

confirmit
- User Manual -
Version 6.0 ~~5.1~~

September 30 ~~July 11~~, 2000

The information on these pages describes **confirmit** and its ~~s~~e features as of September 30th ~~July 11th~~, 2000. New features may be introduced into **confirmit** after this date. Please check “News” for latest updates.

Formatted

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 Introduction	16
1.1 WHAT IS CONFIRMIT?	16
1.2 ABOUT THIS MANUAL	16
1.2.1 Interview Management	16
1.3 CONFIRMIT USERS	16
1.3.1 Advanced Users	16
1.3.2 Executive Users	17
1.4 ABOUT LOG-IN SESSIONS	17
1.5 SECURITY ISSUES	17
1.5.1 Secure Sockets Layer (SSL)	17
1.5.2 Object Security	17
1.5.3 Respondent Security and Anonymity.....	17
1.5.4 Back up of Meta Data	18
1.5.5 Back up of Response Data	18
1.6 SYSTEM REQUIREMENTS	18
1.6.1 confirmit Users.....	18
1.6.2 Respondent Requirement	18
1.6.3 3 rd Party Downloads.....	18
2 Entering confirmit – a General Overview	19
2.1 CONFIRMIT LOGIN	19
2.2 CONFIRMIT MENU STRUCTURE.....	20
3 General	23232323
3.1 MAIN PAGE	23 23 23 23
3.2 USER SETTINGS	25 25 24
USER’S LANGUAGE OF PREFERENCE. <u>THIS IS THE LANGUAGE USED WHEN PREVIEWING QUESTIONS AND REPORTS.</u>	27 27 26
3.3 PROJECT LIST	28 28 27
3.4 TASK MANAGEMENT.....	28 28 27
3.5 NEWS PAGE.....	30 30 29
3.6 TEMPLATES	30 30 29
3.6.1 Web Interview Templates.....	30 30 29
3.6.1.1 The Fields in the WI Template Edit Modus	31 31 30
3.6.1.2 Published Report Templates.....	31 31 30
3.6.2 MS PowerPoint Templates	33 33 32
3.7 HELP	33 33 32
3.8 EXITING CONFIRMIT	33 33 32
4 Project Overview	353533
4.1 OBJECT PERMISSIONS	37 37 35

Formatted

4.2	Duplicate Project	383836
4.3	Delete Project.....	383836
5	Building Questionnaires.....	404038
5.1	Creating a New Questionnaire	404038
5.2	Adding New Objects to the Questionnaire.....	414139
5.2.1	Forms.....	414139
5.2.1.1	Text	424240
5.2.1.2	Answers.....	424240
5.2.1.3	Scale.....	444442
5.2.1.4	Properties	444442
5.2.1.5	Preview	474745
5.2.1.6	Tracking	474745
5.2.1.7	Languages	474745
5.2.2	Conditions.....	474745
5.2.2.1	Description of Fields.....	484846
5.2.2.2	Condition Builder.....	484846
5.2.3	Directives.....	505048
5.2.4	Stop-objects	505048
5.2.5	Scripts	515148
5.2.6	Folders.....	515149
5.2.7	Loops	515149
5.2.8	Quotas.....	535351
5.2.8.1	Defining quotas and quota forms	535351
5.2.8.2	Defining quota counters	545452
5.2.8.3	Compilation of quotas	585856
5.2.8.4	Changing quota limits	585856
5.2.8.5	Updating quota counters during interviewing	585856
5.2.8.6	Checking quotas during interviewing.....	585856
5.2.8.7	Respondent info for full quotas.....	585856
5.2.9	3D – Grid.....	585856
5.3	Editing the Routing	606058
5.3.1	Moving an Object	616159
5.3.2	Duplicating an Object.....	616159
5.3.3	Selecting Multiple Objects.....	616159
5.3.4	Deleting Objects	616159
5.3.5	Pitfalls.....	616159
5.4	Scratchpad.....	626260
5.5	Lists	626260
5.6	Confirmit Library	626260
5.6.1	Templates in Library	636361
6	Confirmit Scripts	656563

6.1	INTRODUCTION.....	656563
6.2	ACCESSING SURVEY VARIABLES – THE F FUNCTION	656563
6.2.1	<i>f("qID")</i>	666664
6.2.1.1	Single	666664
6.2.1.2	Open Text.....	666664
6.2.1.3	Multi.....	676765
6.2.1.4	Multi with OpenText or Ordered Property Checked	676765
6.2.1.5	Grid	686866
6.2.1.6	Other, Specify	696967
6.2.2	<i>f("qID")["precode"]</i>	707068
6.2.2.1	Multi.....	707068
6.2.2.2	Multi with Ordered Property Checked	717169
6.2.2.3	Multi with OpenText Property Checked	717169
6.2.2.4	Grids.....	717169
6.2.3	<i>Loops - f("qID", "iteration_precode", "iteration_precode",)</i>	717169
6.3	CONDITIONS	737371
6.3.1	<i>.inc()</i>	737371
6.3.2	<i>.size()</i>	747472
6.3.3	<i>.toNumber()</i>	747472
6.4	PRECODE MASKS.....	747472
6.4.1	<i>a("qID")</i>	747472
6.4.2	<i>set(), nset(), nnset()</i>	747472
6.4.3	<i>.union()</i>	747472
6.4.4	<i>.isect()</i>	757573
6.4.5	<i>.diff()</i>	757573
6.4.6	<i>Precodes</i>	767674
6.5	TEXT SUBSTITUTION/RESPONSE PIPING	767674
6.6	VALIDATION CODE.....	787876
6.6.1	<i>Default Validation Rules</i>	787876
6.6.2	<i>Adding Your Own Validation Code</i>	797977
6.7	SCRIPT NODES.....	797977
7	Preparing for Data Collection	818179
7.1	GENERATING RESPONSE DATABASES.....	818179
7.2	GENERATING INTERVIEW FILES.....	818179
7.2.1	<i>Web Interview Files</i>	818179
7.2.2	<i>Options Controlling Appearance</i>	848482
7.2.2.1	Recipient of Error Notifications.....	848482
7.2.2.2	Include Progress Bar	848482
7.2.2.3	Reserve a Fixed Area for Error Messages	848482
7.2.2.4	Survey Design.....	848482
7.2.3	<i>Options Affecting Interview Behavior</i>	858583

7.2.3.1	Generate Back Button	858583
7.2.3.2	Allow user to modify answers after initial submission.....	858583
7.2.3.3	Allow user to modify answers after the interview is complete.....	858583
7.2.3.4	Answer Required Checks	858583
7.2.3.5	Exclusivity Tests	858583
7.2.3.6	Other – Specify Checking	868684
7.2.3.7	Rank Order Tests.....	868684
7.2.4	User Defined Error Messages	868684

Language: English Save

Standard forms

General

Forms used in all generated interviews.

"Finished" page title

Displayed text <i>English, system provided</i>	Survey Completed!
System provided text	Survey Completed!
Edit local text:	

"Finished" page text

Displayed text <i>English, system provided</i>	Thank you very much for your time.
System provided text	Thank you very much for your time.
Edit local text:	

.....		878785
7.2.5	Options for Generating Limited Surveys	878785
7.2.5.1	Limited Survey.....	878785
7.2.5.2	Generate Login Page	878785
7.2.6	Pop-Ups.....	888886
7.2.6.1	Generate Pop-up Survey.....	888886
7.2.6.2	An Example of a Script for Pop-up Surveys	888886
7.2.6.3	.Pop-up Script Wizard.....	898987
7.2.7	Default Language	898987
7.3	PROJECT STATUS	898987

7.4	PROJECT LOG	909088
7.5	URL SETUP	909088
8	Handling Respondents in Limited Surveys	929290
8.1	PREPARING THE RESPONDENT LIST	929290
8.2	UPLOADING THE RESPONDENT LIST	929290
8.3	SENDING E-MAIL	939391
8.3.1	Step 1: Respondent Selection.....	949492
8.3.2	Mail Merging Functionality	979795
8.3.3	Avoiding Unnecessary Respondent Inquiries	979795
8.3.4	Step 2: Preview and Confirmation.....	989896
8.3.5	Sending E-mails.....	989896
8.4	HANDLING RESPONDENTS IN MULTILINGUAL PROJECTS.....	999997
8.4.1	Open Multilingual Surveys	999997
8.4.2	Limited Multilingual Surveys.....	10010098
9	Reporting.....	10110199
9.1	INTRODUCTION.....	10110199
9.2	CHECKING A SINGLE QUESTION	10110199
9.3	BUILDING REPORTS	102102100
9.4	BUILDING CROSSTABS.....	104104102
9.5	REPORTING ON LOOPS	105105103
9.6	WEIGHT MODEL	106106104
9.7	TIME SERIES.....	108108106
9.8	SETTINGS.....	108108106
9.8.1	General Settings	108108106
9.8.2	Report Elements Settings.....	109109107
9.8.2.1	Excel Chart Component	109109107
9.8.2.2	PowerPoint Options	112112110
9.8.2.3	Filter Options	114114112
9.8.2.4	Answer Mask/Scale Mask	114114112
9.8.2.5	Crosstab Options	115115113
9.8.2.6	Loop Options.....	116116114
9.9	FILTERS	117117115
9.9.1	Defining Filters.....	117117115
9.9.2	Applying Filters	117117115
9.10	REPORT PUBLISHER.....	118118116
9.10.1	Publishing a Report.....	118118116
9.10.2	Changing a Report after Publishing.....	121121119
9.10.3	Publishing Different Data from One Survey to Separate Target Groups, or Same Report in Different Languages.....	121121119
9.11	MATRIX VIEWS	121121119

9.11.1	Introduction	121+21+19
9.11.2	Configure Matrix Views.....	122+22+20
9.11.3	Using Matrix Views	122+22+20
9.11.4	Using Recoded Variables	Error! Bookmark not defined. Error! Bookmark not defined. 122
10	Recoding Data	126+26+25
10.1	EXPRESSION SYNTAX AND EVALUATION.....	126+26+25
10.1.1	SQL.....	126+26+25
10.1.2	JScript.....	127+27+26
11	Exports of Data Files	129+29+28
11.1	TEMPLATE EDITOR.....	129+29+28
11.2	DATA EXPORT	131+31+30
11.3	SPSS EXPORT	133+33+32
11.4	MS WORD EXPORT	134+34+33
12	APPENDIX A: confirmit Script Reference	136+36+35
12.1	ACCESSING SURVEY VARIABLES.....	136+36+35
12.1.1	Variable Types.....	137+37+36
12.2	USEFUL FUNCTIONS AND METHODS	138+38+37
12.2.1	Basic variable objects.....	138+38+37
12.2.1.1	.get().....	138+38+37
12.2.1.2	.set().....	138+38+37
12.2.1.3	.label().....	138+38+37
12.2.1.4	.value().....	138+38+37
12.2.1.5	.valueLabel().....	138+38+37
12.2.1.6	.values()	138+38+37
12.2.1.7	.categories()	139+39+38
12.2.1.8	.categoryLabels()	139+39+38
12.2.1.9	.domainValues().....	139+39+38
12.2.1.10	.domainLabels()	139+39+38
12.2.1.11	Some practical use	139+39+38
12.2.2	Arithmetic functions.....	140+40+39
12.2.2.1	Sum()	140+40+39
12.2.2.2	Count().....	140+40+39
12.2.2.3	Average().....	140+40+39
12.2.2.4	Max()	140+40+39
12.2.2.5	Min().....	140+40+39
12.2.3	Conversion.....	140+40+39
12.2.3.1	.toString().....	140+40+39
12.2.3.2	.toNumber()	140+40+39
12.2.3.3	.toBoolean()	141+41+40
12.2.4	Classification functions	141+41+40

12.2.4.1	IsNumeric(arg).....	141141140
12.2.4.2	IsInteger(arg).....	141141140
12.2.4.3	IsDateFmt(arg, format).....	141141140
12.2.4.4	IsDate(day, month, year).....	142142141
12.2.4.5	IsEmail(arg).....	142142141
12.2.4.6	IsNet(quadIP, quadNet, quadMask).....	142142141
12.2.5	<i>Context information</i>	142142141
12.2.5.1	CurrentID().....	142142141
12.2.5.2	CurrentPID().....	143143141
12.2.5.3	CurrentSID().....	143143142
12.2.5.4	CurrentLang().....	143143142
12.2.5.5	InterviewStart().....	143143142
12.2.5.6	InterviewEnd().....	143143142
12.2.5.7	GetStatus().....	143143142
12.2.5.8	SetStatus().....	143143142
12.2.5.9	Forward().....	143143142
12.2.6	<i>Browser information</i>	143143142
12.2.6.1	BrowserType().....	143143142
12.2.6.2	BrowserVersion().....	143143142
12.2.6.3	RequestIP().....	143143142
12.2.7	<i>Handling sets</i>	144144143
12.2.7.1	a('qID').....	144144143
12.2.7.2	set().....	144144143
12.2.7.3	nset().....	144144143
12.2.7.4	nnset().....	144144143
12.2.7.5	.members().....	144144143
12.2.7.6	.union().....	144144143
12.2.7.7	.isect().....	144144143
12.2.7.8	.diff().....	144144143
12.2.7.9	.size().....	144144143
12.2.7.10	.inc().....	144144143
12.2.7.11	.add().....	145145144
12.2.7.12	.remove().....	145145144
12.2.7.13	Filter(qID,prefix).....	145145144
12.2.8	<i>General utilities</i>	145145144
12.2.8.1	SendMail(from, to, subject, body).....	145145144
12.3	FORM VALIDATION.....	146146145
12.3.1	<i>Adding Your Own Validation Code</i>	146146145
12.3.1.1	A typical Validation Code.....	146146145
12.3.1.2	RaiseError().....	146146145
12.3.1.3	Error Messages.....	146146145
12.3.1.4	Some default functions for Conditions in Validation Code.....	147147146

12.3.1.5	Template Based Error Messages.....	148148147
13	APPENDIX B: confirmit LANGUAGE CODES	152152151
14	APPENDIX C: RESERVED KEYWORDS	153153152
	TABLE OF CONTENTS	2
1	Introduction	17
1.1	WHAT IS CONFIRMIT?	17
1.2	ABOUT THIS MANUAL	17
1.2.1	Interview Management	17
1.3	CONFIRMIT USERS	17
1.3.1	Advanced Users	17
1.3.2	Executive Users	18
1.4	ABOUT LOG-IN SESSIONS	18
1.5	SECURITY ISSUES	18
1.5.1	Secure Sockets Layer (SSL)	18
1.5.2	Object Security	18
1.5.3	Respondent Security and Anonymity	18
1.5.4	Back up of Meta Data	19
1.5.5	Back up of Response Data	19
1.6	SYSTEM REQUIREMENTS	19
1.6.1	confirmit Users	19
1.6.2	Respondent Requirement	19
1.6.3	3 rd Party Downloads	19
2	Entering confirmit – a General Overview	20
2.1	CONFIRMIT LOGIN	20
2.2	CONFIRMIT MENU STRUCTURE	21
3	General	23
3.1	MAIN PAGE	23
3.2	USER SETTINGS	24
3.3	PROJECT LIST	25
3.4	TASK MANAGEMENT	26
3.5	NEWS PAGE	27
3.6	TEMPLATES	27
3.6.1	Web Interview Templates	28
3.6.1.1	The Fields in the WI Template Edit Modus	28
3.6.1.2	Published Report Templates	29
3.6.2	MS PowerPoint Templates	30
3.7	HELP	30
3.8	FAQ	30
3.9	EXITING CONFIRMIT	30

Formatted

4	<u>Project Overview</u>	31
	<u>Nytt bilde!!</u>	31
4.1	<u>OBJECT PERMISSIONS</u>	32
	<u>Nytt bilde!!</u>	32
4.2	<u>DUPLICATE PROJECT</u>	33
4.3	<u>DELETE PROJECT</u>	33
5	<u>Building Questionnaires</u>	35
5.1	<u>CREATING A NEW QUESTIONNAIRE</u>	35
	<u>Nytt bilde!</u>	35
5.2	<u>ADDING NEW OBJECTS TO THE QUESTIONNAIRE</u>	35
5.2.1	<u>Forms</u>	36
5.2.1.1	<u>Text</u>	36
5.2.1.2	<u>Answers</u>	36
5.2.1.3	<u>Scale</u>	38
5.2.1.4	<u>Properties</u>	38
5.2.1.5	<u>Preview</u>	41
5.2.1.6	<u>Tracking</u>	41
5.2.1.7	<u>Languages</u>	41
5.2.2	<u>Conditions</u>	41
5.2.2.1	<u>Description of Fields</u>	42
5.2.2.2	<u>Condition Builder</u>	42
5.2.3	<u>Directives</u>	44
5.2.4	<u>Stop objects</u>	44
5.2.5	<u>Scripts</u>	44
5.2.6	<u>Folders</u>	45
5.2.7	<u>Loops</u>	45
5.2.8	<u>Quotas</u>	47
5.2.8.1	<u>Defining quotas and quota forms</u>	47
5.2.8.2	<u>Defining quota counters</u>	48
5.2.8.3	<u>Compilation of quotas</u>	52
5.2.8.4	<u>Changing quota limits</u>	52
5.2.8.5	<u>Updating quota counters during interviewing</u>	52
5.2.8.6	<u>Checking quotas during interviewing</u>	52
5.2.8.7	<u>Respondent info for full quotas</u>	52
5.2.9	<u>3D Grid</u>	52
5.3	<u>EDITING THE ROUTING</u>	54
5.3.1	<u>Moving an Object</u>	55
5.3.2	<u>Duplicating an Object</u>	55
5.3.3	<u>Selecting Multiple Objects</u>	55
5.3.4	<u>Deleting Objects</u>	55
5.3.5	<u>Pitfalls</u>	55

5.4	SCRATCHPAD.....	56
5.5	LISTS.....	56
5.6	CONFIRMIT LIBRARY.....	56
5.6.1	Templates in Library.....	57
6	CONFIRMIT Scripts.....	59
6.1	INTRODUCTION.....	59
6.2	ACCESSING SURVEY VARIABLES THE F FUNCTION.....	59
6.2.1	<i>f</i> ("qID").....	60
6.2.1.1	Single.....	60
6.2.1.2	Open Text.....	60
6.2.1.3	Multi.....	61
6.2.1.4	Multi with OpenText or Ordered Property Checked.....	61
6.2.1.5	Grid.....	62
6.2.1.6	Other, Specify.....	63
6.2.2	<i>f</i> ("qID")["precode"].....	64
6.2.2.1	Multi.....	64
6.2.2.2	Multi with Ordered Property Checked.....	65
6.2.2.3	Multi with OpenText Property Checked.....	65
6.2.2.4	Grids.....	65
6.2.3	Loops <i>f</i> ("qID", "iteration_precode", "iteration_precode",).....	65
6.3	CONDITIONS.....	67
6.3.1	<i>inc()</i>	67
6.3.2	<i>size()</i>	68
6.3.3	<i>toNumber()</i>	68
6.4	PRECODE MASKS.....	68
6.4.1	<i>a</i> ("qID").....	68
6.4.2	<i>set()</i> , <i>nset()</i> , <i>nnset()</i>	68
6.4.3	<i>union()</i>	68
6.4.4	<i>isect()</i>	69
6.4.5	<i>diff()</i>	69
6.4.6	Precodes.....	70
6.5	TEXT SUBSTITUTION/RESPONSE PIPING.....	70
6.6	VALIDATION CODE.....	72
6.6.1	Default Validation Rules.....	72
6.6.2	Adding Your Own Validation Code.....	73
6.7	SCRIPT NODES.....	73
7	Preparing for Data Collection.....	75
7.1	GENERATING RESPONSE DATABASES.....	75
7.2	GENERATING INTERVIEW FILES.....	75
7.2.1	Web Interview Files.....	75

7.2.2	Options Controlling Appearance.....	77
7.2.2.1	Recipient of Error Notifications.....	77
7.2.2.2	Include Progress Bar.....	78
7.2.2.3	Reserve a Fixed Area for Error Messages.....	78
7.2.2.4	Survey Design.....	78
7.2.2.5	Body Attributes.....	78
7.2.2.6	Compact Layout.....	78
7.2.3	Options Affecting Interview Behavior.....	78
7.2.3.1	Generate Back Button.....	78
7.2.3.2	Allow Modification of Previous Answers.....	78
7.2.3.3	Answer Required Checks.....	79
7.2.3.4	Exclusivity Tests.....	79
7.2.3.5	Other – Specify Checking.....	79
7.2.3.6	Rank Order Tests.....	79
7.2.3.7	Disable “Answer Required” Validation.....	79
7.2.4	User Defined Error Messages.....	79
7.2.5	Options for Generating Limited Surveys.....	80
7.2.5.1	Limited Survey.....	80
7.2.5.2	Generate Login Page.....	80
7.2.6	Pop-Ups.....	81
7.2.6.1	Generate Pop-up Survey.....	81
7.2.6.2	An Example of a Script for Pop-up Surveys.....	81
7.2.6.3	Pop-up Script Wizard.....	82
7.2.7	Default Language.....	82
7.3	PROJECT STATUS.....	82
7.4	PROJECT LOG.....	83
7.5	URL SETUP.....	83
8	Handling Respondents in Limited Surveys.....	85
8.1	PREPARING THE RESPONDENT LIST.....	85
8.2	UPLOADING THE RESPONDENT LIST.....	85
8.3	SENDING E-MAIL.....	86
8.3.1	Step 1: Respondent Selection.....	87
8.3.2	Mail Merging Functionality.....	90
8.3.3	Avoiding Unnecessary Respondent Inquiries.....	90
8.3.4	Step 2: Preview and Confirmation.....	91
8.3.5	Sending E-mails.....	91
8.4	HANDLING RESPONDENTS IN MULTILINGUAL PROJECTS.....	92
8.4.1	Open Multilingual Surveys.....	92
8.4.2	Limited Multilingual Surveys.....	93
9	Reporting.....	94
9.1	INTRODUCTION.....	94

9.2	CHECKING A SINGLE QUESTION	94
9.3	BUILDING REPORTS	95
9.4	BUILDING CROSSTABS	97
9.5	REPORTING ON LOOPS	98
9.6	WEIGHT MODEL	99
9.7	TIME SERIES	101
9.8	SETTINGS	101
9.8.1	General Settings	101
9.8.2	Report Elements Settings	102
9.8.2.1	Excel Chart Component	102
9.8.2.2	PowerPoint Options	105
9.8.2.3	Filter Options	107
9.8.2.4	Answer Mask/Scale Mask	107
9.8.2.5	Crosstab Options	108
9.8.2.6	Loop Options	109
9.9	FILTERS	110
9.9.1	Defining Filters	110
9.9.2	Applying Filters	110
9.10	REPORT PUBLISHER	111
9.10.1	Publishing a Report	111
9.10.2	Changing a Report after Publishing	114
9.10.3	Publishing Different Data from One Survey to Separate Target Groups, or Same Report in Different Languages	114
9.11	MATRIX VIEWS	114
9.11.1	Introduction	114
9.11.2	Configure Matrix Views	115
9.11.3	Using Matrix Views	115
9.11.4	Using Recoded Variables	117
10	Recoding Data	120
10.1	EXPRESSION SYNTAX AND EVALUATION	120
10.1.1	SQL	120
10.1.2	JScript	121
11	Exports of Data Files	123
11.1	TEMPLATE EDITOR	123
11.2	DATA EXPORT	125
11.3	SPSS EXPORT	127
11.4	MS WORD EXPORT	128
12	APPENDIX A: confirmit Script Reference	130
12.1	ACCESSING SURVEY VARIABLES	130
12.1.1	Variable Types	131

12.2 — USEFUL FUNCTIONS AND METHODS	132
12.2.1 — Basic variable objects	132
12.2.1.1 — .get()	132
12.2.1.2 — .set(v)	132
12.2.1.3 — .label()	132
12.2.1.4 — .value()	132
12.2.1.5 — .valueLabel()	132
12.2.1.6 — .values()	132
12.2.1.7 — .categories()	133
12.2.1.8 — .categoryLabels()	133
12.2.1.9 — .domainValues()	133
12.2.1.10 — .domainLabels()	133
12.2.1.11 — Some practical use	133
12.2.2 — Arithmetic functions	134
12.2.2.1 — Sum()	134
12.2.2.2 — Count()	134
12.2.2.3 — Average()	134
12.2.2.4 — Max()	134
12.2.2.5 — Min()	134
12.2.3 — Conversion	134
12.2.3.1 — .toString()	134
12.2.3.2 — .toNumber()	134
12.2.3.3 — .toBoolean()	134
12.2.4 — Classification functions	135
12.2.4.1 — IsNumeric(arg)	135
12.2.4.2 — IsInteger(arg)	135
12.2.4.3 — IsDateFmt(arg, format)	135
12.2.4.4 — IsDate(day, month, year)	136
12.2.4.5 — IsEmail(arg)	136
12.2.4.6 — IsNet(quadIP, quadNet, quadMask)	136
12.2.5 — Context information	136
12.2.5.1 — CurrentID()	136
12.2.5.2 — CurrentPID()	136
12.2.5.3 — CurrentSID()	137
12.2.5.4 — CurrentLang()	137
12.2.5.5 — InterviewStart()	137
12.2.5.6 — InterviewEnd()	137
12.2.5.7 — GetStatus()	137
12.2.5.8 — SetStatus()	137
12.2.5.9 — Forward()	137
12.2.6 — Browser information	137
12.2.6.1 — BrowserType()	137

12.2.6.2	BrowserVersion()	137
12.2.6.3	RequestIP()	137
12.2.7	Handling sets	138
12.2.7.1	a('qID')	138
12.2.7.2	set()	138
12.2.7.3	nset()	138
12.2.7.4	nset()	138
12.2.7.5	.members()	138
12.2.7.6	.union()	138
12.2.7.7	.isect()	138
12.2.7.8	.diff()	138
12.2.7.9	.size()	138
12.2.7.10	.inc()	138
12.2.7.11	.add()	139
12.2.7.12	.remove()	139
12.2.7.13	Filter(qID,prefix)	139
12.2.8	General utilities	139
12.2.8.1	SendMail(from, to, subject, body)	139
12.3	FORM VALIDATION	140
12.3.1	Adding Your Own Validation Code	140
12.3.1.1	Runtime Functions for Error Handling	140
12.3.1.2	A typical Validation Code	140
12.3.1.3	RaiseError()	140
12.3.1.4	Error Messages	140
12.3.1.5	Template Based Error Messages	141
12.3.1.6	Some default functions for Conditions in Validation Code	142
12.3.1.7	Template based error messages	147
13	APPENDIX B: confirm LANGUAGE CODES	151
14	APPENDIX C: RESERVED KEYWORDS	152

1 Introduction

1.1 What is confirmit?

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

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

Formatted

1.2 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.

1.2.1 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.

1.3 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.

1.3.1 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:

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

1.3.2 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 ~~program-build~~ questionnaires or create reports themselves.

1.4 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.

1.5 Security Issues

Login:

1.5.1 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 ~~probability-possibility~~ of someone listening in on your traffic is very small indeed.

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

1.5.2 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.

1.5.3 Respondent Security and Anonymity

Web surveys can be set up to run as *public surveys*, in which any ~~onebody~~ 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 ~~simply~~ because this information already exists in his or her profile.

1.5.4 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 ~~this~~ 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.

1.5.5 Back up of Response Data

It is possible to delete response data from the web application. You are always asked twice before ~~about~~ 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 exists.

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

1.6 System Requirements

1.6.1 confirmit Users

~~CONFIRMIT~~ **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).

Formatted

1.6.2 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.

1.6.3 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](#)~~Figure 1: Login box~~~~Figure 1: Login box~~~~Figure 1: Login box~~~~picture~~. 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 display~~~~ed~~.

Formatted



Figure 1: Login box

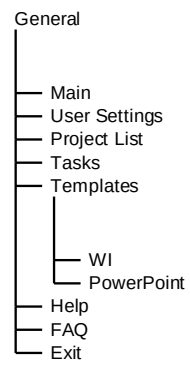
2.2 confirmit Menu Structure

When you have logged in to **confirmit**, you will be brought to the first page of **confirmit**. On 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**.

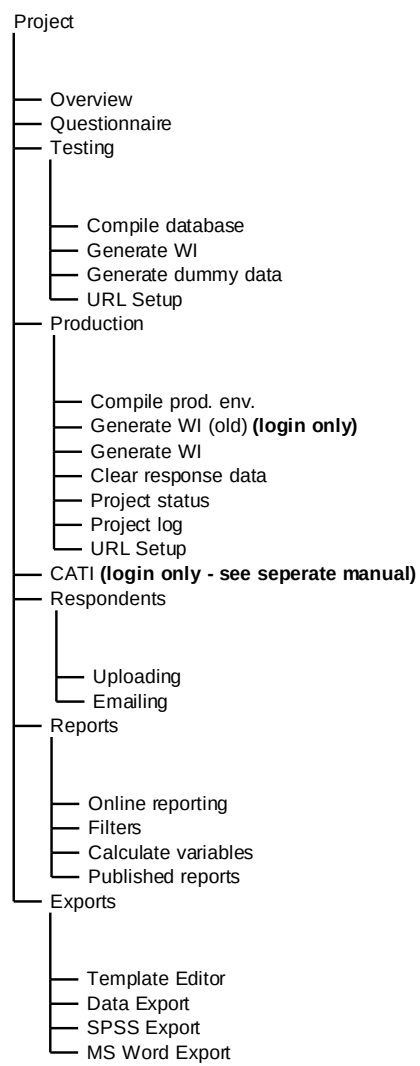
The following chapters offer a more detailed description of every choice in the menu.



Figure 222: 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.

WHAT DO YOU WANT TO DO TODAY? _____ CONFIRM _____

Welcome, John Smith of FIRM AS. Please select an action.

Options		
	View user settings.	Sets the preferred language, email address etc.
	View project list.	List of all previous research.
	Initiate new project.	Select from set of available survey templates.
	Tasks.	Status information on your batch jobs.
	News.	Latest update: Wed Jan 12: - Takk for meq!

WHAT DO YOU WANT TO DO TODAY? p16169441

Welcome, John Smith of FIRM AS. Please select an action.

Options

	View user settings.	Sets the preferred language, email address etc.
	View project list.	List of all previous research.
	Initiate new project.	Select from set of available survey templates.
	Tasks.	Status information on your batch jobs.
	News.	Latest update: No news.

Figure 3333: 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

CONFIRM

Personal settings for johns	
User name:	johns
User key:	JRTMVOJBQYWQYIUTXHTTFXEK
First name:	John
Last name:	Smith
E-mail:	John@somewhere.com
Language:	English
Display:	1024 x 768
User level:	Advanced
Microsoft Office version:	Newest version
Chart component:	Excel
Update settings Change password	

USER SETTINGS

confirmit

Personal settings for kristinem	
User name:	kristinem
User key:	YXWWXMWLWEXKUQVSWBZSFLTW
First name:	John
Last name:	Smith
E-mail:	john@anywhere.com
Language:	English
Display:	1024 x 768
User level:	Advanced
Microsoft Office version:	Newest version
Chart component:	Excel
Coding Screen Size:	0
Working set size:	0
Answerlist font:	
Update settings Change password	

Figure 444: 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 <u>will</u> 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	Preferred language of the user <u>User’s language of preference.</u> This is the language used when previewing questions and reports. <u>Unicode that enables the software to handle Asian languages has been introduced into confirmit.</u> <u>The user will be able to choose any language when designing a questionnaire.</u>
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 1.3.
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).
<u>Coding Screen Size</u>	<u>X</u>	<u>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.</u>
<u>Working set size</u>	<u>X</u>	<u>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.</u>
<u>Answer list font</u>	<u>X</u>	

Formatted

Formatted

Formatted

Formatted

Formatted

Formatted

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 Filter
Favorite projects				
Test Project	p0793814	FIRM AS	Smith, John	Last week [Rem]
Search projects				
				Search
Results				

Figure 5555: 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. **In-queue**: A list of the user's pending tasks.
2. **Executing tasks**: Task currently being run at the batch server.
3. **Aborted/Killed 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. Cancelled tasks: Tasks that have been cancelled by the user before execution. Recurr: Tasks that have been scheduled to recur at specified time intervals.

6. Search: 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, aborted by system.
- Task type: SPSS Export, MS Word Export, MS Office Export (PPT/Excel), Data Export, Batch Emailing.

Formatted: Bullets and Numbering

TASKS

p0793814

Showing tasks for user johns.

In queue	Executing	Completed tasks	Killed tasks	Cancelled tasks			
TaskID	Scheduled start time	Description	Comments	Last changed	ProjectID	Initiated	
220	00:02:00 17:15:20	SPSS export of project p0793814		00:02:00 17:15:20	p0793814	00:02:00 17:15:20	Cancel

Current system time: Tue Feb 8 17:15:29 UTC+0100 2000

Refresh list

Clear list

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)	

Figure 6666: Task management

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.

CONFIRM NEWS ————— CONFIRM —
Latest additions on top.

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

All categories Search Go

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 <i>CONFIRM FAQ</i> is now in place. ▶

Figure 7777: 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 ~~right~~ 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 ~~ef~~ 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 include 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).

3.8FAQ

~~FAQ or Frequently Asked Questions is a repository of the most commonly asked questions and answers concerning the use of confirmit.~~

~~Support link gives confirmit users information about how FIRM's Support service can be contacted.~~

~~The Quick Cards and The User Manual are in PDF format and require Adobe Acrobat Reader plug-in installed in your browser. This can be downloaded free of charge from the Adobe's homepage by clicking on the icon on screen.~~

3.93.8 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.

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

EXIT SESSION _____ **CONFIRM** _____
Goodbye, John Smith.

Close window

IMPORTANT:
To complete the log off process and prevent other users from accessing CONFIRM, you must close this browser and any other browser windows opened during this CONFIRM session.

Exit Internet Explorer

EXIT SESSION _____ **confirmit** _____
Goodbye, John Smith.

Close window

IMPORTANT:
To complete the log off process and prevent other users from accessing confirmit, you must close this browser and any other browser windows opened during this confirmit session.

Exit Internet Explorer

Figure [8888](#): 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.

Formatted

PROJECT OVERVIEW CONFIRM

Project details	
Project ID:	p0793814
Project name:	(New blank project)
Company:	FIRM AS
Description:	
Languages:	☒ English ☐ Spanish ☐ Norwegian ☐ Finnish ☐ Swedish ☐ Danish ☐ German ☐ Icelandic ☐ French ☐ Italian
Status:	
Deployment:	

Save

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

PROJECT OVERVIEW

p16169441

Project details																			
Project ID:	p16169441																		
Project name:	Nye funksjonaliteter 2																		
Company:	FIRM AS																		
Description:																			
Languages:	<table><thead><tr><th>Available</th><th></th><th>Selected</th></tr></thead><tbody><tr><td>Afrikaans</td><td>>></td><td>English</td></tr><tr><td>Albanian</td><td></td><td>Norwegian (Bokmål)</td></tr><tr><td>Arabic</td><td><<</td><td></td></tr><tr><td>Arabic (Algeria)</td><td></td><td></td></tr><tr><td>Arabic (Bahrain)</td><td></td><td></td></tr></tbody></table>	Available		Selected	Afrikaans	>>	English	Albanian		Norwegian (Bokmål)	Arabic	<<		Arabic (Algeria)			Arabic (Bahrain)		
Available		Selected																	
Afrikaans	>>	English																	
Albanian		Norwegian (Bokmål)																	
Arabic	<<																		
Arabic (Algeria)																			
Arabic (Bahrain)																			
Status:																			
Deployment:	WI (public)																		

Save

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

Figure 9999: 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	Working language(s) of the project. Here you can select the working language(s) of your project. The Available field contains all the languages that are supported in confirm it. The Selected field will contain the working languages of your project. You highlight a language in the Available field and click the arrow button to add it to the Selected 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 ————— p0793814

Test Project

Users in organization

OBJECT PERMISSIONS ————— p16163441

Nye funksjonaliteter 2

Users in organization

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

Name	-	R	W	D	Administrate project
firm	☒	☐	☐	☐	☐

Formatted

Formatted

Formatted

Formatted

Formatted

Formatted

Formatted

Formatted

Figure 10101010: Object Permissions

Field	Description
Name	The userid and the full name of the user (if entered in confirm) 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.
<u>Coder of opentext questions</u>	<u>The user has the permission to code open-ended questions.</u>

Formatted

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 **confirm** users.

All **confirm** users in your organization will be listed in *Object permissions*. The project administrator may set up other **confirm** 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 **confirm** with correct user keys of these users. Typically, an external user will send you his or her user key by email; this key is then copied and pasted into **confirm** 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 **confirm** 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*.

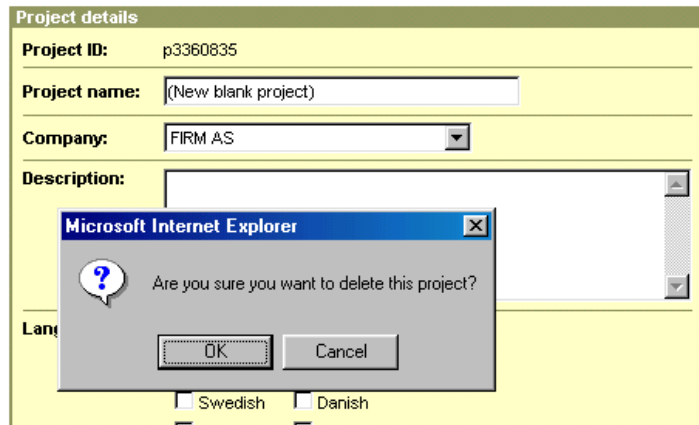


Figure [11111111](#): Deleting a project

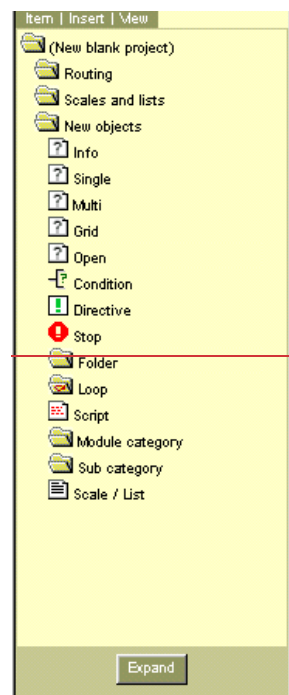
Before actually deleting, **confirmit** will generate the message above, warning you that you are about to delete a project.

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](#)[Figure 12](#)[Figure 12](#)[Figure 12](#)[Figure 12](#)[Figure 12](#)[Figure 12](#)[Figure 12](#)[Figure 12](#)[Figure 12](#). Note the three folders, *Routing*, *Scales and lists* and *New objects*.

Formatted



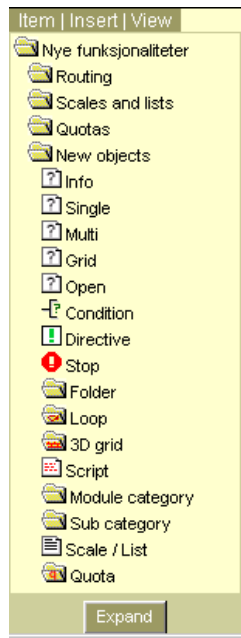


Figure 12121212: Questionnaire view

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 available routing-objects available.

5.2.1 Forms

There are currently five standard types of forms available:

Formatted

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 <code>^FTITLE^</code> where you want to insert this text. See 3.6.1.1)
Text	The form text. (In templates, you may use the primitive <code>^FTEXT^</code> 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 <code>^FCOMMENT^</code> 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	vWeight	ColWidth	BgColor	Punch	KeepPos	Other
Male							
Female							

Add | Add predefined | Clear | Delete rows | Save

Figure 13131313141314131313: 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. "none of the above" 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 previous text in the column, your new text will be appended to the end.

When in editing mode, you can move to the next field by using the tab-button, or by clicking in the column with the mouse. You can ~~directly~~-move 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 pasteing 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 checkbox *Only copy active* button decides whether you copy and entire row or not, and the checkbox *Overwrite* decides whether you insert new rows when you paste or not.~~

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

Formatted

Formatted

Formatted

	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.
<u>Coding of</u>	<u>[Multi with Opentext coding selected] Here you specify the Question ID of the Open-Text question that will be coded in the Multi question.</u>

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 101010101010101110 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.
Quota-var	[Will be implemented later in WI. At present CATI only].
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.

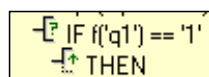


Figure 14141414141514141414: Conditional node

The expressions are written using the JScript-language. Expressions are described in Section [12421212121212121212](#) 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 12421212121212121212 confirm 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.

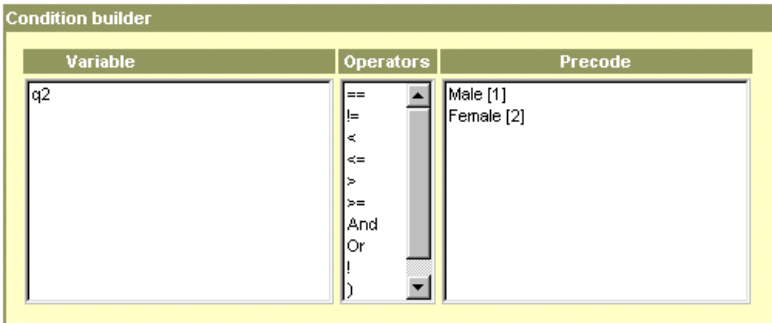


Figure [151515151516151515](#): Condition builder

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 which 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, ~~there exist~~ 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	{Not in use yet} 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. See separate manual.]

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 1616161616161716161616: 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):** See separate manual.

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 is automatically closed for respondents belonging to that sub sample, 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. Normally you would have to “wait” for 2000 respondents (2000*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.

Formatted: Bullets and Numbering

Formatted

Formatted

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

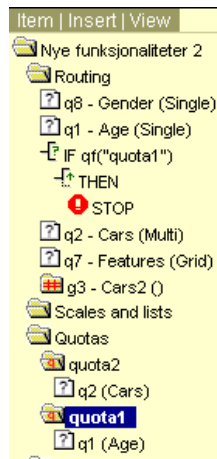


Figure 17: Adding quotas to a survey

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.**

Formatted: Bullets and Numbering

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.

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. **Remember that you have to recompile the database before you can edit the quota details.**

Formatted: Bullets and Numbering

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]

[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.
Quota count	The number of respondents who classify for the quota.

Email	Check here if you wish to receive an email when the particular quota is full.
-------	---

Table 18: Quota details

The quota details screen in Figure 19: Defining quota limit for one variableFigure 19: Defining quota limit for one variableFigure 19: Defining quota limit for one variableFigure 19: Defining quota limit for one variable shows how a quota counter definition may look for the Age quota. The interpretation of this quota is:

- Not more than 100 respondents in the age group 25 - 34 should answer the survey.

Formatted: Bullets and Numbering

If a **multi punch question** is used for a quota, there will be one field for each answer alternative. These answer alternatives will be valid for multi questions:

- 'Chosen', which corresponds to values '1' and upwards in the database.
- 'Not answered' which corresponds to '0' or 'NULL' in the database. 'NULL' means that the question is not asked.
- If you choose 'Any', the quota will be counted as full when any of the variables has reached the specified limit.

Formatted: Bullets and Numbering

QUOTA DETAILS

p16169441

Quota 'quota2'

Quota name:

Quota full email:

Volvo	BMW	Audi	Mercedes	Saab	Ferrari	Quota limit	Quota count	Email
[Any]	Chosen	[Any]	[Any]	[Any]	[Any]	100	29	☒ Del quota

Figure 20202020: Quota details for a multi question

Figure 20: Quota details for a multi questionFigure 20: Quota details for a multi questionFigure 20: Quota details for a multi questionFigure 20: Quota details for a multi question, shows that variable BMW has been quota limited to 100. **We would advise you to make a separate Quota for each variable in a multi question, in case you wish to set a limit for several of the variables, because the system will evaluate the function qf("quotaname") to true once one of the quota variables has been filled up.**

If a **grid question** is used for a quota, there will be one field for each answer alternative like in a multi punch question, with the scale as a dropdown for each of them. Again you have a choice between adding all the variables for quota defining and specifying only some of the variables.

— p16169441

Figure 21212121: Quota details for a grid question

- p16169441

Figure 22222222: Quota details for a grid question 2

We would advise you to make a separate Quota for each variable in a grid question, in case you wish to set a limit for several of the variables, because the system will evaluate the function `qf("quotaname")` to true once one of the quota variables has been filled up.

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 quota counter is only updated by the interview engine when new interviews are completed.

Formatted: Bullets and Numbering

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.

Formatted: Bullets and Numbering

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. When 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.

Formatted: Bullets and Numbering

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 `qi("quotaname")` returns true if the specified quota is full.

The user is hence given a flexible way of handling full quotas. Alternatively the quota counters may be used to create precode masks for preceding questions in the survey. E.g. there may be quotas on a multi question like brand awareness. When the quota for Coca-Cola is full, questions about rating the brand will not be asked for Coca-Cola.

Formatted: Bullets and Numbering

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.....".

Formatted: Bullets and Numbering

Formatted

5.2.9 3D – Grid

Formatted: Bullets and Numbering

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](#)

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

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

Formatted

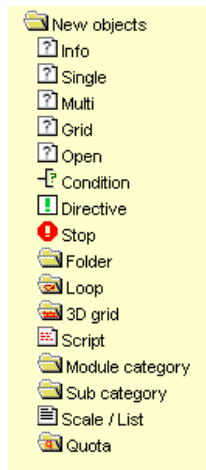


Figure 232323232423: 3D grid object in the New objects folder:

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

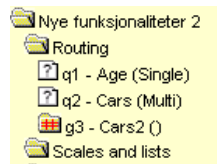


Figure 242424242524: 3D grid in the questionnaire routing

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](#).

Formatted

Formatted

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 end up as the first object in the folder.** The next object you can drop on the first object in the folder.

Formatted

Formatted

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.

Figure 252525252625: Multi punch question menu in a 3D grid

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

Figure 262626262726: 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.

	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 272727272827: 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 **legal allowed** destination. An example of an **illegal-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 **greyed-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 19191919191918: 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: Template

English

Name: Test Age

Description:

Save

Figure 28282828281817171717: 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 20202020202019: 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 [2929292930291918181818](#): Templates in Initiate New Project

6 CONFIRMIT Scripts

6.1 Introduction

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

- `if` conditionals for controlling the flow through the questionnaire,
- `if` for filtering the lists displayed in questions and the iterations in loops based on previous answers (precode masks)
- `if` form elements for text substitution (piping),
- `if` for custom validation of user input and
- `if` 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 21212121212120: Values returned from `f("qID")` for different question types

6.2.1.1 Single

sex - Gender		
Text	Answers	Properties
T		
Answers		
English	Precode	Weight
Male	M	
Female	F	

Figure 3030303031302019191919: Answer list of single question "sex"

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

6.2.1.2 Open Text

Name
What is your name?
John Smith

Figure 3131313132312120202020: 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 323232323322221212121: 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 333333334332322222222: 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 [12APPENDIX A: confirmit Script Reference](#) [Error! Reference source not found.](#) [Error! Reference source not found.](#) [6.8.5.1.5](#) for more details).

Favorite program

What is your favorite program on these channels?

ABC NYPD Blue

CNN

NBC Friends

Figure 3434343435342423232323: 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 35353535362525242424: 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						

Add Add predefined Clear Delete rows Save

Figure 3636363637362625252525: 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 3737373738372726262626: 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 383838383838: 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 precodes is checked off, 0 if it is not.
Multi with Ordered checked	A string with the value (rank) given to the item with this precodes.
Multi with OpenText checked	A string with the text answered for the item with this precodes
Grid	The precodes (from scale) of the answer to the item with this precodes.

Table 22222222222222: 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](#)[Figure 39](#)[Figure 39](#)[Figure 39](#)[Figure 40](#)[Figure 32](#)[Figure 28](#)[Figure 27](#)[Figure 27](#)[Figure 27](#).

hours - Hours watching		
Text	Answers	Scale
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 3939393940392827272727: 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 [special-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.

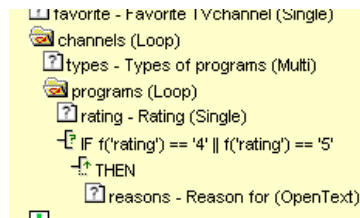


Figure 4040404042403028282828: Nested loops

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 41Figure 41Figure 41Figure 41Figure 43Figure 34Figure 31Figure 29Figure 29Figure 29Figure 29](#)).

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 4141414143413129292929: 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 42Figure 42Figure 42Figure 45Figure 35Figure 33Figure 30Figure 30Figure 30Figure 30](#)). 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 4242424245423330303030: 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 23232323232322: 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). `Linc(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`.

Formatted

Formatted

Formatted

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".

You may also use `f("qID").toNumber() <= 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(), nnsset()

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

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

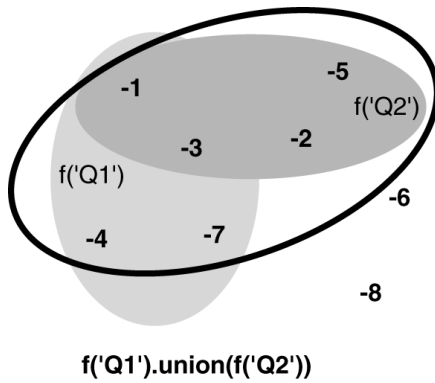
`nset(n)` returns a set populated with the members 1, 2, 3, ..., n.

`nnsset(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 `nnsset(8,12)` in the Precode mask property of that question.

6.4.3 .union()

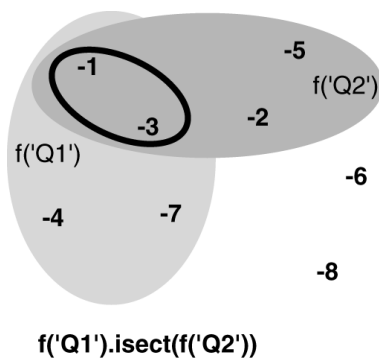
`l.union(r)` returns a new set that is the union of `l` and `r`, that is a set consisting of all precodes in either `l` or `r`. If you want to add precodes to a set in a precode mask, then use .union()



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

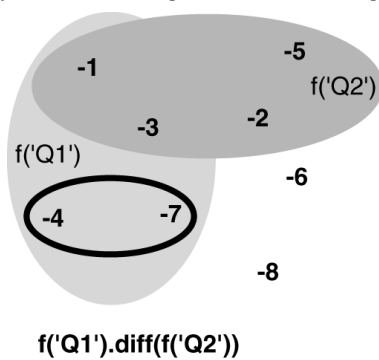
6.4.4 `.isect()`

`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 .



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()`



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 includes tabs for "Text", "Answers", "Properties", "Preview", "Tracking", "Languages", and "Save". The "Text" tab is active, showing a text area with the content "Hello, ^f("name")^.
 Which TV channels do you watch on a regular basis?". Below the text area is an "Instruction" field, which is currently empty. The editor has a yellow background and a green border.

Figure 4343434347433531313131: 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 4444444448443632323232: "TVchannels" question with text substitution.

Question type	Returned from $f(qID)^{\wedge}$
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 $qID_precode_other$ (see 5.2.1.4))
Loop	The text of the loop member corresponding to the iteration with precode qID in the current language.

Table 24242424242423: Texts returned from $f(qID)^{\wedge}$ for different question types

Question type	Returned from $f(qID)[precode]^{\wedge}$
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 25252525252524: Texts returned from $f(qID)[precode]^{\wedge}$ 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.

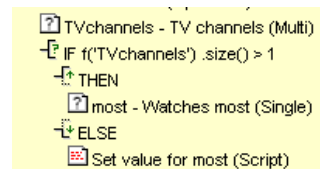


Figure 4545454549453934343434: If-then-else construction

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<precodes.length; ++i)
    {
        var precode=precodes[i];
        if(f("qID")[precode]=="6")
            s.add(precode)
    }
    return s;
}
```

Formatted

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.

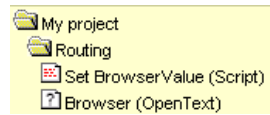


Figure 35: Setting a value for a hidden question

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 ~~most~~ recently made changes.
- Update existing database (rebuild) – the system will update the whole database, ~~i.e. it will run through the whole questionnaire.~~ You would e.g. use “rebuild” on your projects after a system upgrade or after some bugfixing has been carried out.
- Create new database – the system will create a completely new database. ~~All existing~~ All existing old 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 ~~the~~ 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://www.survey.confirmit.com/wi/pXXXXXXX/i.asp>

~~If you need to generate new WI files for projects created before the 21st of February 2000, we recommend that you use the Old WI Generator.~~

~~For projects initiated after the above mentioned date, you may use the New WI Generator.~~

Web Interview Generator
Generates web interview and control files.

☒ Include progress bar.
☒ Reserve a fixed area for error messages.
Survey design: FIRM.Hjemmeside
Body attributes (will override template):
☐ Compact layout.
☒ Generate back button.
☒ Allow user to modify answers after initial submission.
☐ Disable "answer required" validation.
☐ Limited survey (Requires a respondent list).
☐ Generate login page (User ID/password must be provided with respondent data).
☐ Generate pop-up survey. Pop-up width: Pop-up height:
☐ Generate response editor.
Default language: French


[Help...](#)

~~Login only: Figure 50464035353535: Web Interview Generator options (Old)~~

Formatted

Field Code Changed

Web Interview Generator

Generates web interview files.

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.

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. Pop-up width: Pop-up height:

Default language:

 [Help...](#)

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 4647474751474136363636: Web Interview Generator options (New)

Formatted

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.

Formatted

Formatted

Formatted

Formatted

Formatted

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.

Login only:

7.2.2.5 Body Attributes

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

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

Formatted: Bullets and Numbering

7.2.2.6 Compact Layout

This option controls the way in which the questions are displayed. By checking this, they are displayed one after the other using the entire page width instead of one per line.

Formatted: Bullets and Numbering

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 Modification of Previous Answers

Formatted: Bullets and Numbering

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

Formatted: Bullets and Numbering

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

Formatted: Bullets and Numbering

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

Formatted: Bullets and Numbering

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.

7.2.3.57.2.3.6 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.

Formatted: Bullets and Numbering

7.2.3.67.2.3.7 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).

Formatted: Bullets and Numbering

Login-only:

7.2.3.7Disable “Answer Required” Validation

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

Formatted: Bullets and Numbering

7.2.4 User Defined Error Messages

Formatted: Bullets and Numbering

Languages that are supported in **confirmit** come with a set of standard error messages. These messages may be overridden for a server installation. This feature is available for **confirmit** administrators only. Go to **confirmit admin > Messages**, and you will be presented with a screen shown in [Figure 47: Message editorFigure 48: Message editorFigure 48: Message editorFigure 48: Message editorFigure 52: Message editorFigure 41: Message editor](#). You choose the appropriate language and in the field **Edit local text** you enter the message you want to appear for the respondent. Then click **Save**.

Formatted

Formatted

Formatted

Language: English Save

Standard forms

General

Forms used in all generated interviews.

"Finished" page title

Displayed text	Survey Completed!
<i>English, system provided</i>	
System provided text	Survey Completed!
Edit local text:	

"Finished" page text

Displayed text	Thank you very much for your time.
<i>English, system provided</i>	
System provided text	Thank you very much for your time.
Edit local text:	

Figure 4748485248: Message editor

7.2.47.2.5 Options for Generating Limited Surveys

Formatted: Bullets and Numbering

7.2.4.17.2.5.1 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.

7.2.4.27.2.5.2 Generate Login Page

Formatted: Bullets and Numbering

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 8.1Preparing the Respondent List.

7.2.5.7.2.6 Pop-Ups

Formatted: Bullets and Numbering

7.2.5.17.2.6.1 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.

7.2.5.27.2.6.2 An Example of a Script for Pop-up Surveys

Formatted: Bullets and Numbering

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~~ 7.2.6.3 .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 ~~7.2.6.27.2.6.27.2.6.27.2.6.27.2.6.27.2.6.27.2.6.27.2.5.2~~, 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~~ 7.2.7 Default Language

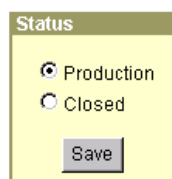
This option is only displayed for multi-language surveys. The language selected here becomes the default used when no language information is available in the request used to access the interview or in the respondent list.

Formatted: Bullets and Numbering

7.3 Project Status

You can set the project status on this page.

PROJECT STATUS ————— p0793814 ———



The screenshot shows a web form titled 'Status' with a yellow background. It contains two radio button options: 'Production' (which is selected) and 'Closed'. Below these options is a 'Save' button.

Figure ~~4849494954494237373737~~: 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 ~~in~~ 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 ~~49505050555043383838~~: Project log

7.5 URL Setup

You have ~~a possibility~~ 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	Link preview
Help link:		
End link:		
Save		

Figure ~~50515151565144393939~~: 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 262626262625: URL setup

Usually, the End link must be defined ~~between before you compile the survey database the processes of Compilation and Generation of WI: Compile prod. Env. > URL Setup > Generate WI Compile prod env.~~

~~You will have to Generate WI anew in case you make any changes in URL Setup.~~

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 272727272726: 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 [PropertiesProperties](#).

Formatted: Font: Bold

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 ~~was~~ 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
email

PREVIEW OPTIONS
Max no of preview records

EMAIL
Mail format
Send as HTML ☐
Sender
Subject
Body
Include link ☐

Setup

SELECTION CRITERIA

Respondent status

Any

noOfEmailsSent

name

email

PREVIEW OPTIONS

Max no of preview records

10

EMAIL

Mail format

MIME encoded

Send as HTML

☐

Sender

john@anywhere.com

Subject

A Survey!

Body

Dear respondent !

.....

Include link

☒

Next >>

Figure 5152525257524540404040: 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 ~~these~~ 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.
Other fields: 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	Not in use yet. The interview is stopped for a certain segment of respondents due to a full quota.

Table 282828282827: 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 possible 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:

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.

Formatted

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 **confirm** 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 EmailingFigure 53: Batch EmailingFigure 53: Batch EmailingFigure 53: Batch EmailingFigure 58: Batch EmailingFigure 46: Batch Emailing.

Formatted

Formatted

Formatted

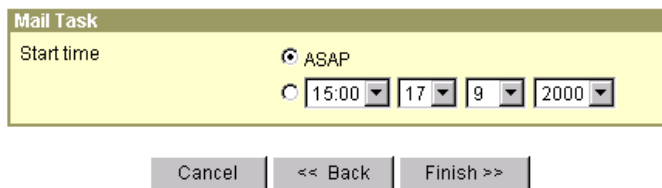


Figure 525353535853: Batch Emailing

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.

Formatted

Formatted

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 535454546054: Task properties for batch emailing

Formatted

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 ~~for~~ 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/pXXXXXXX/i.asp?l=0> (for English)

<http://www.yourconfirmitserver.com/wi/pXXXXXXX/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 *I* in the survey, as mentioned above (see 8.4.1 Open Multilingual Surveys).
- In multilingual surveys with login-page (see [7.2.5.27.2.5.27.2.5.27.2.5.27.2.5.27.2.5.27.2.5.27.2.4.2](#)) the respondents are presented with a drop-down where they can choose language.

9 Reporting

9.1 Introduction

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

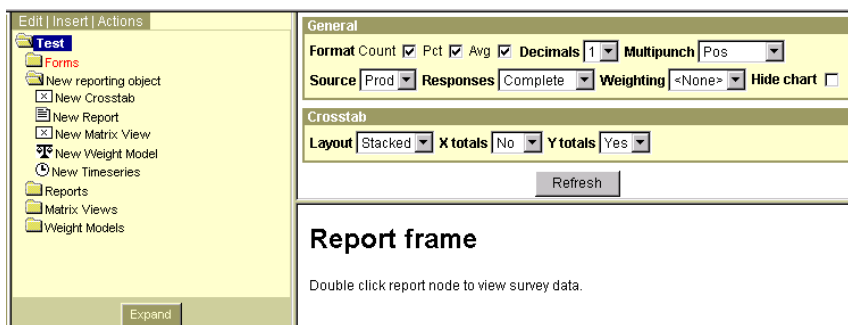


Figure 545555561554641414141: Initial reporting screen

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 the Figure 42.
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?"

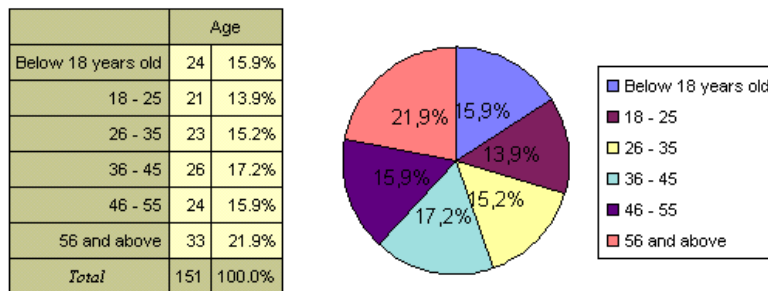


Figure 5556565662564742424242: Reporting a single question with Excel Chart Component

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 by 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;

Formatted

Export properties

Project IDp16169441

Report nameTest

PPT templateconfirmit 01

Export filesMS Excel and MS Powerpoint

DatabaseProduction

LanguageEnglish

Recipient emailjohn@anywhere.com

Comment

Start timeSchedule for later execution

ASAP

Schedule for later execution

Figure 565757576357: 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 Figure 58: Task settings for an ordered report Figure 58: Task settings for an ordered report Figure 58: Task settings for an ordered report Figure 64: Task settings for an ordered report Figure 51: Task settings for an ordered report .

Table 30: Report Export

Formatted

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 575858: Task settings for an ordered report



Figure 585959665948434343: 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 43 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:

- **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).

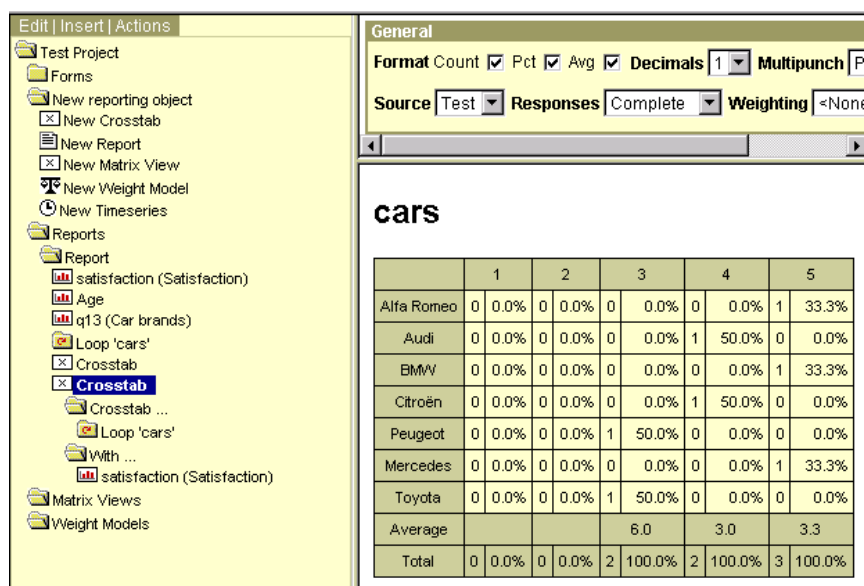


Figure 6162626271625246464646; Loop in crosstab

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 6263636372635347474747: Weight model properties

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

Figure 636464647364: 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.

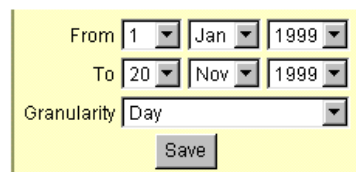
It is not possible to apply weighing on questions that were answered only by part of the respondents (filtered questions).

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



From 1 Jan 1999
To 20 Nov 1999
Granularity Day
Save

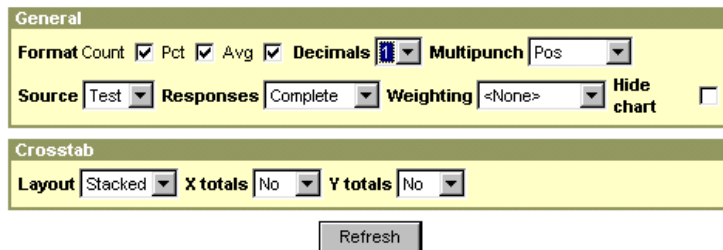
Figure 6465656574655448484848: Time series properties

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.



General
Format Count ☒ Pct ☒ Avg ☒ Decimals ☐ Multipunch Pos
Source Test Responses Complete Weighting <None> Hide chart ☐
Crosstab
Layout Stacked X totals No Y totals No
Refresh

Figure 6566666675665549494949: 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 32323232323229: 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
Global setting

Multipunch
Global setting

Deleted
☐

Chart options

Chart type:
Use default

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
Use Default

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 6667676776675650505050: 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 3333333333330: Excel chart component options

9.8.2.2 First Impression Chart Component

Formatted: Bullets and Numbering

The following options are available for you if you have chosen First Impression in user settings (see 3.2 User Settings):

Chart options			
Type	Use Default	Min Scale	
Content	Use Default	Max Scale	
☐ Hide chart ☐ Stacking ☒ Legend			
☐ 3D ☐ Data in Rows			

Figure 5151: First Impression Chart options

Setting	Description
Chart options First Impression Component	
Type	Defines the chart type to be used. The most normal chart types like bars, pies, lines and areas are supported among some other.
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.
Hide Chart	Use this check box to hide the charts in the report frame.
3D	The chart can be viewed in 3D.
Stacking	The chart is stacked.
Data in rows	Uses data in rows instead of data in columns. This is the terminology of the chart component used. Excel uses the opposite terminology: Series in columns and series in rows.

Table 31: First impression component options

9.8.2.39.8.2.2 PowerPoint Options

Formatted: Bullets and Numbering

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.

Formatted

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.

Formatted

Power Point options

Header text

Object type

Chart

Slide layout

Only object

Text

General

Chart

Ppt

Publisher

Filter

Masks

Save

Power Point options

Object type

Chart

Slide layout

Only object

English

Header

Text

Norwegian (Bokmål)

Header

Text

Copy Publisher texts

Figure [6769696978615852525252](#): 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 343535353532: PowerPoint options

9.8.2.49.8.2.3 Filter Options

Formatted: Bullets and Numbering

See 9.9 Filters.

Filter options
Active filters

Figure 68707070796259535353: Filter options

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

Table 353636363633: Filter options

9.8.2.59.8.2.4 Answer Mask/Scale Mask

Formatted: Bullets and Numbering

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 69717171806360545454: 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 36373737373734: Answer mask/Scale mask options

9.8.2-69.8.2.5 Crosstab Options

Formatted: Bullets and Numbering

Crosstab options

Layout

X Totals

Y Totals

Global setting ▼

Global setting ▼

Global setting ▼

☒ Table content is average of first node

Figure 7072727281646155555555: 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 Crosstab.... folder. The variable must be weighed to get average values.

Table 37383838383835: Crosstab options

9.8.2.79.8.2.6 Loop Options

Formatted: Bullets and Numbering

Loop options

☐ Show average ☐ Show best ☐ Show worst

☒ Show all iterations

Show these iterations:

☐ Alfa Romeo ☐ Audi

☐ BMW ☐ Citroën

☐ Peugeot ☐ Mercedes

☐ Toyota

Figure 7173737382656256565656: Loop options

Loop options

Loop id: cars

☐ Ignore 'cars'

☒ Show all iterations

Show these iterations:

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

☐ Toyota

Figure 72747474836663575757: 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 38393939393936: 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.

Filter

Filter:

26-35

Form/Variable:

q3 (How old are you?)

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

Save

Figure [73757575846764585858](#): 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 graphs** 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-internally in-house**, the location **where reports are published- 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 **chartsgraphs** to be published, open the relevant report form 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 **do make** the adjustments you **need 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.

Formatted

Header text	Content
Age	Only table

The table shows that....

General | Chart | Ppt | **Publisher** | Filter | Masks | Save

Publisher options

Content Only chart ▾

English

Header

Text

Norwegian (Bokmål)

Header

Text

Copy PowerPoint texts

Figure 7476767685686559595959: 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/[graphscharts](#) being published. Your text fonts and colors can be defined by standard HTML commands. You can also choose whether to show tables, [graphscharts](#) 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:

Published Reports for project "p0793814"

Active	Run now	Report	Database	Lang	Creator	Resp	Failed	Frequency	Targets	Log Users	Template	One page	Last run	Del	Remove from web
☐	☐	Test	0	johns	NIA	No	Frequency	Targets	Log Users	None	☐	Never	☐	Remove	

Save

Figure 757777786696660606060: The Publisher screenshot

The settings available to you are:

Setting	Description
Active	Tick this to make the report available to users
Run Now	By activating this, the report will be updated on the next engine run, even if the frequency setting requirements is <u>are</u> not met (the FIRM engine normally runs every 5 minutes)
Report	The name of the report being published
Database	Can be either Test or Production
Lang	Language (based on your <i>Personal Settings</i> at the time of publishing)
Creator	Creator's name
Resp	Number of respondents of latest publishing
Failed	Error notification
Frequency	Allows you to choose how often the data is to be updated. See below
Targets	Settings for how the report is to be published
Log	Log of publishings
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
Delete	Allows to delete a published report
Remove	The report is being removed from the web.

Table 39404040404037: 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 **Frequency**: **confirm** it 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 (minutes). 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.

Published Report generation frequency for report 'R1'

Type	Value	Delete
Count	5	☐
Time	Hourly	☐

[Back](#)

Figure 767878877067616161: Frequency

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 graphchart 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>		☐
<Select one>		
Email		
Cryptic link		
Protected link		

Save Add Del

[Back](#)

Figure 77797979887168626262: 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 graph after publishing. In all these cases you must publish once more. This, however, does not apply to changeing 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 towards 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 40414141414138: 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.** A new matrix view is created by dragging a “New Matrix View”-object from the “New reporting objects” folder into the “Matrix Views” folder. 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.

Formatted: Bullets and Numbering

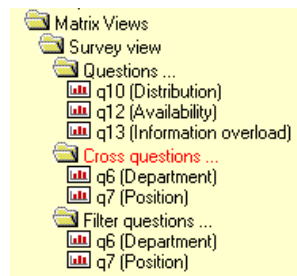


Figure 78808080897269636363: Matrix view configuration

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 ~~that~~ 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. Dark color of A 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 ed between count, percent and index (average) by changing the “ <i>Format</i> ”-option.
Change matrix size	The matrix size can be changed by changing the “Screen”-option Changing the “Screen”-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 ~~414242424239~~ **414242424239**: Matrix view operations

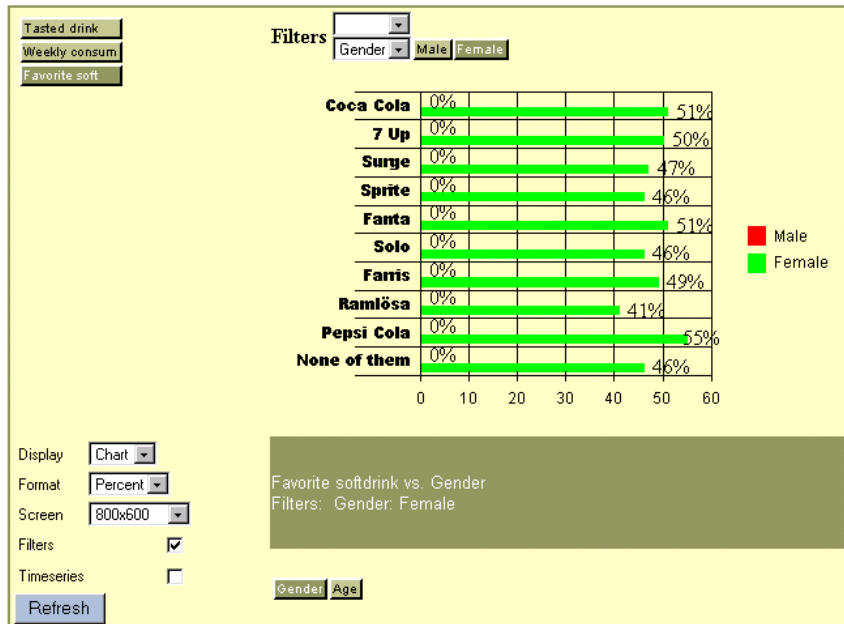


Figure 7981818190737064646464: Matrix view

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. [To build advanced expressions correctly, it is necessary to know how the UPDATE statements of the expressions are created in CONFIRMIT. This is best illustrated by two examples:](#)

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

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

Formatted

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
<u>.getValue()</u> <u>.toString()</u> <u>.valueOf()</u>	The ADO value of the ADO field method value(). This is a string except for interview_start and interview_end that return date.
<u>.int()</u>	parseInt (getValue ())
<u>.str()</u>	New String ((getValue ()))
<u>.date()</u>	New Date (getValue ())
<u>.float()</u>	parseFloat (getValue ())
<u>.int()</u>	parseInt (getValue ())
<u>.in(param1, param2,...)</u>	True if getValue() equals at least one of the parameters.
<u>.weight()</u>	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.

Text
Answers
Properties
Preview
Permissions
Tracking
Languages
Save

Answers

Norwegian	Expression	Precode	Weight
Indoor	sprm2.In('Y', 'b', 'k')	I	
Outdoor	sprm2.In('m', 'Y')	U	

Add
Save
☒ Overwrite
☒ Only copy active column

Expression

sprm2.In('I', 'b', 'k')

Condition builder

Column	Operators and methods	Legal values
sprm2	== != < <= > >= && ! parseInt(date(average(.In(	Livingroom [l] Bedroom [b] Kitchen [k] In the mountain [m] At sea [s]

Figure 8082828291747165656565: 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.

FIELD CHOOSER TEMPLATE EDITOR p0797203

Production database templates.

Template list

Demographics
Short file

Edit Copy Delete New

Template details

Template name
Demographics

Verbatim key field
 (default: respid)

Std fields

responseid
respid
status
interview_start
interview_end

Export fields

Age

User defined fields

cars_tested_1
cars_tested_2
cars_tested_3
cars_tested_4
cars_tested_5
cars_tested_6
cars_tested_7
cars
satisfaction

Recoded fields

Rec_age

> >> < <<

^ v

Save

Figure 8183838392757266666666: 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. <table> <tr> <th>Field</th><th>Description</th></tr> <tr> <td>responseid</td><td>Respondent identification given at the time the respondent starts answering the survey.</td></tr> <tr> <td>respid</td><td>Respondent identification given at the time the respondent is added to the respondent list.</td></tr> <tr> <td>status</td><td>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.</td></tr> <tr> <td>interview_start</td><td>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.</td></tr> <tr> <td>interview_end</td><td>The time when the respondent finished the questionnaire, either from a STOP node or by reaching the end of the questionnaire.</td></tr> </table>	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.
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 101010101010101110 Recoding Data).												
Export fields	These are the fields that will be included in the exports done with this template.												

Table 424343434340: 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 p0793814

Export survey data.

Export format

☒ Excel file
☐ Delimited text file, comma separated, quoted
☐ Delimited text file, tab separated, not quoted
☐ Fixed-width text file
☐ Definition file for fixed-width text file
☐ Open-end questions

Export properties

Project ID: p0793814

Export data: Complete responses

Export template: All fields

Export language: English

Recipient email address: John@somewhere.com

Start time: ☒ As soon as possible
☐ At midnight
☐ At: Year 2000 Month 2 Day 14 Hour 13:00

Place request

Figure 8284848493767367676767: 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.) <table border="1"> <thead> <tr> <th>Field</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Form ID</td><td>The Question (form) ID</td></tr> <tr> <td>Var ID</td><td>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.</td></tr> <tr> <td>Start Col</td><td>The start position of this variable in the export file, counting characters from left to right.</td></tr> <tr> <td>End Col</td><td>The end position of this variable in the export file, counting characters from left to right.</td></tr> <tr> <td>Punch</td><td>Will list the possible punches (precodes) possible for these answers corresponding to the texts in the Answer Text field.</td></tr> <tr> <td>Question Text</td><td>The text of the question.</td></tr> <tr> <td>Answer Text</td><td>The text of the answer. (Corresponding to Punch).</td></tr> <tr> <td>Data Type</td><td>Question type (or system generated type).</td></tr> <tr> <td>Level ID</td><td>Loop level. Gives the id of the loop this question is inside.</td></tr> </tbody> </table>	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.
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 11.111.111.111.111.111.111.112.111.1)																				

Table 4344444444441: 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 11.111.111.111.111.111.111.112.111.1)
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 shall is to be put in the task list. Either immediately, by midnight or at a specific time.

Table [44454545454542](#): Data export properties

Click *Place Request* to send the request to Task Management (see 3.4).

11.3 SPSS Export

SPSS DATA EXPORT ————— p0795400 ———

Export survey data to SPSS.

Export properties

p0795400

Report data:

Report language:

Recipient email address:

Batch job comment:

Start time:

☒ As soon as possible

☐ At midnight

☐ At: Year

Month

Day

Hour

Figure [8385858594777468686868](#): 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 454646464643: SPSS Export properties

Click *Place Request* to send the request to Task Management (see 3.4).

11.4 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~~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) ————— p0795400 ———
Export questionnaire to MS Word format (last compiled version).

Export properties

p0795400

Report data

Production data

Report language

English

Recipient email address

John@somewhere.com

Batch job comment

Start time

☒ As soon as possible
☐ At midnight
☐ At: Year

2000

Month

2

Day

14

Hour

16:00

Place request

Figure ~~8486868695787569696969~~: MS Word Export

Click *Place Request* to send the request to Task Management (see 3.4).

12 APPENDIX A: confirmit Script Reference

12.1 Accessing Survey Variables

f function
Description Accesses survey variable values.
Declaration $f(qID, [expr1[, expr2, \dots]])$

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](#) ~~Error! Reference source not found.~~

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. [\) for coded variables.](#)

Returns an array of all labels (texts from answer list) in the current language [for compounds.](#)

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 [7.2.3.47-2.3.47-2.3.47-2.3.3](#)). 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, 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, 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.confirmat.com/wi/' + CurrentPID() + '/i.asp?r=' + CurrentID() + '&s=' + CurrentSID() + '&l=' + CurrentLang()
```

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

```
http://www.confirmat.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](#) **Error! Reference source not found.** 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 *intimate 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 (see **Error!** **Reference source not found.**).

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 8587878987: 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 to 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: confirmit LANGUAGE CODES

Language code	Language	ISO 639 Language code
0	English (Default)	en
2	Norwegian	no
3	Swedish	sv
4	German	de
5	French	fr
6	Spanish	es
7	Finnish	fi
8	Danish	da
9	Icelandic	is
10	Italian	it
12	Dutch	nl
13	Flemish	nl_be
14	Portuguese	pt
15	German - Austria	de_at
16	Latin American Spanish	es_la
17	Brazilian Portuguese	pt_br

More languages will be added on demand.

14 APPENDIX C: 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