigned values to (weighed).
Show best	The best average value is shown.
Show worst	The worst average value is shown.
Show all iterations	Check this box if you wish all iterations to be shown. Otherwise, uncheck this box, and mark the iterations you wish to report upon instead.
Loop id	Denotes the loop you are editing.
Ignore 'loop id'	This choice is available only for the questions in loops. If you check this box, the iterations will not be shown. Therefore, we would recommend unchecking it when reporting on the loop questions alone. On the other hand, you may check it when cross tabulating loops and loop questions, in order to prevent iterations from appearing both vertically and horizontally.

Table 38: Loop options

9.9 Filters

9.9.1 Defining Filters

Filters are defined like this:

1. Select the question to be used as filter within the “Form”-folder.
2. *Actions* → *Add Filter* from the menu above the reporting tree.
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).

DEFINE FILTER

p2974940—

Edit name and/or answer alternatives for the filter.

Element	Use
1 (Below 18 years old)	☐
2 (18 - 25)	☐
3 (26 - 35)	☐
4 (36 - 45)	☐
5 (46 - 55)	☐
6 (56 and above)	☐

Figure 73: Filter definition

9.9.2 Applying Filters

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

- **Global filters:** Select *Online Reporting* → *Define filters* from the main **confirmit** menu. Then check the correct filter in the list.
- **Element level filters:** See Filter options in 9.8.2.3

9.10 Report Publisher

9.10.1 Publishing a Report

The **confirmit** Publisher allows you to make your reports available to anyone you want. You select the data/charts to display, and **confirmit** updates results according to your settings.

The reports may be published on the WEB (either openly available or password protected), or attached as an HTML file to an e-mail. If your company has installed **confirmit** in-house, the location for publishing reports (either your intranet or the WEB) has been decided by your company.

To be able to publish a report, you must create one as described in chapter 9 Reporting. The settings you choose for the report are also applied to the Publisher. However, note that the Publisher engine does not support 3D graphs. Also check that you have Excel as your *Chart Component* in *User Settings*, or you will get a different result published than what you see in the report.

To edit charts to be published, open the relevant report from the *Reports* folder, and single-click on the slide you want to edit (as shown in the picture below). Choose *Properties* from the *Edit* menu, and make the adjustments you have to. For more information about the options available to you, see User Manual chapter 9 Reporting.

Publisher options can be edited in all project languages simultaneously. When you publish a report, it will arrive in the language you have specified in your *User Settings*.

If you have edited Power Point options prior to editing Publisher options, you can easily copy Power Point Header texts and Texts into your Published Report by clicking *Copy Power Point texts* button.

If you do not edit the Publisher options, the question titles will be by default piped in as Header texts.

General	Chart	Ppt	Publisher	Filter	Masks	Save
---------	-------	-----	------------------	--------	-------	------

Publisher options

Content Only chart ▾

English

Header

Text

Norwegian (Bokmål)

Header

Text

Copy PowerPoint texts

Figure 74: Editing a slide to be published

Settings that are unique for Publisher, are found in the *Publisher Options*-window (picture above). You can insert *Header text* and / or *Text*. They will both be placed above the tables/charts being published. Your text fonts and colors can be defined by standard HTML commands. You can also choose whether to show tables, charts or both.

After having prepared the report, you publish it by highlighting the report name (in the example above: R1), and choose *Publish Report* from the *Actions* menu. The following screenshot will appear:

Selected Properties for process "p1/0103205"									
Report	Database	Default lang	Creator	Targets Users	Template	1 page	Scheduling Status	Last run	Run now
Report	Production	English	johns	Targets users	confirms-and-S		☐ Scheduled	Failed	Never
							☐ Run		
Save									

Figure 75: The Publisher screenshot

The settings available to you are:

Setting	Description
Report	The name of the report being published
Database	Can be either Test or Production
Dafault lang	Language (based on your <i>User Settings</i> at the time of publishing). The reports will be published in all project languages with a unique link for each language under Targets.
Creator	Creator's name
Status	Success or failed
Scheduling	Allows you to choose how often the data is to be updated. See 3.4 Task Management and Figure 82: Recurrence pattern.
Targets	Settings for how the report is to be published.
Users	Users allowed to view the report when publishing on a <i>protected link</i>
Template	Report layout (see 3.6 Templates).
One page	All reporting on one page
Last run	Time of latest update of data
Run Now	By activating this, the report will be updated on the next engine run, even if the frequency setting requirements are not met (the FIRM engine normally runs every 5 minutes).
Delete	Allows to delete a published report
Remove	The report is being removed from the web.

Table 39: Report publisher options

You must always define *Frequency* and *Targets* (and *Users* if you publish on a *protected link*) before a report is available for publishing.

About **Scheduling**: **confirmit** will update the published report every x respondents (count), or at every given percentage increase of the respondent base (percent) or at a given time interval . The procedure is: Click Add, then choose a parameter, click save, enter the alternative you wish, and click save again. You can also combine different parameters.

About **Target**: Following the same add / save procedure as described above, you can choose three ways of publishing the report.

- The e-mail option sends the report as attachments to the mail-addresses you choose (each chart will be sent as separate gif).
- Cryptic link places the report on the address automatically generated. You can send the link to your users, or make it available from a home page. Anyone with knowledge of the link can view the report.
- Protected link places the report on the address automatically generated, but users need a *userid* and a *password* to access. You should have prepared a list of users as a tab delimited text document, with column headings *userid* and *password*. Click Users, Upload, Append or Overwrite, browse for the source file and complete by uploading. You must make login information available to the users.

Published Report targets for report 'R1'

Type	Value	Delete
Cryptic link	http://www.firm.no/sr/p3253595/FZIPOGHCIIE/2/0.htm	☐
<Select one>		☐

Back

Figure 76: Targets

NB! Depending on the browser's settings of those entering the report, they might have to refresh (F5) each page of the report when entering a second time to get the updated data.

9.10.2 Changing a Report after Publishing

The report in the Report Folder is the source of the published report. So if you e.g. want to change the chart type, or edit the Publisher Options, the changes will be applied the next time the Publisher engine runs (based on *Frequency settings* or *Run now* activation).

You cannot however change the overall settings after publishing. So if you are publishing a report with Count Format (from *Format* in the *General*-window in figure 59), and later wish to change to Percent, you will have to publish the report once more after having changed the settings. The same applies for change in language, or if you want to remove or insert a chart after publishing. In all these cases you must publish once more. This, however, does not apply to changing from Test data to Production data in *Source* in *General Settings*.

Outdated publishings are removed by clicking Delete and Save.

9.10.3 Publishing Different Data from One Survey to Separate Target Groups, or Same Report in Different Languages

You might want to make different results from a survey available to different target groups. You have to prepare the reports separately in *Online Reporting*. Thereafter you publish them one by one, selecting the wished publishing mode.

If you need the report to be in different languages, you just have to duplicate the report in the Edit Menu. Enter information in the relevant language in the Publisher Options, change your User settings to the relevant language, and publish. Do these changes every time you publish a report in a new language.

9.11 Matrix Views

9.11.1 Introduction

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

Question set	Description
Questions	These questions are placed in the left part of the matrix. Only these questions can be reported as single questions within the matrix view.
Cross questions	These questions are placed in the bottom part of the matrix. The questions can be cross tabulated against the cross questions.
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 40: Matrix view question sets

9.11.2 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.** Dragging a “New Matrix View” -object from the “New reporting objects” -folder into the “Matrix Views” -folder creates a new matrix view.
- **Change name.** Select *Edit* → *Properties* in the report tree menu, and change the matrix view name and id 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.

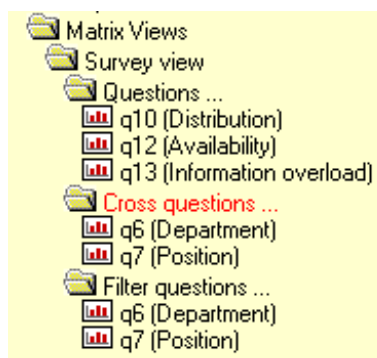


Figure 77: Matrix view configuration

9.11.3 Using Matrix Views

When the matrix view is initially entered, no charts or tables are displayed. The table below shows what options are available in the matrix view.

NB! The chart or table must be refreshed every time the user changes options.

Option	Description
Select a question	By pressing a question part of the matrix, a question is selected. The question can be selected on and off by pressing them repeatedly. A dark color on the button indicates a selected question
Select a cross question	Cross-questions are operated the same way as the indicator questions. When a cross-question is selected, the selected indicators are crossed with this demography question.
Make filters available	Select the “ <i>Filters</i> ”-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 to the 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 “ <i>Display</i> ”-option.
Change format	Format can be switched between count, percent and index (average) by changing the “ <i>Format</i> ”-option.
Change matrix size	Changing the “ <i>Screen</i> ” -option can change the matrix size. There are predefined matrix sizes defined to fit the following screen resolutions: 800x600, 1024x768 and 1280x1024
Make timeseries available	Select the “ <i>Timeseries</i> ”-option. This will add a time series row below the chart area.
Apply a timeseries	In the time series 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 41: Matrix view operations

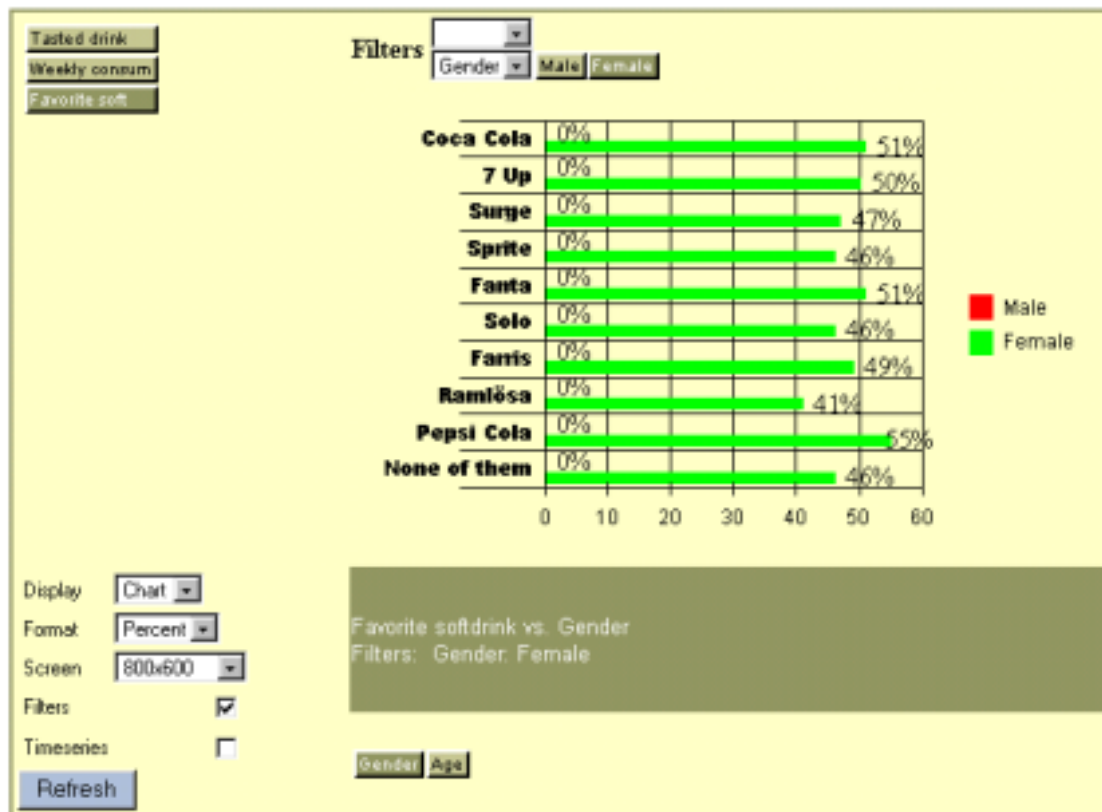


Figure 78: Matrix view

Here are some rules and guidelines for what can be done and what cannot be done within a matrix view.

- 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 5.2.1.2 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 cannot 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.

10 Recoding Data

Forms in the questionnaire can be turned into recoded variables by selecting [Properties => Recoded var] for the question. 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.

In addition to the forms in the questionnaire you can use the system generated variables interview_start, interview_end and status in recoded variables.

There are two kinds of recoded variables in **confirmit**, SQL based and JScript based variables. Calculation depends on what kind of variable is used. SQL based is the default variable type. Select [Properties => JScript based] to change the variable type.

The expressions are entered in the answer list of the recoded variable. To create a variable based on another specific question, it may be helpful to use the expression builder. This is activated by clicking the question you will base your variable upon when you are in the answer list of the variable.

10.1 Expression Syntax and Evaluation

Common for SQL and JScript based variable expressions is the way values of other questions are specified:

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

The values returned by such a reference is always a character. This is important to remember for rank order comparison. For strings, the value '11' is less than '2'.

10.1.1 SQL

For SQL based variables, any legal SQLServer primitives can be used within the expressions. Hence it is recommended to know some SQL to create advanced expressions. *Example 1:*

You have an opentext question 'age'. You want to report on 'Young' and 'Old'. Add a single question and select Recoded var, then save. In the answer page enter this information:

English	Expression	Precode
Young	<code>convert(int ,age)<40</code>	Y
Old	<code>convert(int ,age)>=40</code>	O

convert() is a SQL method for type conversion.

Example 2:

You have a multi('car') with precodes 'chevy', 'buick', 'opel', 'volvo' and 'saab'. After the survey is complete you want to make an aggregated report on American vs. European cars. Add a multi question and select Recoded var, then save. In the answer page enter this information:

English	Expression	Precode
American car	<code>car_chevy<>'0' OR car_buick <> '0'</code>	A
European car	<code>car_opel<>'0' OR car_vovlo<>'0' OR car_saab<>'0'</code>	E

Example 3:

You want to make an online report on the respondents' status, similar to the WI interview progress. Add a single question and select Recoded var, then save. In the answer page enter this information:

English	Expression	Precode
Screened	<code>status='screened'</code>	s

<i>Complete</i>	<i>status= 'complete'</i>	<i>c</i>
<i>Incomplete</i>	<i>status is null</i>	<i>i</i>
<i>Error</i>	<i>status= 'error'</i>	<i>e</i>

Observe that if a respondent leaves the questionnaire, status is not set (status is null).

In addition to normal SQL syntax the following conversions are done by the CONFIRMIT recoding engine:

From	To
==	=
&&	AND
	OR
!=	<>
qn.in('a', 'b')	qn IN ('a', 'b')
parseInt(q1)	CASE WHEN ISNUMERIC(REPLACE(CONVERT(varchar, q1),',''))=1 THEN CONVERT(int, REPLACE(REPLACE(CONVERT(varchar, q1),CHAR(10),"),CHAR(13),")) ELSE NULL END

10.1.2 JScript

JScript based variables are recoded by looping through all the variables in a JScript environment. Hence all primitives of the JScript language are available from the expression.

When referencing other questions from the expressions, the calculation engine will create a JScript object based on the value in the database for the current respondent. This object has the following methods available:

Method	Return value
.getValue() .toString() .valueOf()	The ADO value of the ADO field method value(). This is a string except for interview_start and interview_end that return date.
.int()	ParseInt (getValue ())
.str()	New String ((getValue ()))
.date()	New Date (getValue ())
.float()	parseFloat (getValue ())
.int()	ParseInt (getValue ())
.in(param1, param2,...)	True if getValue() equals at least one of the parameters.
.weight()	The weight associated to the value of the field. This is the weight used to report averages.

In the evaluation context, the following functions are also available:

Function	Description
average(p1, p2, ...,pN)	The averages of the parameters: $(p1 + p2 + \dots + pN)/N$ All the parameters are converted by parseFloat() before added. Parameters equaling an empty string are not included in the average calculations at all. The parameter is also ignored in the parameter count N.

Test Answers Properties Preview Parameters Trading Languages Save

Answers

Norwegian	Expression	Precode	Weight
Indoor	apm2 .ln('f', 'w', 'v')	1	
Outdoor	apm2 .ln('w', 'v')	1	

Add Save ☒ Overwrite ☐ Only copy active column

Expression

apm2 .ln('f', 'w', 'v')

Condition builder

Column	Operators and methods	Legal values
apm2	== != < <= > >= && ! ! parseInt(date(average(ln(	Livingroom [l] Bedroom [b] Kitchen [k] In the mountain [m] At sea [s]

Figure 79: Variable definition with condition builder

11 Exports of Data Files

It is possible to export response data to SPSS, MS Excel and ASCII files. The export files will have the question IDs as labels (see 5.2.1.4 for an explanation of labels for different forms.)

For exports without loop variables the export files will have one row per respondent. If the export includes a loop variable, the export files of all formats except SPSS will include one file for each loop. There will be one row for each loop iteration, so there will be several rows for each respondent.

All exports will be sent as compressed files with .zip extension ("zipped") and may be extracted by using standard software like WinZip®.

11.1 Template Editor

You may build templates to choose a subset of data fields for your exports. Instead of choosing all variables when preparing an export, you may choose one of these templates to get the subset you want.

The screenshot shows the 'FIELD CHOOSER TEMPLATE EDITOR' window. The title bar includes the text 'Production database templates.' and a version number 'p0797203'. The interface is divided into two main panes. The left pane, titled 'Template list', contains a list with 'Demographics' (highlighted) and 'Short file'. Below this list are buttons for 'Edit', 'Copy', 'Delete', and 'New'. The right pane, titled 'Template details', is for editing the 'Demographics' template. It contains a 'Template name' field with 'Demographics' entered, a 'Verbatim key field' field with '(default: respid)' shown, and three lists of fields: 'Std fields' (responseid, respid, status, interview_start, interview_end), 'User defined fields' (cars_tested_1 through cars_tested_7, cars, satisfaction), and 'Recoded fields' (Rec_age). Arrows indicate movement between these lists and an 'Export fields' list on the right, which currently contains 'Age'. At the bottom right of the 'Template details' pane is a 'Save' button.

Figure 80: Field chooser template editor

From the template list you may choose to *Edit*, *Copy* or *Delete* an existing template (select template from the list and click the desired action), or make a new template by clicking *New*.

Field	Description
Template name	The name of the template.
Verbatim key field	Open Text questions that have no field width specified (see 5.2.1.4) will not be included in the export files when you export to fixed width ASCII. (Open Text questions initially have no field width limitations, so if it is not specifically applied to them, they could be infinite.) So when exporting to fixed width ASCII, a separate export has to be done for the Open Text questions (see 5.2.1.4 Properties). These files will by default have respid as identification of the respondents, but if the data files include another id you would rather use (e.g. a member id) this may be set in the “Verbatim key field”.
Std fields	These are standard system-generated fields. Field Description responseid Respondent identification given at the time the respondent starts answering the survey. respid Respondent identification given at the time the respondent is added to the respondent list. status Status of the interview. <i>Complete</i> if the respondent has reached the end of the questionnaire (or a STOP node where status is set to complete), <i>screened</i> if the respondent has reached a stop node where status is set to screened, <i>quota_full</i> if the respondent has reached a stop node where the status is set to quota full, and empty if the respondent has not finished the questionnaire. interview_start The time when the respondent entered the questionnaire. If the respondent is allowed to re-enter the questionnaire, it will be the last time the respondent entered the questionnaire. interview_end The time when the respondent finished the questionnaire, either from a STOP node or by reaching the end of the questionnaire.
User defined fields	All the variables from the questionnaire, including hidden questions, but excluding recoded variables.
Recoded fields	All recoded variables from the questionnaire (see 10 Recoding Data).
Export fields	These are the fields that will be included in the exports done with this template.

Table 42: Template details

You move variables from *std fields*, *user defined fields* and *recoded fields* to *export fields* by selecting a variable and clicking > (or < to move it back) or by clicking >> to move all from the field. You may change the order of the variables by adjusting the list with ^ and v. Click on *Save* when you are finished making changes to the template.

11.2 Data Export

DATA EXPORT

p16169441

Export survey production data to ASCII or MS Excel format.

Export properties

Project ID

p16169441

Export format

Excel file

Export template

All fields

Data export

Complete responses

Language

English

Recipient email

john.smith@firm.no

Comment

Start time

ASAP

Submit request

Figure 81: Data export

Export Format	Description
Excel file	MS Excel file in the Office version specified in User Settings (see 3.2). The variables may be split over several worksheets. Excel exports will never be made in Excel 4.0 and older, since these old versions do not support more than one worksheet for each workbook.
Delimited text file, comma separated, quoted	Comma (,) separated ASCII text file. All answers will be inside quotes ("").
Delimited text file, tab separated, not quoted	Tab separated ASCII text file. No quotes ("")
Fixed-width text file	Fixed-width ASCII text file. Will give an export file of single, multi and grid questions. If Open Text questions have field width specified (see 5.2.1.4), they will be included as well. Exporting Open Text questions where the field width has not been specified can be done by selecting "Open Text Questions", The field widths will be equal to the field widths from properties of the questions if provided or else to the default of the database. The fields with numeric responses will be zero-filled, so e.g. a precode 1 with a field width of 3 will be reported as "001".
Definition file for fixed-width text file	Will generate a MS Excel file with a description of the exported fixed-width text file. (The information provided may be useful for other exports as well.)

	Field	Description
	Form ID	
	The Question (form) ID	
	Var ID	
	The variable name. Form ID and Var ID will be equal for Single and Open Text questions, but will differ for Grids and Multi questions. See 5.2.1.4 for an explanation of the variable names of multis and grids.	
	Start Col	
	The start position of this variable in the export file, counting characters from left to right.	
	End Col	
	The end position of this variable in the export file, counting characters from left to right.	
	Punch	
	Will list the possible punches (precodes) possible for these answers corresponding to the texts in the Answer Text field.	
	Question Text	
	The text of the question.	
	Answer Text	
	The text of the answer. (Corresponding to Punch).	
	Data Type	
	Question type (or system generated type).	
	Level ID	
	Loop level. Gives the id of the loop this question is inside.	
Open-end questions	Open-end questions are all open questions (Open Text questions, Other, specify from answer lists or Multi questions with Open Text property selected) that do not have a field width specified (see 5.2.1.4). They will be included in the normal export of the tab- and comma-separated text files and the Excel files, but not	

	in the fixed width text files. If you choose "Open-end questions" you will receive a "fixed-width" ASCII text file with three columns: Respondent ID, Question ID and that respondents answer to that Open-end question. The respondent ID will by default be respid, but you may choose a different ID by selecting a different "Verbatim key field" in your template (see 11.1)
--	---

Table 43: Export formats

Export property	Description
Project ID	The system generated ID of the project
Export data	Choose if you want your export to include all responses, or just the ones with status "complete".
Export template	All fields or a user-defined template for a subset of the fields (see 11.1)
Export language	Select which language Question titles and labels should appear in. Choose only from active languages.
Recipient email address	The export file is sent to this mail address.
Start time	Specifies when the report is to be put in the task list. Either immediately, by midnight or at a specific time.

Table 44: Data export properties

Click *Submit Request* to send the request to Task Management (see 3.4), or, if you have chosen Schedule for later execution, to enter Recurrence pattern settings.

Start time

☐ Disable task (clear queue)

☐ ASAP

☒ 0:00 30 9 2000

Recurring task

Yes

Recurrence pattern type

☐ Hourly

☐ Daily

☐ Weekly

☐ Monthly

☒ By data change

Range of recurrence

☒ No end date

☐ End by: 30 12 2000

By data change

☒ Count 1

☐ Percentage

Save

Figure 82: Recurrence pattern

11.3 SPSS Export

SPSS EXPORT

Export survey data to SPSS.

p16169441

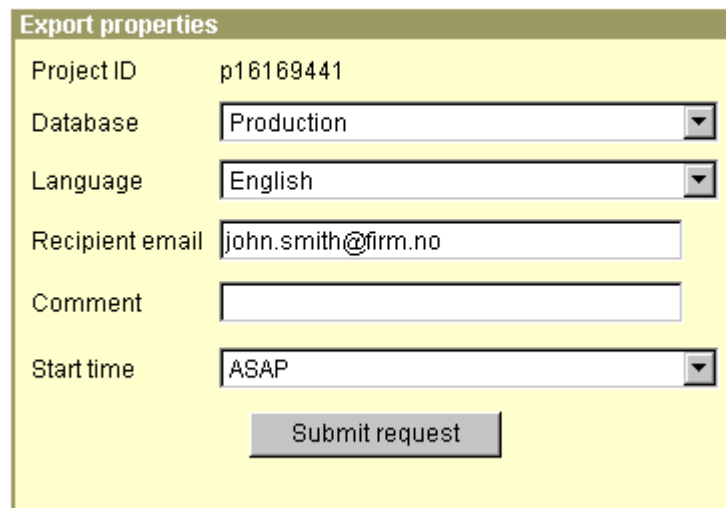
The image shows a web-based form titled "Export properties" with a yellow background. It contains several input fields: "Project ID" with the value "p16169441", "Database" with a dropdown menu showing "Production", "Language" with a dropdown menu showing "English", "Recipient email" with the text "john.smith@firm.no", "Comment" with an empty text box, and "Start time" with a dropdown menu showing "ASAP". At the bottom of the form is a grey button labeled "Submit request".

Figure 83: SPSS Data Export

SPSS Export will generate two files: A SPSS file (with the extension .sav) and an ASCII text file with a description of the Variable-ID mappings. SPSS variable IDs are restricted to a length of 8 characters. Question IDs that are longer than 8 characters will be replaced with IDs of the format "@n" where n is a number. The text file has a list of all the variables, their name in CONFIRMIT and their corresponding names in the SPSS file.

SPSS Export property	Description
Report data	The data set you want an export from. Either "Production data" or "Test data"
Report language	The SPSS file will include the question text and labels from the questionnaire in the language selected here. Choose only from active languages.
Recipient email address	The export is sent to this mail address.
Batch job comment	This comment will appear in the comment field of the Task Management lists. Use it to distinguish between reports.
Start time	Specifies when the report shall be put in the task list. Either immediately, by midnight or at a specific time.

Table 45: SPSS Export properties

Click *Submit Request* to send the request to Task Management (see 3.4), or, if you have chosen Schedule for later execution, to enter the Recurrence pattern settings.

11.4 Report Export

For a description see 9 Reporting.

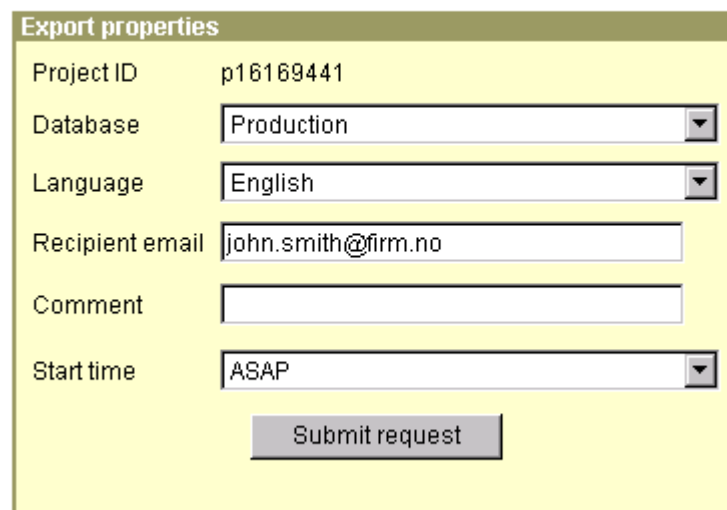
11.5 MS Word Export

You can also export your questionnaire to MS Word format. This is currently beta functionality, but it has been added because of the large number of requests. **This will export the last compiled version of the questionnaire.** It does not include any kind of skip logic.

Note that you have to have compiled a database as well as chosen the same language user settings (see 3.2) as in *Report language* in order to be able to export your questionnaire into MS Word format.

Comment: This is unsupported functionality, meaning that we do not currently bug fix this feature.

MS WORD EXPORT (BETA) _____ p16169441 —
Export questionnaire to MS Word format (last compiled version).



The screenshot shows a web form titled "Export properties" with a yellow background. It contains the following fields:

- Project ID: p16169441
- Database: A dropdown menu showing "Production".
- Language: A dropdown menu showing "English".
- Recipient email: A text input field containing "john.smith@firm.no".
- Comment: An empty text input field.
- Start time: A dropdown menu showing "ASAP".
- A "Submit request" button at the bottom.

Figure 84: MS Word Export

Click *Submit Request* to send the request to Task Management (see 3.4), or, if you have chosen Schedule for later execution, to enter the Recurrence pattern ssettings.

12 APPENDIX A: confirmit Script Reference

12.1 Accessing Survey Variables

f function
Description Accesses survey variable values.
Declaration <code>f(qID, [expr1[, expr2, ...]])</code>

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

`qID` is the ID of the form, as specified in the form's properties, where the variable(s) in question are declared.

`[expr1[, expr2,...]]`: none or more expressions used to identify a specific iteration if the question is defined inside a loop. Within loops, tables of variables are associated with each question instead of the single item normally used. Each expression should evaluate to a valid loop iterator value, starting with the value of the innermost nested loop. If the number of expressions provided is less than the current loop nesting level, then the current values will be used for the loop iterators of the enclosing blocks.

Say that you have a question – `q1` – placed inside a loop over the set `{"1", "2", "3", ...}`.

During execution of loops, the default values of `[expr1[, expr2,...]]` are set to the current values of each loop's iterator variable. In our case, this means that the default value for `expr1` will be "1", "2", etc. If you want to refer to `q1`'s value within the current iteration you simply use:

```
f("q1")
```

Call

```
f("q1", "1")
```

```
f("q1", "2")
```

etc..

if you need to refer to answers given in a specific iteration. Explicit qualification is only necessary if you need to refer to an answer given in a previous iteration or when execution flows past the loop where the question is defined.

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.

12.1.1 Variable Types

confirmit currently provides support for three types of variables:

- *Coded variables* are assigned distinct values from a pre-coded domain of alternatives. In **confirmit** users enter the 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, which denote a positive or a negative response respectively. **confirmit** uses checkboxes for input of dichotomies. These variables are declared in **multi** forms.
- *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 and/or Ordered property for a **multi** form.

Compounds group one or more variables that are defined in the same form. In **confirmit** a compound is defined for each multi or grid form.

Type information is available at run-time by examining an object property. For each variable type a different property is defined:

Property	Description
x.CODED	Object represents a coded variable.
x.OPEN	Object represents an open variable.
x.DICHOTOMY	Object represents a dichotomy.
x.COMPOUND	Object represents a compound.

Example:

```
function foo(qID)
{
    var v = f(qID);
    if (v.CODED)
        return "" + v.value();
    else if (v.COMPOUND)
        return "" + v.categories();
}
```

12.2 Useful functions and methods

This section describes some functions and methods that are useful when you are building questionnaires. The functions and methods can be used in validation codes, in precode masks, in conditions or in script elements. Remember that the examples only show you one possible use of the function.

12.2.1 Basic variable objects

12.2.1.1 .get()

Returns the associated value (precode). Most often the `f('qid')` function will return exactly the same as `f('qid').get()`. However, for ordered multi questions it is necessary to use the `.get()` method.

Example: You want to ask your respondents a question about the alternative they have ranked as number one in an earlier ordered multi question. Then you may add this generic function in a script node:

```
function rankedfirst(qid){
    var strfirst;
    strfirst = "";

    var precodes=f(qid).domainValues();
    for(var i=0; i<precodes.length; ++i) {
        var precode=precodes[i];
        if (f(qid)[precode].get() == "1"){
            strfirst = f(qid)[precode];
        }
    }
    return strfirst;
}
```

In the precode mask property you call this function.

12.2.1.2 .set()

The `.set()` method sets one object equal to another object. Notice that the `.set()` method is *not* the same as the `set()` function, described in 12.2.7.2

Example:

```
f("qID1").set(f("qID2"))
```

12.2.1.3 .label()

Returns the label in the current language.

12.2.1.4 .value()

Similar to `.get()`. Returns the value (precode)

12.2.1.5 .valueLabel()

Returns the label (text from answer list) in the current language.

12.2.1.6 .values()

Returns an array of the values of each item in the answer list.

12.2.1.7 .categories()

Returns an array of all precodes (from the answer list) with an answer (checked in multi/answered in grid).

12.2.1.8 .categoryLabels()

Returns an array of labels (text from Answer list) associated with each category in the current language

12.2.1.9 .domainValues()

Returns an array of all valid values (precodes) for coded variables.

Returns an array of all categories (the precodes from the answer list) for compounds.

12.2.1.10 .domainLabels()

Returns an array of all labels (texts from answer list) in the current language.

12.2.1.11 Some practical use

.get(), .value(), .values() and .categories() are used to ensure that you get the values stored in the database returned. In most contexts you wouldn't need to use it, because f("qID") will give you exactly what you want, but you may e.g. sometimes want the precodes or the values answered to be piped in instead of the labels (in ^ mode in the question text).

When you use ^f("qID")^, by default you will get the labels piped in for all question types other than open text questions, where you get the value answered. But perhaps (for some reason) you for example want the precode (or precodes) to be piped in instead. Then you may use these functions.

.value() gives the value stored for that field. For f("qID").value() on a single question it will be the precode of the answer, for an open text question it will be the text answered, for f("qID")["precodeID"].value() on a multi question it will be 1 or 0 (or nothing) depending if the answer with precode "precodeID" has been selected or not (or the question has been asked), and on a grid it will be the precode of the answer to the item from the answer list with precode "precodeID".

f("qID").values() gives an array of the values of question qID. If it is a multi question it will be an array with 1's and 0's depending on which items is selected, if it is a grid it will be the precodes of the answers.

f("qID").categories() gives an array of precodes of items that are answered on question qID (where .toBoolean() returns TRUE). For a multi this will be all items that are selected. For a grid it will be all items that are answered (normally in a grid all items are required, so this will return an array with all precodes from the answer list).

f("qID").domainValues() gives an array of all the precodes in the answer list (all possible answers/precodes). This may be very useful in scripting when you want to do some checking on all elements in a multi or a grid (e.g. to set a precode mask based on the answers of a grid or to copy a multi question into a hidden multi variable - or something like that.)

Example: Use this function to copy a multi into another:

```
function copymulti(from,to)
{
  var precodes=f(from).domainValues()
  var i;
  for(i in precodes)
  {
    f(to)[precodes[i]].set(f(from)[precodes[i]])
  }
}
```

.valueLabel(), .domainLabels(), categoryLabels(), .domainLabels() are all functions to return the variable label(s) on functions, either the label of a specific answer (.valueLabel()) or the labels of all possible answers (.domainLabels() on a single question, .categoryLabels() on a multi or grid for the answer list and .domainLabels() on a grid for the labels of the scale). They are used if you need the labels outside the ^ mode, in script nodes and validation code.

12.2.2 Arithmetic functions

12.2.2.1 Sum()

Returns the sum of its arguments. Valid argument types are numbers, arrays or any kind of variable object.

Example: You have a multi question with OpenText selected and want the respondents to enter a percentage that sums up to 100%. Include this script in the Validation code property:

```
if (Sum(f('qID'))!=100){
  RaiseError();
  SetErrorMessage(LangIDs.en, "Your responses added up to " +
Sum(f('qID')) + ", they must add up to 100");
}
```

12.2.2.2 Count()

Returns the number of arguments.

12.2.2.3 Average()

Returns the average of its arguments.

12.2.2.4 Max()

Returns the maximum of its arguments.

Example: You ask the respondents how much time in hours they spend watching some TV channels a week in a multi question with OpenText selected. You want to ask a follow-up question if some TV channel(s) is(are) watched for more than 15 hours a week.

Include this condition:

```
IF Max(f('qID')) > 15 THEN...
```

12.2.2.5 Min()

Returns the minimum of its arguments.

12.2.3 Conversion

12.2.3.1 .toString()

The .toString() method converts the variable to a string.

12.2.3.2 .toNumber()

The .toNumber() method converts the variable to a number (see 6.3.3 for details).

12.2.3.3 .toBoolean()

The .toBoolean() method converts the variable to a Boolean (TRUE/FALSE). If you use an expression like this in a validation code:


```
if(f("qID").toBoolean())
```

-and qID is a single question, it would be TRUE if it is answered, FALSE otherwise
-and qID is a open text question, it would be TRUE if it is answered, FALSE otherwise
-and qID is a multi question, it would be TRUE if one or more of the checkboxes is selected, FALSE otherwise
-and qID is a grid, it would be TRUE if one or more of the elements in the answer list has an answer, FALSE otherwise (so there might be some answers missing in some lines)

If you use

```
if(f("qID ")[ "1" ].toBoolean())
```

-and qID is a multi question, it would be TRUE if the element with precode "1" is selected, FALSE otherwise
-and qID is a grid question, it would be TRUE if the element from the answer list with precode "1" is answered, FALSE otherwise

Example: You have a multi question (q1) and want the respondents to select at least one answer. By default you don't have to select any answers in a multi question, even with the Answer Required test selected (see 7.2.3.4). You have to include a script like this in q1's validation property:

```
if (!f('q1').toBoolean()){  
    RaiseError();  
    SetErrorMessage(LangIDs.en, "You must provide at least one  
answer!");  
}
```

12.2.4 Classification functions

12.2.4.1 IsNumeric(arg)

Returns true if the argument is numeric.

Example: You have a multi question with the OpenText property selected and want the first answer input to be numeric.

```
if((!IsNumeric(f("qID")["1"])) && (f("qID")["1"] != "")){  
    RaiseError();  
    SetQuestionErrorMessage(LangIDs.en, "Not a valid number! Please  
correct!");  
}
```

In this example you don't want the user to get an error message when the first answer is left empty. That's why we have included the second comparison.

Note that you could use the property Numeric as an alternative (see 5.2.1.4 for details)

12.2.4.2 IsInteger(arg)

Returns true if the argument is an integer.

12.2.4.3 IsDateFmt(arg, format)

Determine whether the provided argument is a valid date according to format.

Within a date format, the following character sequences have special significance:

Y: Year, four digits.

YY: Year, two digits.

YYYY: Year, four digits.

- M: Month number, one or two digits.
- MM: Month number, exactly two digits.
- D: Day number, one or two digits.
- DD: Day number, exactly two digits.

All other characters are treated as separators.

All parts are optional. If omitted, then the system date is used to determine month and year and the number 1 is used for the day.

Century interpolation for 2-digit years is handled with a break-point value of 10: values between 0-10 are interpreted as the years 2000-2010, while 11-99 are treated as 1911-1999.

IsDateFmt returns an object with the properties day, month, and year if the date is valid, null otherwise.

12.2.4.4 IsDate(day, month, year)

Determines whether the provided year, month, day combination constitutes a valid Gregorian date.

The month and year arguments are optional. If they are not provided then the current month and/or year is used.

Century interpolation for 2-digit years is handled with a break-point value of 10: values between 0-10 are interpreted as the years 2000-2010, while 11-99 are treated as 1911-1999.

IsDate returns an object with the properties day, month, and year if the date is valid, null otherwise.

12.2.4.5 IsEmail(arg)

Checks whether arg has the format of a valid e-mail address.

Note: the function does not attempt any name lookups or address verification.

12.2.4.6 IsNet(quadIP, quadNet, quadMask)

Determines whether an IP address belongs to an Internet network.

Arguments:

- quadIP: The IP address, on quad form (X.X.X.X)
- quadNet: The network number, on quad form.
- quadMask: An optional mask, on quad form, if subnetting is used.

If no mask is provided then the number of bits that constitute the network part of the address is determined from the address class type. In this case the function returns true if the IP address' network number is identical to the supplied network number and it has a non-zero local address. Otherwise it returns false

If a mask is provided then the function returns true if the IP address' network and subnet parts are identical to the one supplied and the host number part is non-zero. Otherwise it returns false.

12.2.5 Context information

12.2.5.1 CurrentID()

Returns the current respondent ID, may be used to arbitrate between blocks of questions, etc.

12.2.5.2 CurrentPID()

Returns the ID of the project, e.g. p1234567.

12.2.5.3 CurrentSID()

Returns the s ID of the current respondent. Each respondent is identified by two parameters in the url; r and s.

12.2.5.4 CurrentLang()

Returns current language used.

Example:

```
f('link') = 'http://www.confirmit.com/wi/' + CurrentPID() + '/i.asp?r=' + CurrentID() + '&s=' + CurrentSID() + '&l=' + CurrentLang()
```

f('link') would then be something like:

```
http://www.confirmit.com/wi/p1234567/i.asp?r=214&s=KTQWXZF&l=0
```

12.2.5.5 InterviewStart()

Returns the respondent's interview start time.

12.2.5.6 InterviewEnd()

Returns the respondent's interview end time.

12.2.5.7 GetStatus()

Returns the current interview status.

12.2.5.8 SetStatus()

Sets the current interview status.

12.2.5.9 Forward()

Returns true if moving forward through the questionnaire.

12.2.6 Browser information

12.2.6.1 BrowserType()

Returns the browser type, as detected by the MSWC.BrowserType component.

Example: Together with the function BrowserVersion()(see under), this information can be included as hidden variables instead of having the respondents to enter this information manually in a question themselves (see 6.7 for an example).

12.2.6.2 BrowserVersion()

Returns the browser version, as detected by the MSWC.BrowserType component (see 6.7 for an example).

12.2.6.3 RequestIP()

Returns the IP address which the request originated from as a string on quad IP form.

12.2.7 Handling sets

12.2.7.1 a('qID')

`a("qID")` returns the complete set of precodes defined in the answer list belonging to qID, no matter if they are answered or not.

12.2.7.2 set()

`set("precode", "precode", ...)` returns a set consisting of the precodes you specify (see 6.7 for an example). Notice that the `set()` function is *not* the same as the `.set()` method (see 12.2.1.2 for a description of the `.set()` method).

12.2.7.3 nset()

`nset(n)` returns a set populated with the members 1, 2, 3,...,n.

12.2.7.4 nnset()

`nnset(m, n)` returns a set with values from m to n: m, m+1,..., n-1, n (see 6.4.2 for an example).

12.2.7.5 .members()

Returns an array of all the set's members. Typically used for iteration over the set's members.

12.2.7.6 .union()

`l.union(r)` returns a new set that is the union of l and r (see 6.4.3 for details).

12.2.7.7 .isect()

`l.isect(r)` returns a new set that is the intersection of l and r (see 6.4.4 for details).

12.2.7.8 .diff()

`l.diff(r)` returns a new set that is the difference between l and r (see 6.4.5 for details).

12.2.7.9 .size()

The size method returns the number of selected precodes in a multi question.

As an alternative to the size method you can use the `Sum()` function (see earlier in this chapter for a description).

Example: You have a multi question and want to force the respondent to select exactly 3 answers. Then select the not required property for that multi question and add this script in the validation code property:

```
if(f("qID").size()!="3")
{
    RaiseError();
    SetQuestionErrorMessage(LangIDs.en,"You must select exactly 3 answers!");
}
```

12.2.7.10 .inc()

The include method is used to check if a set includes some specified precodes.

Example: The example shows a condition node where we check if a question does *not* include the precode "99". (could be an answer like "None of the above").

```
IF !f('qID').inc(['99']) THEN...
```

This is a work around for the lack of a GOTO functionality in confirmit. The example is an alternative to `IF f('qID').inc(['99']) THEN GOTO...` You have to rephrase your GOTO logic.

12.2.7.11 .add()

The .add() method is used to add elements to a set (see 6.7 for an example). When working with the f('qID')-function you should use the .union(set(...)) method instead (see 6.4.3 for details).

12.2.7.12 .remove()

The .remove() method is used to remove elements from a set. When working with the f('qID')-function you should use the .diff(set(...)) method instead (see 6.4.5 for details).

12.2.7.13 Filter(qID,prefix)

Returns a set of precodes whose labels are prefixed by the supplied prefix. The Filter(qID,prefix) is useful when you have long answer lists.

Example: You ask the respondent to enter the first few characters in an Open question (q1). Then you include a precode mask in the next question (q2):

```
Filter('q2', f('q1'))
```

Now, the answer alternatives in q2 will only include the ones with the supplied prefix, entered by the respondent in q1.

12.2.8 General utilities

12.2.8.1 SendMail(from, to, subject, body)

SendMail can be used to send an email-message using CDONTS

Arguments:

from: Message sender
to : Recipient's address.
subject: Message subject line
body: Message body

Example: This is an example used in a script node at the end of a questionnaire for registering to a seminar.

```
SendMail ("marketing@confirmit.com",f("contact")["e-mail"],  
"Confirmation - Expo Media 2000",  
"Dear "+f("contact")["name"]+", "+"\\n"+"\\n"  
+"Thank you for visiting our microsite and registering for the Expo  
Media 2000. This is a confirmation that you wish to attend this  
seminar:"+ "\\n"+"\\n"  
+f("seminar").label()+"\\n"+"\\n"  
+"Best Regards, "+"\\n"  
+"Marketing Executive"+"\\n"  
+"Future Information Research Management (FIRM) AS"+"\\n"+"\\n"  
+"-----"+"\\n"  
+"-- Email powered by confirmit --"+"\\n"  
+"-----"  
)
```

In the example information about the recipient is collected from a multi question with the open text property selected and information about the seminar is collected from a single question. “\n” means a line break

12.3 Form Validation

12.3.1 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 detailed 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.

12.3.1.1 A typical Validation Code

A typical validation code would consist of a condition with an expression that evaluates to TRUE for some type of error situation where you want the respondent to be unable to move to next page and you want an error message to be displayed:

```
if(expression)
{
    RaiseError();
    <some function(s)setting the text of the error message(s)>
}
```

12.3.1.2 RaiseError()

The RaiseError function 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.

12.3.1.3 Error Messages

When one or more questions on a page have an error, an error message will be displayed on top of the screen informing the respondent that one or more of the questions on the page has errors (Default: "Please answer all questions properly. Please review all questions on this page."), and more detailed error messages will be displayed above the question(s) where the error(s) occurred.

Please answer all questions properly. Please review all questions on this page.

Please enter consecutive answers in the range 1 to 3.

Rank

Please rank the channels you watch on a regular basis, 1 being the one you like best.

ABC

CNN

NBC

Figure 85: Error messages

A number of functions are provided to manipulate the error messages:

ClearErrorMessage() – clear the error message at the top of the page (will remove the default error message).

SetErrorMessage(langID, message) – set the error message at the top of the page (will replace the default error message).

ClearQuestionErrorMessage() – clear the error message for the current question (will remove the default question error message).

AppendErrorMessage(langID, message) – appends *message* to the current page's error message.

SetQuestionErrorMessage(langID, message) – sets the error message for the current question.

AppendQuestionErrorMessage(langId, message) – appends *message* to the current question's error message.

langID is a language identification. When conducting multilingual surveys you should provide one message for each of the languages used for interviewing. The runtime provides an object **LangIDs** which maps language codes to the internal values of **confirmit**. The codes used are specified in the international standard ISO 639, and are listed in Appendix B. E.g. **LangIDs.en** is English, **LangIDs.fr** is French and so on.

message is a string value with the error message you want to display. This could be a text inside quotes, a Jscript string variable or you may use the **ErrorTemplate** function (below).

12.3.1.4 Some default functions for Conditions in Validation Code

The **confirmit** runtime provides a number of functions that may be used in the conditions of your validation code:

QuestionErrors() - returns TRUE if an error has been raised during validation, FALSE otherwise

MissingRequiredError() – returns TRUE if one or more required answers to a question are missing, FALSE otherwise

ExclusivityError() – returns TRUE if an exclusive answer ("single punch") in a multi has been selected together with any other alternative, FALSE otherwise

SizeError() – returns TRUE if one or more open-ended answers exceeded the maximum length specified in field width, FALSE otherwise

NotSpecifiedError() – returns TRUE if an alternative in an Other-Specify construct has been selected/answered without filling in the associated "Specify" text box, FALSE otherwise.

NotSelectedError() – returns TRUE if a "Specify" value has been entered in the text box of an Other-Specify construct without selecting/answering the associated alternative, FALSE otherwise.

RankError() – returns TRUE if the "Ordered" property has been selected for the form and the answers to the questions are non-numeric, do not start at 1 or are non-consecutive, FALSE otherwise.

NumericError() – returns TRUE if the "Numeric" property has been selected and the answer is not numeric, FALSE otherwise.

PrecisionError() – returns TRUE if the "Numeric" property has been selected and the answer can not be stored within the defined precision, FALSE otherwise.

ScaleError() – returns TRUE if the "Numeric" property has been selected and the answer contains more decimals than in the defined scale, FALSE otherwise.

RangeError() – returns TRUE if the "Numeric" property has been selected and the answer is outside the defined range, FALSE otherwise.

Example:

```
if (ExclusivityError())
{
    AppendQuestionErrorMessage(LangIDs.en,
    ErrorTemplate("^DEF_EXCL^ cannot be selected with any other
    alternative. "))
}
```

(Notice that you do not need to use RaiseError(). When ExclusivityError() is TRUE this situation is already marked as a error situation by standard validation).

12.3.1.5 Template Based Error Messages

The system allows you to use templates for your error messages in custom validation. For each error type, various elements are filled in that can improve the information content of the messages you report to the end users. Instead of simply using fixed strings for error messages, the **ErrorTemplate** function can be used to "plug in" information about the current question.

The syntax of ErrorTemplate is **ErrorTemplate(spec)** where spec is a string with the template specification. Inside the spec you may use different elements that are singled out with caret (^) as pre- and suffix, e.g. ^MISSING^.

Example:

```
AppendQuestionErrorMessage(LangIDs.en, ErrorTemplate("^MISSING^ has
not been answered. "))
```

will append the text "*label(s)* has not been answered" to the error message.

Templates are available for all the standard validation.

Missing required

The following template elements are defined when **MissingRequiredError()** returns true:

MISSING	Labels of all questions that lack answers, comma separated.
MISSING_AND	Same as above, but with the word "and" separating the two last items.
MISSING_OR	Same as above, but with the word "or" separating the two last items.

Example of use:

Please select an answer for ^MISSING_AND^.

Exclusivity tests

The following template elements are defined when `ExclusivityError()` returns true:

SEL_EXCL	The labels of all exclusive (single punch) answers that were selected.
SEL_EXCL_AND	Same as above, but with the word “and” separating the two last items.
SEL_EXCL_OR	Same as above, but with the word “or” separating the two last items.

SEL_NEXCL	The labels of all non-exclusive (multi punch) answers that were selected.
SEL_NEXCL_AND	Same as above, but with the word “and” separating the two last items.
SEL_NEXCL_OR	Same as above, but with the word “or” separating the two last items.

DEF_EXCL	The labels of the defined set, i.e. the unmasked options, of exclusive (single punch) members in the inspected set.
DEF_EXCL_AND	Same as above, but with the word “and” separating the two last items.
DEF_EXCL_OR	Same as above, but with the word “or” separating the two last items.

DEF_NEXCL	The labels of the defined set of non-exclusive (multi punch) members in the inspected set.
DEF_NEXCL_AND	Same as above, but with the word “and” separating the two last items.
DEF_NEXCL_OR	Same as above, but with the word “or” separating the two last items.

Example of use:

Please do not check ^SEL_EXCL_OR^ if you check other answers to the question.

Other-Specify verification

The following template elements are defined when `NotSpecifiedError()` or `NotSelectedError()` returns true:

OTHER	Label of the alternative/input that requires specification in an Other-Specify construct.
SPEC	The specification

Examples of use:

If you specify ^OTHER^ then please select the ^OTHER^ option.

Please specify if ^OTHER^ is chosen.

Answer size tests

The following template elements are defined when `SizeError()` returns true:

TOO_LONG	The labels of the inputs that were too long.
TOO_LONG_AND	Same as above, but with the word “and” separating the two last items.
MAX_SIZE	The maximum allowed size.

Example of use:

The answers to `^TOO_LONG^` were longer than `^MAX_SIZE^` characters and have been truncated. Please review.

Rank order tests

The following template elements are defined when `RankError()` returns true:

RANK_MIN	The label of the first member of the ranking scale, or 1 if the question is open ended.
RANK_MAX	The label of the nth member of the ranking scale, or n if the question is open ended, where n is the number of items to rank.

Example of use:

Please enter consecutive answers in the range `^RANK_MIN^` to `^RANK_MAX^`.

Numeric error tests

The following template elements are defined when `NumericError()` returns true:

NUMERIC_ERRORS	The labels of all elements with a numeric error.
NUMERIC_ERRORS_AND	Same as above, but with the word “and” separating the two last items.

Example of use:

Please enter these answers in a numeric format: `^NUMERIC_ERRORS^`.

Precision error tests

The following template elements are defined when `PrecisionError()` returns true:

PRECISION	The defined precision
PRECISION_ERRORS	The labels of all elements with a precision error.
PRECISION_ERRORS_AND	Same as above, but with the word “and” separating the two last items.

Example of use:

Please enter answers within the precision `^PRECISION^`.

Scale error tests

The following template elements are defined when `ScaleError()` returns true:

SCALE	The defined scale
SCALE_ERRORS	The labels of all elements with too many decimals.

SCALE_ERRORS_AND	Same as above, but with the word “and” separating the two last items.
------------------	---

Example of use:

Please enter answers with no more than ^SCALE^ decimals.

Range error tests

The following template elements are defined when `RankError()` returns true:

RANGE_MIN	The label of the defined minimum value.
RANGE_MAX	The label of the defined maximum value.
RANGE_ERRORS	The labels of all elements with to many decimals.
RANGE_ERRORS_AND	Same as above, but with the word “and” separating the two last items.

Example of use:

Please enter answers in the range ^RANGE_MIN^ to ^RANGE_MAX^.

13 APPENDIX B: Online Coding Tool

13.1 Why Have a Coding System?

The coding system is a tool for coding open-ended questions in order to give the answers pre codes and code frames.

This system has been designed to enable coders to classify open-ended or "open-text" questions into categories, thereby having the ability to report on these questions in the same way as for single and multi questions. Open ended or "open text" questions (forms) are here defined as *Open* questions, *Multi* questions with "open-text" checked off in *Properties*, and any question where the "other" functionality is enabled in the answer list of the question. An example for how one can make use of online coding can be seen in the following report images. In this example, the respondents were asked in an open-end question to provide answers describing what types of sports they liked playing. Through use of the online coding, coders are able to take the open-end answers and code them into specific categories thus making this information available for reporting purposes.

The report below shows the respondent's answers from the open-ended questions. The table and chart on the following page, show the how the answers can be used in reporting after having been recoded.

Which sports do you like playing?

Favorite sports
I love to swim and relax.
Football, space walking, jogging, tennis
Bowling
I like to play football
Soccer and skiing
I love to ski
I like to play soccer. I also like something similar to broom ball.
I like to play handball, and a bit of basketball
I like golf and swimming
I like playing basketball and golf
Sailing

Figure 86: Respondents answers for the open-ended question before coding

Which sports do you enjoying playing in your freetime?

	Sports					
	Pos		Neg		Total	
Hockey	1	5.9%	16	94.1%	17	100.0%
Tennis	2	11.8%	15	88.2%	17	100.0%
Football	2	11.8%	15	88.2%	17	100.0%
Basketball	2	11.8%	15	88.2%	17	100.0%
Cricket	1	5.9%	16	94.1%	17	100.0%
Other	14	82.4%	3	17.6%	17	100.0%

Figure 87: A Sample example

The sample table above was produced in reporting, using the recoded open-end answers as displayed in the table on the previous page. As you can see, the answers not belonging to a specific category were coded as "other", and the answers, which fit into one of the available categories such Hockey, Tennis, Football... were given codes accordingly.

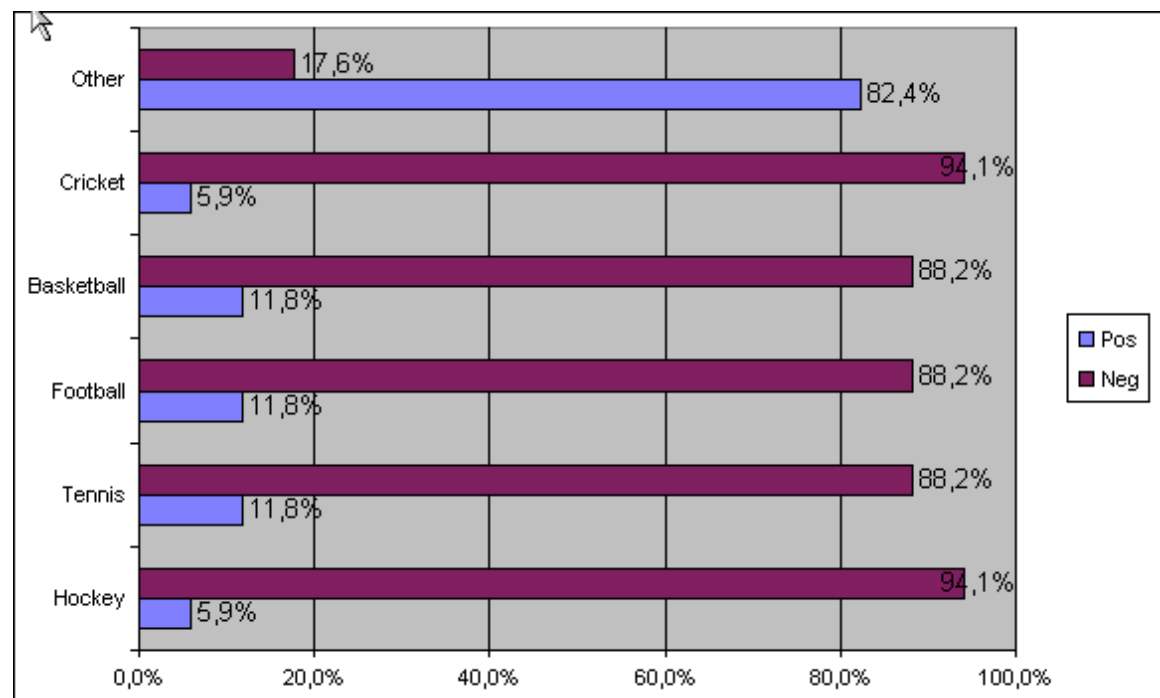


Figure 88: This Chart was produced using open-ended text answers after the online coding was complete

13.2 Specification Summary

This chapter provides a detailed overview of how the new coding system of open-ended questions operates. The main features of the system are:

- For each open-ended question a multi question is required in order to define the available codes for the question. Specifying the corresponding open-ended question in the properties for the multi question links these two questions. This is explained in section 13.3 - "Changes / additions to ConfirmIT".
- A permission control panel is used to show the status of the questions, and to lock and unlock questions. The basic idea is to have multiple accesses to one survey and single access to one question within this survey. This part of the system is described in section 13.5 - "Locking the Questions for Coding".
- The main window of the coding system is the Coding Window where the coders can define a working set of answers for a chosen open-ended question, code each answer, and navigate through the working set of answers. The functionality of the coding system is described in section 13.6 - "Coding Window".

13.3 Changes/Additions to confirmit

13.3.1 General Additions

13.3.1.1 New Menu Item - Coding

A new menu item called "Online Coding" has been added to the main menu in confirmit. When choosing this menu item the window described in section 13.5 opens.

13.3.1.2 User Settings - 2 new items

Item 1: Coding screen size: This is the maximum number of answers, which can be displayed on one screen, while coding. The size will be set to 10 as default the first time the coding tool is used, but can be changed at any time, by entering the desired number and pressing update settings to save

Item 2: Working set size: This is the maximum number of answers for a given working set. The size will be set to 1000 as default the first time the coding tool is used, but can be changed at any time, by entering the desired number and pressing update settings to save

User key:	PRDXSLJAIRKXBNUVEHVJCWSE
First name:	Adminstrator
Last name:	Aministrator
E-mail:	administrator@confirmit.com
Language:	English
Display:	1024 x 768
User level:	Advanced
Microsoft Office version:	Newest version
Chart component:	Excel
Coding Screen Size:	10
Working set size:	1000
Answerlist font:	
Update settings Change password	

Figure 89: User Settings window

Object Permissions

The project administrator must define who will have the permission to code the open-ended questions. This is done in the Object Permissions window, found under User settings. It will appear similar to the following image.

-	R	W	D	Administrate project	Coder of opentext questions
☒	☐	☐	☐	☐	☐
☒	☐	☐	☐	☐	☐
☒	☐	☐	☐	☐	☐
☒	☐	☐	☐	☐	☐
☒	☐	☐	☐	☐	☐
☒	☐	☐	☐	☐	☐
☒	☐	☐	☐	☐	☐
☒	☐	☐	☐	☐	☐
☒	☐	☐	☐	☐	☐
☒	☐	☐	☐	☐	☐

Figure 90: Object Permissions

13.4 One Multi for Each Open-Ended

For each open-ended question, which is to be coded, a new multi question must be added. This is done in the same way as before, by dragging a new multi question to the routing of the project.

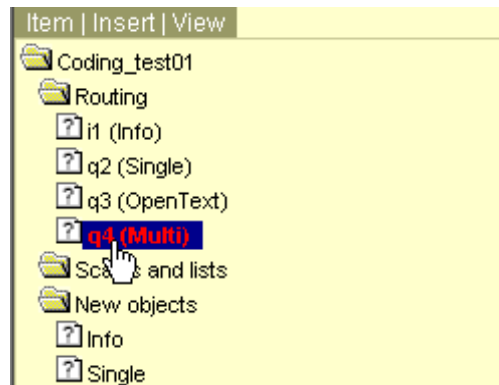


Figure 91: Multi question for online coding

This question works like other multi questions, i.e. answer alternatives and corresponding precodes are to be defined in the Answers window for the question. This is shown in the window below. The example shows 7 codes, but it is of course possible to define any number of codes.

q6 - Holiday activities

Text **Answers** Properties Preview Tracking Languages Save

Answers

English	Precode	Weight	ColWidth	BgColor	Punch	KeepPos	Other
Skiing	1						
Culture, opera, ballet	2						
Hiking	3						
Adventure	4						
Water sports	5						
Computer/playstation	6						
Exhibitions	7						

Add Add predefined Clear Delete rows Save

Figure 92: Defining precodes for online coding

Once the answer alternatives and precodes have been defined under the answer menu, select *Properties* in order to define which questions are to be linked. As you can see from the following figure, there are two new fields ("open text coding" and "coding of "). Select the "open text coding" field and the new field "coding of" will appear. It is in this field that the qID of the open-ended question is to be entered. This will now create a link to the chosen open-ended question. In addition, remember to tick off hidden, in order to, hide the multi question from the respondent.

Properties

Question type **Multi**

Created **20. september 2000 11:04:34 GMT+02:00**

Modified **20. september 2000 15:59:13 GMT+02:00**

Question ID

Field width

List Rows List Columns

Precode mask

Randomize ☐

Recoded var ☐

Background var ☐

Hidden ☒

Not required ☐

Ordered ☐

Opentext coding ☒ Coding of

Figure 93: Defining the multi question for online coding

Question (form) type	qID to be typed in the “coding of” field:	Remarks:
Open	qID	
Multi with “Open Text” checked off in <i>Properties</i>	qID _precode	In example: q5_3 will let you recode the answer alternative with precode 3 from the Open Text Multi question q5.
Multi or Single with an “other” – option specified in the answer list.	qID _precode_ other	Example: q6_98_other Here 98 is the set precode of the “other” option in the answer list. With “other” you always have to include _other after typing the qID and the precode.

Table 46: Examples of defining multi questions for online coding

13.5 Locking the Questions for Coding

Before a coder can begin working on a specific open-ended question, the question must be locked from other coders, in order to avoid two people making changes to the same question; this will cause problems with the database. Therefore, a separate window where questions can be locked and unlocked is the first window that appears when starting the coding system. The window also displays the binding between the open-ended questions and the corresponding multi questions.

Whenever a coder locks a question in this window a hyperlink will be made to the coding window for that specific question. Only the coder who locked the question will see this hyperlink.

Multiple accesses of many users to one survey can be given, but only one coder at a time can work on a question.

The window for permissions and bindings will look different depending on if you are the administrator or if you are a coder. The administrator shall be able to choose which coder is working on which question. Also, if a coder has chosen to lock a question that has the status "(none)", the name of the coder will automatically appear in the administrator window. The administrator window will look like this:

Current coder: lucillel

Open id	Multi id	Multi question title	Coder	Status
q1	q2	Sports	(none)	0 of 2
q3	q4	Frequency sports are played	coder2	0 of 2
q5	q6	Holiday activities	lucillel	2 of 2

Save permissions

Figure 94: The administrator window

The coders will see a window like this:

Current coder: coder2

Open id	Multi id	Multi question	Locked by	Unlock	Status
q1	q2	Sports	none	-	0 of 2
q3	q4	Frequency sports are played	coder2	☐	0 of 2
q5	q6	Holiday activities	lucillel	-	2 of 2

Unlock

Figure 95: The coder window

13.6 Coding Window

After a question has been locked, follow the hyperlink by clicking on the question, which is to be coded. A screen similar to the one below will appear.

The coding window is a window where the coders can create a working set by searching for records using either a text search of respondent's answers and / or a code number search. The coder can choose which field to sort by, whether he wants to sort ascending or descending, and if he wants to see coded and / or uncoded questions.

Current coder: lucillel

Question no	Question text	# WorkingSet	Current pos.
q5	What activities do you enjoy while on holiday?	2	1

☒ Show code frames

☒ coded Answer-text search: Code search:

☒ uncoded

Resp-ID	Answer-text	Code	code frame
1	Sightseeing, opera, movies, and relaxing	2	Culture, opera, ballet
2	Boating, waterskiing, mountain climbing and glacier trips.	3,4,5	Hiking, Adventure, Water sports

Code	Code frame
1	Skiing
2	Culture, opera, ballet
3	Hiking
4	Adventure
5	Water sports
6	Computer/playstation
7	Exhibitions

Figure 96: The coding window

13.6.1 Elements of the Coding Window

13.6.1.1 Heading

The heading of the Coding Window shows the question that the coder is currently working on, and includes: Question no, Question text, Working set size, and current position in the working set.

13.6.1.2 Searching and Sorting

The next section of the coding window is for "searching and sorting"-answers.

☐ Show code frames

☒ coded Answer-text search: Code search:

☒ uncoded

Resp-ID	Answer-text	Code	code frame
1	Sightseeing, opera, movies, and relaxing	1	Skiing
2	Boating, waterskiing, mountain climbing and glacier trips.	3	Hiking

Figure 97: The searching and sorting section

The coder can choose to include or exclude coded and / or uncoded questions in his working set of respondent answers. The default is "uncoded", in order to list all the questions that have not been coded yet. To be able to specify the working set, especially when the total number of respondents is very high, the coder can also search for a text string and / or code numbers.

The text string search can be done with wildcards to find answers that starts with, ends with, or includes a specific string.

Search criteria	How to write the search word
Includes:	%Textstring%
Starts with:	Text string%
Ends with:	%Text string
Exact search:	Text string

To perform a code search: Enter the number of the code(s) to be searched. If searching for multiple codes, separate the codes with commas. Press go to begin search.

Once your working set is established, it may be sorted by Respondent-ID or Answer-text in ascending or descending order. Answer lists are by default sorted by respondent ID (Resp-ID).

An important thing to notice is that sorting and searching is carried out on all of the answers for the specific question. When sorting/search is changed the working set is changed and hence pressing any button will return the start of the new working set.

When all search criteria has been entered, depress the go button to activate the search.

13.6.1.3 Working Set and Coding

The working set consists of all the respondents for the current question, or the result of the search that the coder has executed in the search fields shown above. As answers are coded, updates to the working set is made after the navigation or save button has been depressed; the coder will now only see the remaining uncoded answers.

The screenshot shows a software interface for coding survey responses. At the top, it says 'Current coder: lucillel'. Below this is a table with columns: 'Question no', 'Question text', '# WorkingSet', and 'Current pos.'. The first row shows 'q5', 'What activities do you enjoy while on holiday?', '2', and '1'. Below this table are search filters: 'coded' (checked), 'uncoded' (checked), 'Answer-text search:' (with a text box containing '%'), 'Code search:' (with a dropdown set to 'Resp-ID' and a 'Go' button), and 'Show code frames' (checked). The main area contains a table with columns: 'Resp-ID', 'Answer-text', 'Code', and 'code frame'. It lists two responses: Response 1 with text 'Sightseeing, opera, movies, and relaxing' and code '2'; Response 2 with text 'Boating, waterskiing, mountain climbing and glacier trips.' and code '3,4,5'. To the right of the main table is a 'Code frame' list with items: 1 Skiing, 2 Culture, opera, ballet, 3 Hiking, 4 Adventure, 5 Water sports, 6 Computer/playstation, 7 Exhibitions. At the bottom are buttons: 'Replace', 'First 14', '<< prev 14', 'next 14 >>', 'Last 14', and 'Save'.

Figure 98: The remaining uncoded answers

The codes that have been added previously might be changed and whenever a code is changed or added it is reflected in the code frame field with the corresponding code words.

Each answer has a corresponding edit box. Arrow keys can be used to move between these edit fields. As explained in the next section, the codes that are added or changed will be saved whenever the navigation buttons or the save button has been pressed.

Codes should be entered corresponding to the precodes defined for the multi question. Should a code be entered, which is not included in the precode list of the multi question, a system generated error message will appear, warning the coder to enter a valid code.

For answer alternatives with multi-punches, remember to separate the codes with a comma, as shown in the screen above.

13.6.1.4 Navigation

The coder is able to navigate through the set of respondent answers in the current open-ended question, through use of the arrow buttons. The buttons in the image below show an example where the coder has chosen to have only 5 answers on the screen where he can then navigate back (<< prev 5), forward (next 5 >>), or to the beginning or end of the record set.

The number of answers to be displayed in the coding window should be entered in the "coding screen size" field in the user-defined menu. This feature lets the coder determine how many answers he wants displayed on the screen.

Clicking on the navigation buttons will save data shown for the current displayed code. The save button should also be used to save codes when an interruption in the coding shall take place (Such as a break or upon completion of the coding).

13.6.1.5 Replace Function

The replace-function is similar to a global replace. It can be used to recode answers in order to sum up different codes under an "over code". An example could be that you want to put all answers, which have been coded for skiing (i.e. code 1) and football (i.e. code 3) as their favourite activities, into another category called sports (i.e. code 99)

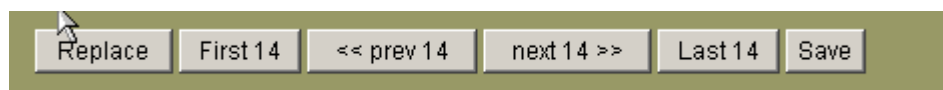


Figure 99: Replace function

Since this is an operation that replaces the codes and is IRREVERSIBLE, the replace function should be used with caution. The warning message below will appear showing how many codes will be affected. If the codes are correct, click ok to continue, otherwise press cancel to end the operation.

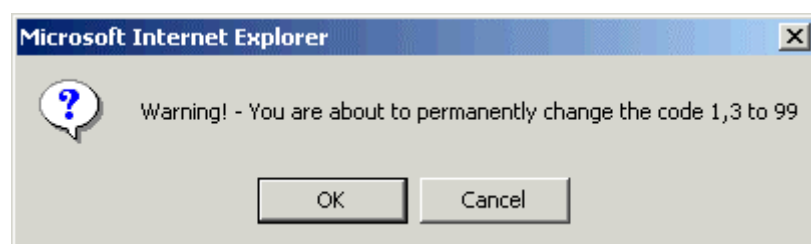


Figure 100: Warning

An error message will also be generated when the coder tries to use a number that is not on the list of codes defined in the precode list of the multi question.

13.6.1.6 Show or Hide Codes

It is possible to show the available codes for the question in the right section of the Coding Window. When the checkbox "show codes" is selected the codes will appear. When deselecting the checkbox, the codes/code frames will disappear from the screen.

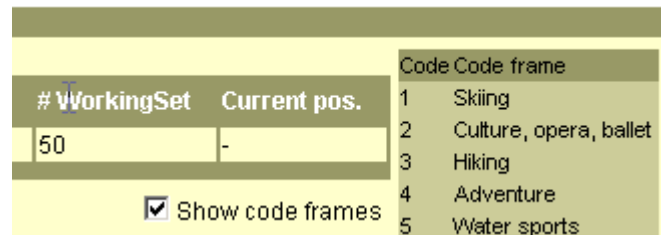


Figure 101: Show and hide codes

13.7 Final Comments

In summary the new Online Coding Tool consists of the following main parts:

- Additions to ConfirmIT on the menu, in the user settings and in project permissions.
- Control panels for setting permissions to the specific open-text questions, and for choosing which questions are to be coded.
- The main coding window where all the coding is done.

13.7.1 Step-by-Step Guide for Online Coding:

1. To change the default user settings for Coding Screen Size (default size is 10) and Working size, (default size is 1000) select User Settings from the main menu and edit the default sizes. Press "Update User Settings" to implement the change. (Coder and Administrator)
2. Click on Object Permissions from the overview menu. The Project Administrator can now grant coding access to online coders. (Administrator only)
3. In the questionnaire tree, drag and drop one multi question, for each open-ended question to be coded. (Administrator only)
4. For each of the multi questions belonging to an open question, fill in the applicable codes in the answer window. Next, open the properties window, and fill in the belonging open-ended question ID and also check off as "hidden". (Administrator only)
5. Click the "Online Coding"-item from the main menu. (Coder and Administrator)
6. Lock the questions and choose which coders will work on which questions. (Administrator only)
7. Click on the hyperlink of the question you wish to begin coding. (Coder)
8. The coding window will now appear showing the list of answers for the chosen question. The answers will appear in Response ID order. (Coder)

9. You may now search for certain strings or codes to ease your work with the working set. Answers can also be sorted by any of the table headings. (Coder)

10. After your search is complete and your working set established, coding of the open-ended questions can begin. (Coder)

14 APPENDIX C: confirmit LANGUAGE CODES

Language code	Language	Sub name	Combident
54	Afrikaans		af
28	Albanian		sq
1	Arabic		ar
1025	Arabic	Saudi Arabia	ar_sa
2049	Arabic	Iraq	ar_iq
3073	Arabic	Egypt	ar_eg
4097	Arabic	Libya	ar_li
5121	Arabic	Algeria	ar_al
6145	Arabic	Morocco	ar_mo
7169	Arabic	Tunisia	ar_tu
8193	Arabic	Oman	ar_om
9217	Arabic	Yemen	ar_ye
10241	Arabic	Syria	ar_sy
11265	Arabic	Jordan	ar_jo
12289	Arabic	Lebanon	ar_le
13313	Arabic	Kuwait	ar_ku
14337	Arabic	U.A.E.	ar_ua
15361	Arabic	Bahrain	ar_ba
16385	Arabic	Qatar	ar_qa
43	Armenian		hy
77	Assamese		as
44	Azeri		az
1068	Azeri	Latin	az_la
2092	Azeri	Cyrillic	az_cy
45	Basque		eu
35	Belarusian		be
69	Bengali		bn
2	Bulgarian		bg
3	Catalan		ca
4	Chinese		zh
1028	Chinese	Taiwan	zh_ta
2052	Chinese	PRC	zh_pr
3076	Chinese	Hong Kong SAR, PRC	zh_hk
4100	Chinese	Singapore	zh_si
5124	Chinese	Macau SAR	zh_ma
5	Czech		cs

6	Danish		da
19	Dutch		nl
1043	Dutch	Netherlands	nl_nl
2067	Dutch	Belgium	nl_be
9	English		en
1033	English	United States	en_us
2057	English	United Kingdom	en_uk
3081	English	Australia	en_au
4105	English	Canada	en_ca
5129	English	New Zealand	en_nz
6153	English	Ireland	en_ir
7177	English	South Africa	en_sa
8201	English	Jamaica	en_ja
9225	English	Caribbean	en_ca
10249	English	Belize	en_be
11273	English	Trinidad	en_tr
12297	English	Zimbabwe	en_zi
13321	English	Philippines	en_ph
37	Estonian		et
56	Faeroese		fo
41	Farsi		fa
11	Finnish		fi
12	French		fr
1036	French	Standard	fr_fr
2060	French	Belgium	fr_be
3084	French	Canada	fr_ca
4108	French	Switzerland	fr_sw
5132	French	Luxembourg	fr_lu
6156	French	Monaco	fr_mo
55	Georgian		ka
7	German		de
1031	German	Standard	de_de
2055	German	Switzerland	de_sw
3079	German	Austria	de_au
4103	German	Luxembourg	de_lu
5127	German	Liechtenstein	de_li
8	Greek		el
71	Gujarati		gu

13	Hebrew		he
57	Hindi		hi
14	Hungarian		hu
15	Icelandic		is
33	Indonesian		id
16	Italian		it
1040	Italian	Standard	it_it
2064	Italian	Switzerland	it_sw
17	Japanese		ja
75	Kannada		kn
96	Kashmiri		ks
2144	Kashmiri	India	ks_in
63	Kazak		kk
87	Konkani		ki
18	Korean		ko
1042	Korean	Korea	ko_ko
2066	Korean	Johab	ko_jo
38	Latvian		lv
39	Lithuanian		lt
1063	Lithuanian	Lithuania	lt_lt
2087	Lithuanian	Classic	lt_cl
47	Macedonian		mk
62	Malay		ms
1086	Malay	Malaysian	ms_ms
2110	Malay	Brunei Darussalam	ms_br
76	Malayalam		ml
88	Manipuri		ma
78	Marathi		mr
97	Nepali		ne
2145	Nepali	India	ne_in
20	Norwegian		no
1044	Norwegian	Bokmål	no_bo
2068	Norwegian	Nynorsk	no_ny
72	Oriya		or
21	Polish		pl
22	Portuguese		pt
1046	Portuguese	Brazil	pt_br
2070	Portuguese	Standard	pt_st

70	Punjabi		pa
24	Romanian		ro
25	Russian		ru
79	Sanskrit		sa
26	Serbian / Croatian		sr
1050	Serbian / Croatian	Croatian	sr_yu
2074	Serbian / Croatian	Latin	sr_la
3098	Serbian / Croatian	Cyrillic	sr_cy
89	Sindhi		sd
27	Slovak		sk
36	Slovenian		sl
10	Spanish		es
1034	Spanish	Traditional Sort	es_es
2058	Spanish	Mexican	es_me
3082	Spanish	Modern Sort	es_ms
4106	Spanish	Guatemala	es_gu
5130	Spanish	Costa Rica	es_cr
6154	Spanish	Panama	es_pa
7178	Spanish	Dominican Republic	es_dr
8202	Spanish	Venezuela	es_ve
9226	Spanish	Colombia	es_co
10250	Spanish	Peru	es_pe
11274	Spanish	Argentina	es_ar
12298	Spanish	Ecuador	es_eq
13322	Spanish	Chile	es_ch
14346	Spanish	Uruguay	es_ur
15370	Spanish	Paraguay	es_pa
16394	Spanish	Bolivia	es_bo
17418	Spanish	El Salvador	es_el
18442	Spanish	Honduras	es_ho
19466	Spanish	Nicaragua	es_ni
20490	Spanish	Puerto Rico	es_pr
65	Swahili		sw
29	Swedish		sv
1053	Swedish	Sweden	sv_sv
2077	Swedish	Finland	sv_fi
73	Tamil		ta
68	Tatar		tt

74	Telugu		te
30	Thai		th
31	Turkish		tr
34	Ukrainian		uk
32	Urdu		ur
1056	Urdu	Pakistan	ur_pa
2080	Urdu	India	ur_in
67	Uzbek		uz
1091	Uzbek	Latin	uz_la
2115	Uzbek	Cyrillic	uz_cy
42	Vietnamese		vi

15 APPENDIX D: RESERVED KEYWORDS

The following is a list of Reserved Keywords that should not be used as Question IDs.

All questions starting with underscore (_)

responseid
respid
rowguid
r
s
l (unless you want to use it as described in 8.4)
rid
interviewer
c
page
status
interview_start
interview_end
projectid
env
last_handled
callback
interviewerid
tries
never_again
in_use
not_in_quota
iterationid
order
comment
sample_category
noofemailssent

Only for use in respondentlists:

userid
password
language

INDEX

3D grid 44
 question type 44
 3rd Party Downloads 11

A

A function 60; 124
 Abort tasks, Task Management 19
 Access to modules and questions 48
 Add
 predefined list 30
 project to favorite list 19
 quotas 40
 validation code 65; 126
 add (function) 125
 Advanced users 9
 Answer list font, User Settings 18
 Answer mask 95; 96
 Answer required checks, interview
 behavior 70
 Answers 28
 list field 48
 required testing 64
 size tests 129
 Answers page
 Background color (property) 29
 Column width 29
 KeepPos 30
 Language fields 29
 Other 30
 Precode 29
 Punch 29
 Weight 29
 Appearance
 controlling 68
 error notifications recipient 68
 fixed area for error messages 69
 include progress bar 69
 survey design 69
 AppendErrorMessage (function) 127
 AppendQuestionErrorMessage (function) .. 127
 Apply filters 98
 Arithmetic functions 120
 ASCII
 export response data to 109
 export to 111
 Automatic time out 10
 Average (function) 120

B

Back button, interview behavior 69
 Back up
 meta data 11
 response data 11
 Background color (property), Answers
 page 29
 Background variables (property) 32

Batch e-mailing 80
 Boolean expression 33; 106
 Branches, conditional field 34
 Browser
 information 123
 limitations 48
 BrowserType (function) 123
 BrowserVersion (function) 123
 Building
 crosstabs 86
 questionnaire 27
 reports 84

C

categories (function) 119
 categoryLabels (function) 119
 CATI 9
 Cell matching - percentage, weight model ... 88
 Cell matching - weight, weight model 89
 Change
 display in matrix views 104
 format in matrix views 104
 matrix size 104
 quota limits 44
 report after publishing 102
 Changes and additions 134
 Chart component
 3rd Party Downloads 11
 User Settings 18
 Chart options, Excel chart component 92; 93
 Check expression 36
 Classification functions 121
 ClearErrorMessage (function) 127
 ClearQuestionErrorMessage (function) 127
 Closed, project status 72
 Code
 hide 142
 language 144
 show 142
 validation 64
 Coded variables 117
 Coders window 138
 Coding 134; 140
 Coding of (property) 32
 Coding screen size, User Settings 18; 134
 Coding system 132; 134
 Coding window 138; 139
 elements 139
 Column name (respondent list) 75
 Column width, Answers page 29
 Combident, language 144
 Comma (operator) 35
 Compile
 quotas 44
 response database 67
 Complete, status-code (stop object) 37
 Completed tasks, Task Management 19
 Compound variable 117

Computer Assisted Telephone Interviewing...	9	nodes	33; 38; 47
Condition Builder	34	Deleted (property)	33; 47
operators and methods	35	Description (module property)	49
Precode	35	Description, folder field	37
variable	35	Design surveys	69
variable definition	108	Destination object.....	47
Conditional fields.....	34	Details view.....	28
Conditions.....	33; 59	Dichotomies variable.....	117
Configure matrix view	103	diff (function)	61; 124
Context information	122	Difference (operator).....	35
Control panel, permission	134	Directives	36
Conversion	120	Multiple_questions	36
Copy		Single_questions.....	36
object.....	47	Display, User Settings	18
project	26	domainLabels (function)	119
rows.....	30	domainValues (function).....	119
Count (function).....	120	Dropdown (property).....	33
Create		Duplicate	
folder	37	object.....	47
form.....	28	project.....	26
matrix view	103		
project	24	E	
questionnaire	27	Edit	
response database.....	67	3D grid objects	45
skip-logic	33	graphs	99
Creator, Report publisher option.....	101	mode (answer page).....	30
Crosstab		routing	47
building.....	86	slides.....	100
options.....	96	template	109
Crosstab settings	91	Element level filters	95
Layout	91	Else node (operator)	34
X totals.....	91	E-mail	
Y totals.....	91	batch	80
Cryptic link	102	send	76; 80
CurrentID (function)	122	setup	77; 79
CurrentLang (function)	123	User Settings	18
CurrentPID (function).....	122	End link (URL setup)	74
CurrentSID (function).....	123	End parenthesis (operator).....	35
D		Engine	
Data		interview.....	51
collection.....	67	scripting.....	51
export	111	Equality (operator)	35; 59
files export	109	Error messages	126; 127
Database, Report publisher option	101	reserve fixed area for	69
Decimals, general setting	91	template based	128
Default language	72	Error notifications recipient.....	68
Define		Error tests	
answers	28	numeric.....	130
filters	98	precision	130
quota counters	40	range.....	131
quota limits	41; 42	scale.....	130
quotas	39	Evaluation of expressions.....	106
Delete		Excel chart component	
object.....	47	3rd Party Downloads.....	12
project	26	chart options	92; 93
quotas	44	report element settings.....	91
Report publisher option.....	101	Excel file, export to	111; 112
Deleted		Exclusivity testing	64; 70; 129
conditional field	34	ExclusivityError (function)	127
		Executing tasks, Task Management	19

Executive	
rights	9
users	10
Exit	23
Export	
ASCII file	111
data	111
data files	109
Excel file	111; 112
format	111; 113
MS Word	115
open questions	113
properties	113
questionnaire	115
reports	85; 115
SPSS	114
Task Management	19
Export data (export property)	113
Export fields	110
Export language (export property)	113
Export properties	
Export data	113
Export language	113
Export template	113
Project ID	113
Start time	113
Export template (export property)	113
Expressions	
Boolean	33; 106
check	36
conditional field	34
JScript	31
syntax and evaluation	106
F	
F function	51; 116
Failed, Report publisher option	101
Favorite projects	19
Field width	38
Field width (property)	31
Fields	
descriptions (conditions)	34
export	110
recoded	110
standard system generated	110
user defined	110
verbatim key fields	110
Web Interview template	21
Filter	98
apply	98
apply in matrix views	104
define	98
element level	95
global	98
make available in matrix views	104
options	95
Filter (function)	125
Folder	
create	37
fields	37
New objects	27
Routing	27
Scales and lists	27; 48
Form	27
create	28
flags	33
properties	32
standard types	27
validation	126
Form types	
3D grid	44
Grid	28; 117
Info	28
Multi	28; 117
Open	28; 117
Single	28; 117
Format	
change	104
export	111; 113
general setting	91
Forms, quotas	39
Forward (function)	123
Frequency, Report publisher option	101
Functions	
a	60; 124
add	125
AppendErrorMessage	127
arithmetic	120
Average	120
BrowserType	123
BrowserVersion	123
categories	119
categoryLabels	119
classification	121
ClearErrorMessage	127
ClearQuestionErrorMessage	127
Count	120
CurrentID	122
CurrentLang	123
CurrentPID	122
CurrentSID	123
diff	61; 124
domainLabels	119
domainValues	119
ExclusivityError	127
f	51; 56; 116
Filter	125
Forward	123
get	118
GetStatus	123
inc	59; 124
InterviewEnd	123
InterviewStart	123
IsDate	122
IsDateFmt	121
isect	61; 124
IsEmail	122
IsInteger	121
IsNet	122
IsNumeric	121

label.....	118	form type	28; 117
Max	120	operators.....	35
members.....	124	Grid component, 3rd Party Downloads	11
Min.....	120		
MissingRequiredError.....	127	H	
nnset.....	60; 124	Handling sets	124
NotSelectedError	128	Help	23
NotSpecifiedError	127	Quick card	23
nset.....	60; 124	Step-By-Step	23
NumericError	128	User Manual	23
PrecisionError	128	Help link (URL setup).....	74
QuestionErrors	127	Hidden (property).....	32
RaiseError	126	Hide chart, general setting.....	91
RangeError.....	128	Hide codes.....	142
RankError	128		
remove	125	I	
replace.....	141	Image (module property).....	49
RequestIP	123	inc (function)	59; 124
ScaleError	128	Include progress bar	69
SendMail.....	125	Includes (operator)	35
set.....	60; 118; 124	Inequality (operator).....	35; 59
SetErrorMessage.....	127	Info, form type.....	28
SetQuestionErrorMessage.....	127	Initiate new project.....	24
SetStatus	123	templates.....	50
size	60; 124	Inner loop	58
SizeError	127	Instruction field	28
Sum	120	Internet-browser limitations	48
toBoolean.....	120	Intersection (operator)	35
toNumber	60; 120	Interview behavior.....	69
toString	120	allow user to modify answers after initial	
union	60; 124	submission.....	69
value.....	118	allow user to modify answers after the	
valueLabel.....	118	interview is complete.....	70
values	118	answer required checks	70
		exclusivity tests	70
G		generate back button.....	69
General		other - specify checking	70
main sections.....	14; 16	rank order tests	70
utilities	125	Interview files, generate	67
General settings.....	90; 91	Interview Management.....	9
Decimals	91	InterviewEnd (function)	123
Format.....	91	InterviewStart (function)	123
Hide chart.....	91	IsDate (function)	122
Multipunch.....	91	IsDateFmt (function)	121
Responses	91	isect (function).....	61; 124
Source	91	IsEmail (function)	122
Weighting	91	IsInteger (function).....	121
Generate		IsNet (function)	122
interview files	67	IsNumeric (function)	121
response database.....	67		
Web interview files	67	J	
Generate surveys, pop-up.....	71	JScript.....	32; 51; 107
get (function)	118	based variables	106
GetStatus (function)	123	code	32
Global filters	98	expression.....	31; 34
Global replace	141	language	34
Greater than (operator).....	35; 59	logical operators	59
Greater than or equal to (operator).....	35; 59	object.....	107
Grid			
3D	44		

K

KeepPos, Answers page.....	30
Keywords, reserved	149

L

label (function).....	118
Language.....	33
Answers page.....	29
codes	144
combident	144
default	72
JScript	34
project	33
Report publisher option.....	101
sub name	144
subset	33
User Settings	18
Last run, Report publisher option	101
Layout, crosstab setting	91
Less than (operator)	35; 59
Less than or equal to (operator)	35; 59
Library	48
templates in	49
Limitations, Internet-browser	48
Limited surveys.....	10
generate login page	70
generating	70
handling respondents	75
multilingual projects	81; 82
pop-ups	71
preparing respondent list.....	75
uploading respondent list	75
Link preview (URL setup)	74
List Rows/Columns (property)	32
Lists	48
nested	48
predefine	48
reuse.....	48
Lock questions for coding.....	137
Log	
project	73
Report publisher option.....	101
Log out.....	23
Logical AND (operator).....	35; 59
Logical NOT (operator).....	35; 59
Logical operators, JScript	59
Logical OR (operator).....	35; 59
Login.....	13
name.....	18
Log-in sessions	10
Loop.....	37; 57
ID	38
inner	58
members.....	39
nested	37; 58
options.....	96; 97
outer	59
reporting.....	87
standard columns in	39

M

Mail merging functionality.....	79
Main page.....	16
Main sections	
General	14; 16
Project	14
Management	
Interview.....	9
Task	19
Matrix size, change	104
Matrix view	102; 105
configuration	103
create	103
cross questions.....	103
filters	103
questions.....	103
rules and guidelines.....	105
using	103
Matrix view operations.....	104
Apply a filter	104
Apply a timeseries.....	104
Change display	104
Change format	104
Change matrix size	104
Make filters available	104
Make timeseries available	104
Select a cross question.....	104
Select a question.....	104
Max (function).....	120
members (function)	124
Menu structure.....	14
Merge mail	79
Meta data, backup	11
Microsoft JScript	51
Microsoft Office Version, User Settings	18
Microsoft, 3rd Party Downloads	12
Min (function)	120
MissingRequiredError (function)	127
Mode, edit (answer page)	30
Modify answers after initial submission,	
interview behavior.....	69
Modify answers after the interview is	
complete, interview behavior	70
Module	
access.....	48
Interview Management.....	9
properties.....	49
reuse	48
Move	
objects	47
variables	111
MS Excel, export response data to	109
MS PowerPoint	
options	94; 95
templates.....	21; 22
MS Word, export to.....	115
Multi	
form type	28; 117
operators/methods	35
Multilingual projects	

handling respondents	81	coding.....	132
limited surveys	82	OpenText (property).....	33
open surveys	81	Opentext coding (property)	33
Multiple_questions, directive	36	Operators	
Multipunch, general setting	91	grid	35
N		multi	35
Name		single	35
folder field	37	Operators and methods, Condition Builder ..	35
list field	48	Order report.....	85
User Settings	18	Ordered (property).....	32
Name (module property).....	49	Other - specify checking.....	70
Name, column (respondent list).....	75	Other, Answers page	30
Navigation.....	141	Outer loop.....	59
Nested loops.....	37; 58	Overview	
New objects, folder	27	general	13
New project.....	24	Project	24
News Page	20	P	
nnset (function)	60; 124	Parameters, search.....	19
Not required (property)	33	Password	13
NotSelectedError (function).....	128	Password (property)	33
NotSpecifiedError (function)	127	Password (user settings)	18
nset (function)	60; 124	Paste rows.....	30
Numeric (property)	32	Permissions	
Numeric error test	130	control panel.....	134
NumericError (function)	128	object	25; 135
O		read.....	10
Object		write.....	10
copy.....	47	Piping	51; 62
delete.....	47	Pitfalls	48
destination.....	47	Pop-up	
duplicate.....	47	limited surveys	71
JScript	107	open survey	71
move	47	script wizard	72
permissions	25; 135	PowerPoint	
remove	47	options	94; 95
script-	37	templates.....	21; 22
security.....	10	Precision error test.....	130
select multiple	47	PrecisionError (function).....	128
session.....	10	Precode.....	28; 62
stop-	36	Answers page	29
undelete.....	47	Condition Builder	35
Object types		Precode mask.....	38; 51; 60
Info.....	28	Precode mask (property)	31
Loop.....	37; 57	Predefine lists	48
One page, Report publisher option	101	Predefined list, add	30
Online coding.....	134	Prepare for data collection.....	67
step-by-step guide	142	Preview.....	33
Online coding tool	132	Preview and confirmation	80
Online reporting.....	83; 102	Primitives, SQLServer.....	106
Open		Production	
form type.....	28; 117	database.....	67
variable	117	project status.....	72
Open questions, export	113	Profile.....	17
Open surveys		Programming questionnaire	27
multilingual projects	81	Progress bar, include	69
URL	68	Project	
Open-ended questions		add	19
categories	132	closed status.....	72
		copy	26

create	24
delete	26
duplicate	26
favorite	19
initiate new	24
languages	33
list	19; 24
log	73
main sections	14
organizer	19
overview	24
production status	72
removed	19
search	19
status	72
Project ID (export property)	113
Properties	30
form	32
module (library)	49
question	32
x.CODED	117
x.COMPOUND	117
x.DICHOTOMY	117
x.OPEN	117
Protected link	102
Publish	
different data	102
reports	99
slides	100
Published report	
change	102
template	22
Publisher	
engine	99
options	100; 101
reports	99
Publishing mode	102
Punch, Answers page	29
Q	
Question ID (property)	31
Question sets	
add question	103
matrix view	103
Question types	52; 56; 63
3D grid	44
Grid	52; 54; 56; 57; 63; 70
Loop	63
Multi	52; 53; 56; 57; 63; 70
Open	52; 63; 70
Other	52; 55; 63
Single	52; 63; 70; 83
QuestionErrors (function)	127
Questionnaire	
building	27
export to MS Word	115
new	27
organizing	37
reuse	48
scripts	51
Questionnaire design, 3rd Party	
Downloads	11
Questions	
access	48
define answers	28
export open	113
locking for coding	137
precode	28
properties	32
reuse	48
unlock	137
Queue, Task Management	19
Quick card, help	23
Quota counters	
define	40
update	44
Quota full, status-code	37
Quotas	39
add	40
change limit	44
check	44
compile	44
define	39
define limits	41; 42
delete	44
details	40; 41; 42; 43
forms	39
respondent info	44
R	
RaiseError (function)	126
Randomize (property)	32
Randomized	38
Range error test	131
RangeError (function)	128
Rank order testing	65; 70; 130
RankError (function)	128
Read permission	10
Recent projects	19
Recipient email address (export property) ..	113
Recipient email address, project ID	113
Recipient, error notifications	68
Recoded fields	110
Recoded variables (property)	32
Recoding	
data	106
variables	106
Recurring tasks, Task Management	19
Remove	
link	19
object	47
project	19
Report publisher option	101
remove (function)	125
Removed project	19
Replace function	141
Report	
building	84
change after publishing	102
Excel	86

export	85; 115	lists	48
open-ended questions.....	132	modules (library)	48
order	85	questionnaires.....	48
PowerPoint.....	86	questions.....	48
publish.....	99	Rights	
Report publisher option.....	101	advanced.....	9
show	85	executive.....	9
Report element settings.....	91	Routing	
Excel chart component.....	91	editing.....	47
Report publisher.....	99	folder	27
Report publisher options	101	Rows	
Active.....	101	copy	30
Creator	101	paste.....	30
Database.....	101	Rows/Columns (property)	31
Delete	101	Rules	
Failed	101	matrix view.....	105
Frequency	101	validation.....	64
Language.....	101	Run Now, Report publisher option.....	101
Last run	101		
Log.....	101	S	
One page	101	Sample balancing, weight model.....	88
Remove.....	101	Scale	30
Report	101	error test.....	130
Respondents.....	101	mask	95; 96
Run Now	101	ScaleError (function).....	128
Targets	101	Scales and lists, folder	27; 48
Template	101	Scratchpad	48
Users	101	Screened, status-code	37
Reporting	83	Script	37; 51
3rd Party Downloads.....	11; 12	nodes	65
loop	87	reference	116
online	83; 102	wizard.....	72
report frame.....	83	Scripting	
reporting tree.....	83	advanced.....	26
settings frame.....	83	engine	51
RequestIP (function)	123	Script-objects.....	37
Required answer testing.....	64	Search	
Reserve fixed area for error messages	69	coding window	139
Reserved keywords.....	149	criteria	140
Respondent		parameters	19
avoid unnecessary inquiries	79	projects	19
handling in limited surveys	75	Task Management	20
handling in multilingual projects	81	Secure Sockets Layer	10
preparing list	75	Security	
Report publisher option.....	101	issues	10
security and anonymity	10	object.....	10
selection	77; 78	Select	
specific questionnaire.....	10	cross question in matrix views	104
system requirements.....	11	multiple objects	47
uploading list.....	75	project.....	19
Response data, backup.....	11	question in matrix views.....	104
Response database		respondents.....	77
create.....	67	Selection criteria.....	77
generate.....	67	Send e-mail.....	76; 80
production	67	SendMail (function)	125
test.....	67	Server installations, Unicode support.....	11
update.....	67	Session	
Response piping.....	62	Log-in.....	10
Responses, general setting	91	object	10
Reuse		set (function)	60; 118; 124

SetErrorMessage (function)	127	System requirements	11
SetQuestionErrorMessage (function).....	127	3rd Party Downloads	11
SetStatus (function).....	123	Respondent	11
Settings	90	Unicode support for server installation	11
general.....	90	users.....	11
report elements.....	91		
stop node.....	37	T	
user.....	17	Target	
Setup e-mail	77; 79	groups	39; 102
Show		Report publisher option	101
codes	142	Task Management	19
deleted nodes.....	47	Abort tasks.....	19
Single		Completed tasks	19
form type.....	28; 117	Executing tasks.....	19
object.....	27	export.....	19
operators	35	Queue	19
question check.....	83	Recurring tasks	19
Single_questions, directive	36	Search.....	20
size (function)	60; 124	Task type	20
SizeError (function)	127	Task type, Task Management.....	20
Skip-logic.....	33	Templates	21
Slide		editor	109
edit	100	error messages	128
publish.....	100	in Initiate New Project.....	50
Sort, coding window	139	in library	49
Source, general setting.....	91	name	110
Specification summary	134	PowerPoint	21; 22
SPSS	31	Published Report	22
export response data to.....	109	Report publisher option.....	101
export to	114	Web Interview	21
SQL.....	32; 78; 106	Test	
based variables	106	answer size	129
SQLServer primitives	106	database	67
SSL	10	Testing	
Standard form types	27	answers	64
Start time (export property)	113	exclusivity	64; 70; 129
Status-code.....	36	rank order	65; 70; 130
Complete.....	37	Text	
Quota full	37	field	28
Screened.....	37	substitution	51; 62
Step-By-Step, help	23	URL setup	74
Stop node settings	37	Then node (operator).....	34
Stop-objects	36	Third Party Downloads	11
Structured Query Language	32; 78; 106	Time out, automatic.....	10
Sub samples	39	Time series	90
Sum (function).....	120	apply.....	104
Survey variables, accessing	116	make available	104
Surveys		properties.....	90
default language	72	Title field	28
design.....	69	toBoolean (function)	120
generating limited	70	toNumber (function).....	60; 120
generating pop-up	71	toString (function).....	120
limited	10; 70	Tracking	33
login page.....	70	Trend reports	90
open.....	68	Type (module property).....	49
public	10		
quota	39	U	
variables	51	Undelete objects	47
Syntax, expressions.....	106	Unicode support, server installations	11
System generated fields	110	union (function).....	60; 124

Union (operator)	35
Update	
quota counters	44
response database	67
URL setup	73; 74
End link	74
Help link	74
Link preview	74
Text	74
URL to survey	68
User	
advanced	9
executive	10
system requirements	11
User defined	
fields	110
User key, User Settings	18
User level, User Settings	18
User name	13
User Settings	18
User Profile	17
User Settings	17; 18
Answer list font	18
Chart component	18
Coding screen size	18; 134
Display	18
E-mail	18
Language	18
Microsoft Office Version	18
Name	18
User key	18
User level	18
User name	18
Working set size	18; 134
User state	10
Users	9
Report publisher option	101
V	
Validation	
code	64
custom	51
default rules	64
form	126
other-specify	65
Validation code	
add	65; 126
default functions for conditions	127
Validation code (property)	32
value (function)	118
valueLabel (function)	118
values (function)	118
Variables	
access survey	116
basic object	118
change order	111
coded	117
compound	117
Condition Builder	35
definition with Condition Builder	108
dichotomies	117
JScript based	106; 107
move	111
open	117
recoding	106
SQL based	106
survey	51
types	117
Verbatim key field	110
Verification, other-specify	129
VideoSoft, 3rd Party Downloads	11
Views, matrix	102
Visual components, 3rd Party Downloads ...	11
W	
Web Interview	
generator	67; 68
template fields	21
templates	21
Web Interview files, generate	67
Weight model	88
properties	89
Weight model types	88
cell matching - percentage	88
cell matching - weight	89
sample balancing	88
Weight, Answers page	29
Weighting, general setting	91
Welcome page	16
WI Template	21
Editor	22
fields	21
WinZip	109
Working set	140
Working set size, User Settings	18; 134
Write permission	10
X	
X totals, crosstab setting	91
x.CODED (property)	117
x.COMPOUND (property)	117
x.DICHOTOMY (property)	117
x.OPEN (property)	117
Y	
Y totals, crosstab setting	91
Z	
Zip files	109