

Welcome

VoxcoTab User Guide.

Version: 2.10

Date: May 21, 2013



All Rights Reserved - VOXCO

Contacting Voxco: support@voxco.com

Table of Contents

Getting Started	12
Opening an existing study:.....	13
Importing a study :.....	13
New Study	13
Definitions	13
Studies without levels.	13
Studies with levels.....	14
Declaring the structure of a new study.....	14
Defining data record types	15
Defining levels.....	16
Importing a study.....	17
Overview - Importing a study	17
Importing from Conversoft	17
Importing from COSI.....	18
Importing from Quancept.....	18
Processing specific to Quancept	19
Importing from SPSS.....	19
Importing from TripleS.....	20
Importing Excel (xls or.xlsx).....	20
Opening a study.....	21

Properties.....	21
Deleting a Study	21
Overview - Variables	23
Variable Attributes	24
Manipulating variables	25
Editing variable attributes.....	27
Formats	28
Overview	28
Assigning a Format.....	28
Creating a Format.....	28
Automatic format generation	30
Examples of Formats	30
Functions and Operations.....	31
Overview - Functions.....	31
Input function (ASCII)	32
Input CSV function.....	34
Import function	35
Import CSV function.....	36
Arithmetical Operations	37
Logical Operations.....	38
Rules and behaviour:	38
Recoding function	39

Table Of Contents

NBRESP (no. of responses) function	43
NBVAL function - Number of non-missing values for a dimensioned variable	44
SUMVAL (sum of values) function for a dimensioned variable	44
Dicho1 function - Distribute a Multiple variable to a dimensioned Single variable.....	45
DICHO2 function - distribute a dimensioned variable to a Multiple variable	45
NBLEV function.....	46
SUMLEV function	47
CONST function	48
TABLE function.....	49
CONCAT function	50
MIN and MAX functions.....	51
RANK and RANKNF functions - Case ranking.....	52
GETLEV function - Get Level	52
CONV functions.....	53
Text-type resulting variable	53
Logical-type resulting variable	54
Single-type resulting variable	54
TIME and DATE functions	56
CurDate and CurTime functions.....	56
Year, Month, Quarter, DayWeek, DayYear and Week functions	56
Difference between dates	57
AddTime function : Add time duration to a date.....	58

CHAIN function	59
SUBSTR function	60
POSTR function.....	60
TmutS function.....	61
CPL function (procedure).....	61
External Tables	62
Concept.....	62
Reading data.....	62
Bases.....	62
Execution	64
General	64
Processing of missing values (NA).....	64
Process	64
Cache file	66
Preview.....	67
Frequency Tables	68
Overview - Templates.....	69
Managing Templates.....	70
Template Options.....	71
Managing Option fields	71
Managing Table Requests	73
Table Request	74

Table Of Contents

Table Request configuration settings	74
About Stages and cross-tabulation.....	76
Table Request variables - attributes	77
Execution of a table request	81
Constraints on a table request	81
Execution.....	81
Component table breakdowns.....	83
Percentages (example)	85
Presentation of Results	86
Preview.....	86
Table Display Settings.....	87
Stubs	87
Banners	88
Counts	89
Means	93
Page Settings	95
Page Index	98
Printing tables.....	99
Printing several requests.....	99
Page Preview	99
Gamma (Goodman et Kruskal) and Chi Squared tests.....	100
Exporting tables.....	101

Table Requests at the Folder Level	102
Row and Column Operations	103
Table Requests	103
Assigned Template	104
Execution	104
Export	105
An Export Request	105
Weighting	111
Overview	111
Weighting request	111
Options (tab)	112
Criteria (tab)	113
Execution	114
Overview (Procedures)	115
Description	115
Creating a procedure	115
CPL Language	117
Symbols and notation	117
Definitions	117
System symbols	117
Variable types and implicit conversion	118
Types of variable	118

Table Of Contents

Rules for conversion between types.....	118
Structure of a CPL procedure	119
Basic CPL instructions.....	121
Dimension of instructions (CPL).....	121
Assignment instructions	122
Other forms of assignment (shortform)	123
Other functions.....	124
Conditional expressions and control structures (CPL)	126
Conditional expressions	126
Control Structure.....	127
Special instructions.....	131
exit and stop.....	131
reset	131
debug and write.....	131
fwrite	132
Example of a procedure.....	134
Predictive Model	135
Overview.	135
Implementation.	135
Building the predictive model.....	135
Simulation	138
Overview.	138

Simulation request.....	138
Export CoTab Reader.....	141
Overview.....	141
Creating a Cotab Reader for a given study..	141
Running CoTab Reader.....	141
CoTab Reader - Overview.....	143
Attributes of Variables.....	144
Viewing an Existing Table.....	144
Table Settings.....	146
Page Settings.....	147
Setting up a New Table.....	147
Templates.....	151
Dimensioned Variables.....	153
Requests from a Folder.....	155
Reports (Excel).....	157
Overview.....	157
Instructions.....	157
VoxcoTab Main Menu.....	161

VoxcoTab - Overview

VoxcoTab is a software application for creating statistical cross-tables. It operates on PCs running Windows (Millenium, 2000, XP) equipped with a version Microsoft .net 2.0.

The source data to be processed can be in various formats:

- ASCII files including multiple records.
- Files produced by survey software: Conversoft, Quancept, Triple S,
- Files produced by Cosi software.
- Excel

There is no limit on the volume of source data which can comprise one or several levels of cases (statistical units).

VoxcoTab comprises 2 major process areas:

- Creation of variables, and their configuration from the source data file, including all of the necessary recoding tools. This area involves the description of data which is useful at an individual case level.
- Creation of tables, in which the preceding data is broken-out and cross-tabulated. Many tables can be produced at a time, configured with multiple options.

Supporting modules are provided for:

- Sample adjustment
- Data export in various formats
- Predictive model
- Reports

You can switch between modules instantly, enabling you to define variables, then define tables, execute them, take a look at the tables, go back and modify variables, return to the tables etc.

Multiple file operation: All VoxcoTab operations (tables, exportation etc.) can operate with multiple files. For example, you can have a multiple wave study in which the files are processed sequentially as a single concatenated file. All you have to do is specify a list of these files in the file selection window.

Ergonomics: If you are familiar with Windows, you will recognize your customary interface: menus, icons, mouse right-click, keyboard shortcuts etc. Use of a mouse is obligatory.

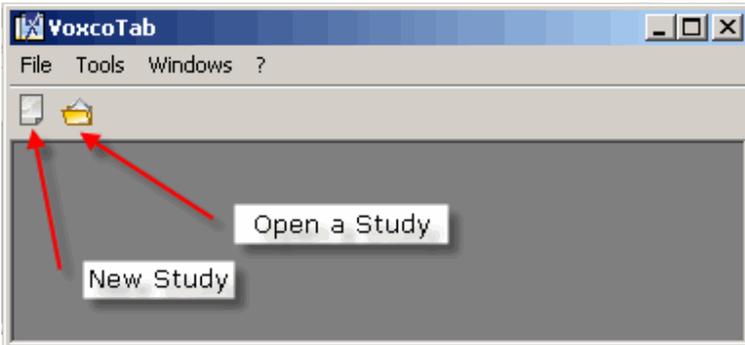
VoxcoTab is designed to provide project managers with a reliable and powerful tool that is easy to use. Its recent design (2004) ensures that it is capable of adapting successfully to evolving trends in information technology (Windows, Internet.).

Studies

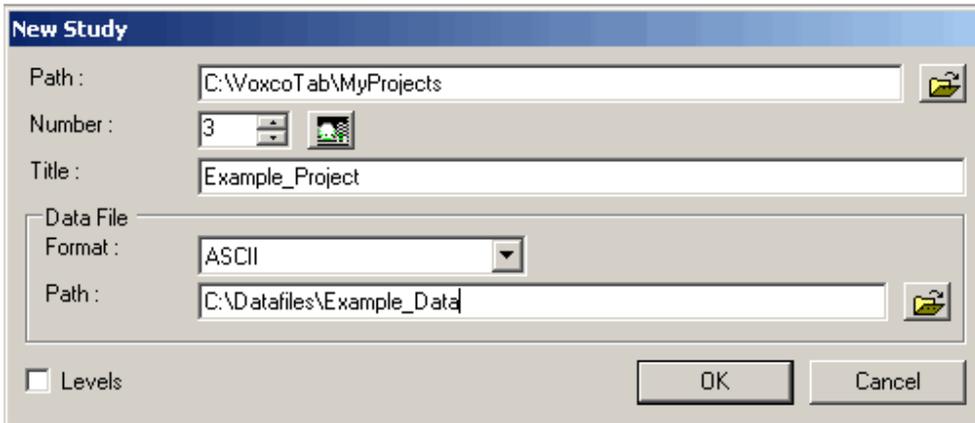
Getting Started

A study consists of a user-controlled process (involving variables, tables and tools) which is applied to a data file.

When you launch VoxcoTab, the first menu asks you whether you wish to create, import, open or delete a study.



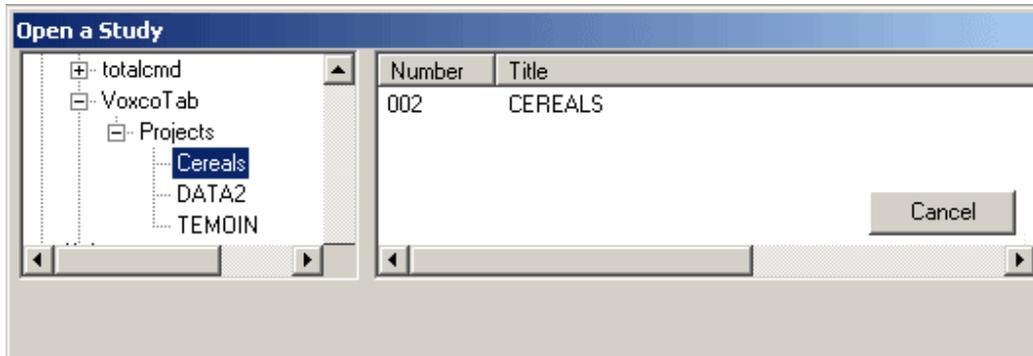
When creating a study you must specify its path and title, together with a study number which VoxcoTab then incorporates into the names of all the study files.



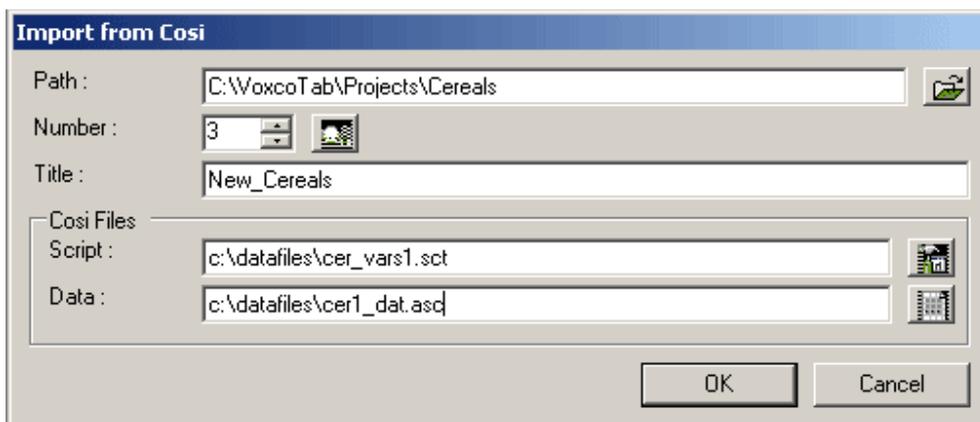
You must also indicate whether the study involves levels.

If you know it already, you can specify the name of the source data file. If not, you can provide this at a later stage.

Opening an existing study:



Importing a study :



You can save the study that you are working on at any time, or save it under another study number. Furthermore, whenever you close a study, you will be prompted whether you wish to save your work.

New Study

Definitions

You can configure a new study to have levels or not. A study with levels has several kinds of cases, while a study without levels has a single kind of case.

Studies without levels.

In a study without levels, the source data file to be processed may have zero, one, or several data record types .

If there is no more than one data record type for each case, a case identifier is not needed (each data record represents a case).

Nevertheless, even "single data record" cases can be assigned a type, thus permitting filtered selection of data records by their type, or different interpretation (reading) of the data record according to its type. However, when several types of data record can make up a single case (statistical unit) you are obliged to identify each one with its case identifier and record type.

The case identifier is an alphanumeric field positioned anywhere in the record. Record types are determined by one or several rules specified in the [Study Structure](#).

Example: Case identifier in columns 1-5, and data record type in column 10 (A, B, C or D).

Important: The data file must be sorted in case identifier order, but the record type order is not important.

Data records which don't belong to any declared type are ignored. Also, if more than one record bears the same case identifier and type, only the last one encountered in the file will be used.

Comment : If several types of data record are declared, and no case identifier is declared, each data record will be considered to be a separate case, regardless of its type.

Studies with levels

There can be any number of levels, hierarchical or not, each applying to a specified set of variables for each case.

Example:

LEVEL		VARIABLES
1	Household	Household size, region, equipment
2	Personal	Sex, age, profession
3	Travel	Mode of transport, origin, destination

When using levels, you must specify their hierarchical relationship. The highest level is Level 1 ; each level (with the exception of Level 1) is dependent on a single higher level, called its Parent level.

Note: Data files for studies with levels must be **fixed position** files i.e. field-delimited files are not recognized.

In relation to the data file, each level corresponds to either:

- One or several types of data records. Each record type is defined by one or several conditions applying to data in specified column positions in the data file.
- Or, the level can be defined uniquely through its case identifier. Whenever this identifier changes, a new case is created at this level.

This case identifier is required if there is more than one type of data record at this level, or if the level has Dependent levels.

Only levels without Dependent levels may exist without a case identifier or type. In all situations, any declared data record, or all data records if no type has been declared at this level, will be considered as belonging to this level.

At a given level, if no case identifier is specified, the end of data for this case/level is assumed when the defined data record types have been exhausted. On the other hand, if a level identifier is used, it must be used to identify all types of data records at this level, and for its dependents.

The data file must be sorted on the identifier used for Level 1. However, all other data records in the same family (i.e. bearing the same identifier for Level 1) can be presented in any order.

Declaring the structure of a new study

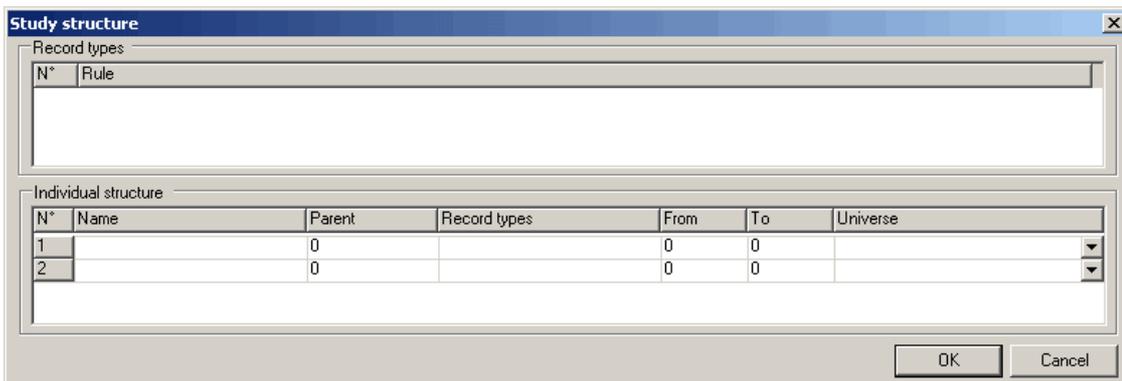
There are two basic types of study:

1. A study **without levels**, and with a **single type of data record** (ASCII file with fixed column positions for the data, or a field-delimited CSV file). There is no structure to declare; you can create variables right away.
2. A study using **multiple data records**, and/or **multiple levels**. For this type of study, you need to define the structure of the data file i.e the types of data record, and if applicable, levels and their structure.

To access to the Study Structure window, click on the « define structure » button in the Toolbar:

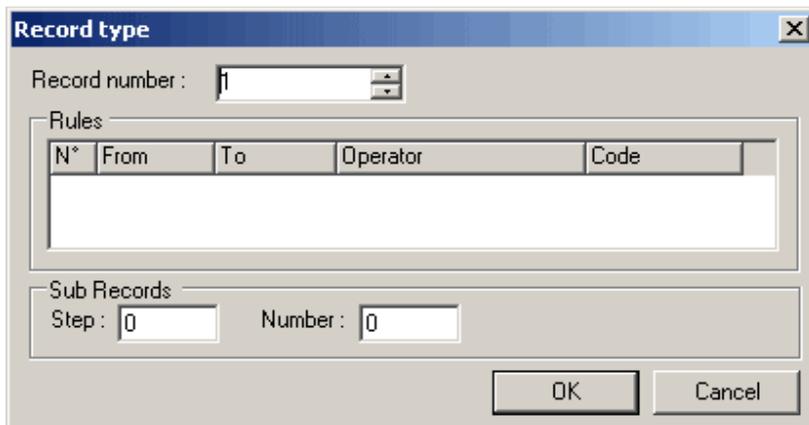


In the Study Structure window you can define the structure of the data file i.e. the types of data record and, if applicable, the level structure.



Defining data record types

To create a new record type, right-click in the Record types field or press the Insert key; the Record type definition window appears:



Each record type is defined through one or more Rules; each rule consists of a conditional test against a value taken from a specified character string in the data file. The From and To fields refer to the location of the character string. The test uses the =, != ; >, <, <=, >= operators combined with a specified argument.

To create or delete a condition, right-click in the Rules list area.

You can also define sub-records, in which the record contains several cases whose variables are all offset by the same amount. In this situation you specify:

- the **step**, assumed to be a constant, which separates two consecutive cases.
- the **number of sub-records** i.e. the number of cases (statistical units) to create.

Example: The record applies to a single person, the sub-records correspond to his/her n possible children described by the variables Sex, Age etc.

Comments:

- A record in the data file can belong to several types simultaneously though this not common.
- If the study is declared without levels, and if you define sub-records, the only cases (statistical units) taken into account will be those corresponding to the sub-records. In the above example, only the children will be counted; not the individuals (parents).
- When you use the sub-records option, if the variables do not have identical steps (column widths), for example a step of 1 for Sex and a step of 2 for age, then you need to declare two types of record : one for sub-records with a step of 1, and another for sub-record with a step of 2. The Sex and Age variables will be read from each of these sub-records respectively.

Defining levels

To create a new level, right-click in the Levels structure pane.

Record types	
N°	Rule
1	(I0-I0 > "0")
2	(I2-I2 > "0")
3	(I9-I9 > "0")

Individual structure						
N°	Name	Parent	Record types	From	To	Universe
1	Household	0	1	1	2	
2	Member	1	2	1	2	
3	Trip	2	3	1	2	

For each level you need to declare:

- The name for the level
- Its parent level (0 if it is already the highest level i.e. Level 1)
- The record type(s) applying to this level. For several record types, they can be listed separated by commas.
- If applicable, the position of the case identifier **for this level** (From and To fields)
- If applicable, the universe (sub-set) which conditionally selects for this level.

In the Universe field you can put a logical variable allowing you to filter for (select) cases at this level meeting the conditions of this universe. Only those meeting the true condition will be considered.

Cases filtered in this way will be the only ones taken into account during operations such as table computation, weighting, exporting etc.

Comment: Even if a study does not have levels, you can use the first line to define a universe which filters for specific cases.

Comments:

- If a level references a record type containing sub-records, this level must be the lowest i.e. it must not have any dependents.
- If a level references a record type containing sub-records, it may call other types of record which generate the same number of sub-records, but it may not call 'normal' records i.e. those which do not generate sub-records.
- At the lowest level (one without a dependent), it is permissible not to have a case identifier provided that it calls record types with sub-records, or a single normal type record.
- For any given level, if no record type is declared, then all of the records are assumed to belong to that level.
- All of the above applies also to studies without levels, given that implicitly they consist of a single level (Level 1) without dependents.

Importing a study

Overview - Importing a study

Imported studies do not have levels.

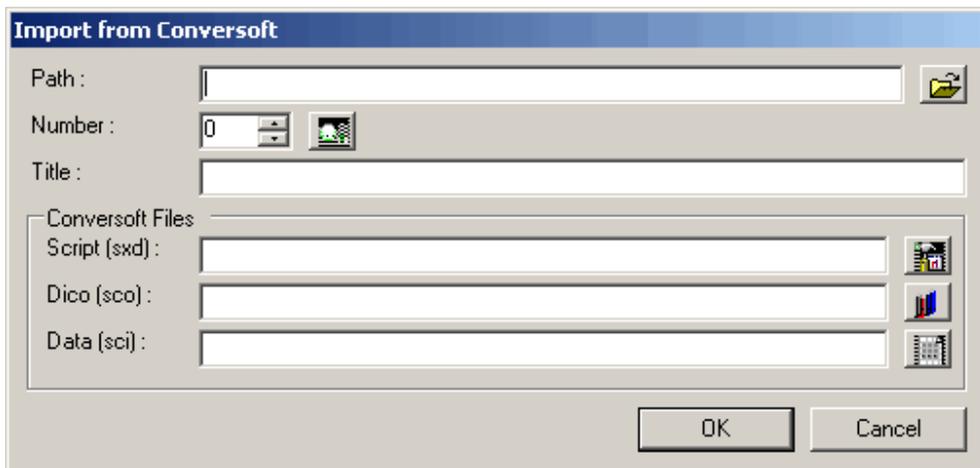
You can import from studies conforming to the data format of several software applications::

- Conversoft
- Cosi
- Quancept
- SPSS
- Triple S
- Excel

Importing from Conversoft

You must have files created by Conversoft bearing the extensions sxd, sco et sci.

1. In the **File>Import** menu, select **Conversoft**

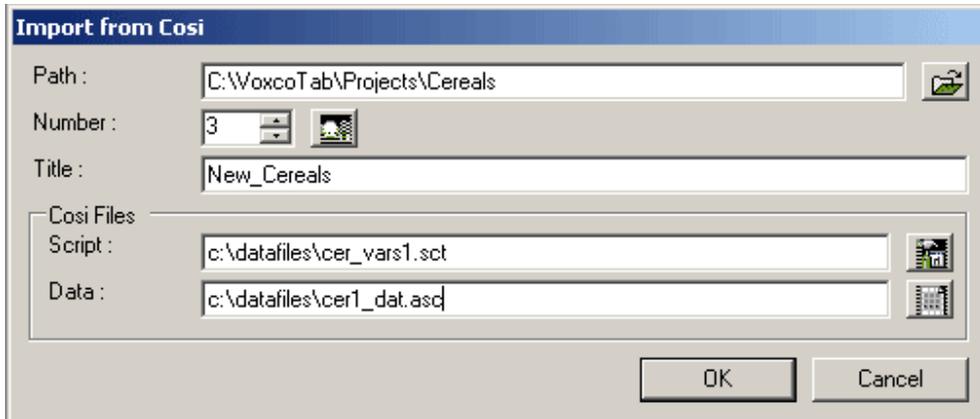


2. Indicate number of the study that you wish to create, together with its title and path.
3. Specify the Conversoft files bearing the extension sxd, sco et sci.
4. When you click on OK, the study will be created. You can modify this study by adding variables, deleting source variables etc.

Importing from COSI

You can import files created by Cosis's Export command. The files bear the sct (script) and asc (data) extensions.

1. In the **File>Import** menu, select **COSI**

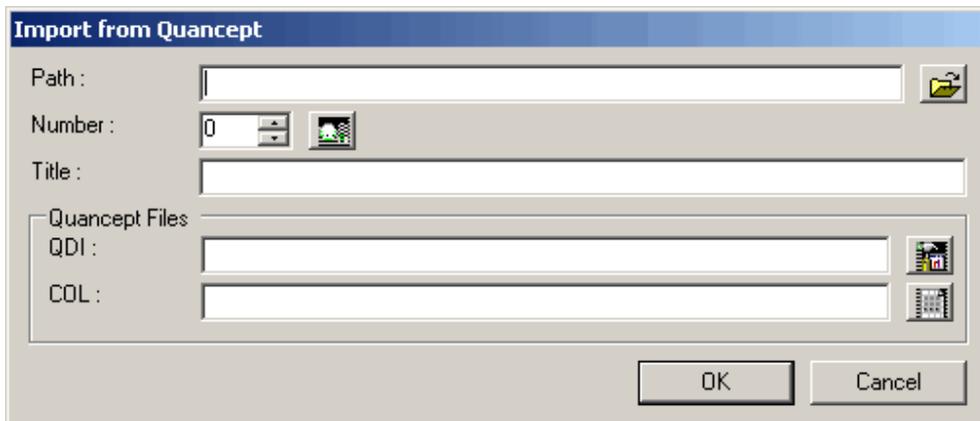


2. Indicate number of the study that you wish to create, together with its title and path.
3. Specify the COSI files bearing the extension sct and asc extensions.
4. When you click on OK, the study will be created. You can modify this study by adding variables, deleting source variables etc.

Importing from Quancept

The data file must be in ASCII mode (1 case=1 record), not in binary mode (1 case=n records).

1. In the **File>Import** menu, select **Quancept**



2. Indicate number of the study that you wish to create, together with its title and path.
3. Specify the files which were created by Quancept. When Quancept creates the files it gives them the QDI and COL extensions but it does not matter if they have been renamed.
4. When you click on OK, the study will be created. You can modify this study by adding variables, deleting source variables etc.

Processing specific to Quancept

The instructions below assume that you are familiar with the format attributes of variables (described later in this user guide).

Files produced by Quancept assign specific values for single or multiple variables:

	Value (string)	Meaning
1	0	Other values (OS)
2	-	No answer (NA)
3	&	Dont Know (DK)

A value of 0 (zero) does not cause any problem, since it is accepted by VoxcoTab both for single and multiple response variables.

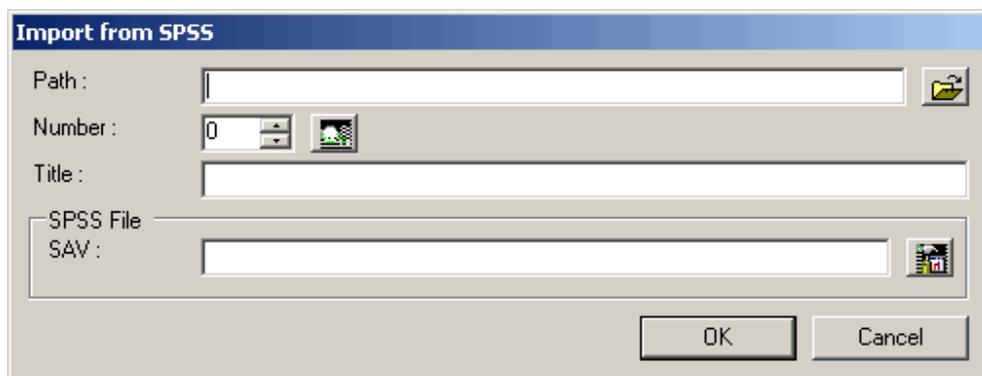
However, the two other values are transformed as follows:

- **Multiple response variable:** The special codes - and & are transformed, not only in the format assigned to the variable but also in the values read in the data file. They are converted to the new values -1 and -2 respectively.
- **Single response variable:** In order to handle these 2 new codes (which are not numbers) VoxcoTab transforms the Quancept single response variable into a multiple response variable. The special codes - and & are transformed, both in the format assigned to the variable and also in the values read in the data file. They are converted to the new values -1 and -2 respectively. However, given that this variable might be used in the calculation of means, the « value » column of the created format is initialized with the numerical values 1 to n. The special codes discussed above are not given values.

Importing from SPSS

You can import files bearing a SAV extension (produced by SPSS).

1. In the **File>Import** menu, select **SPSS**



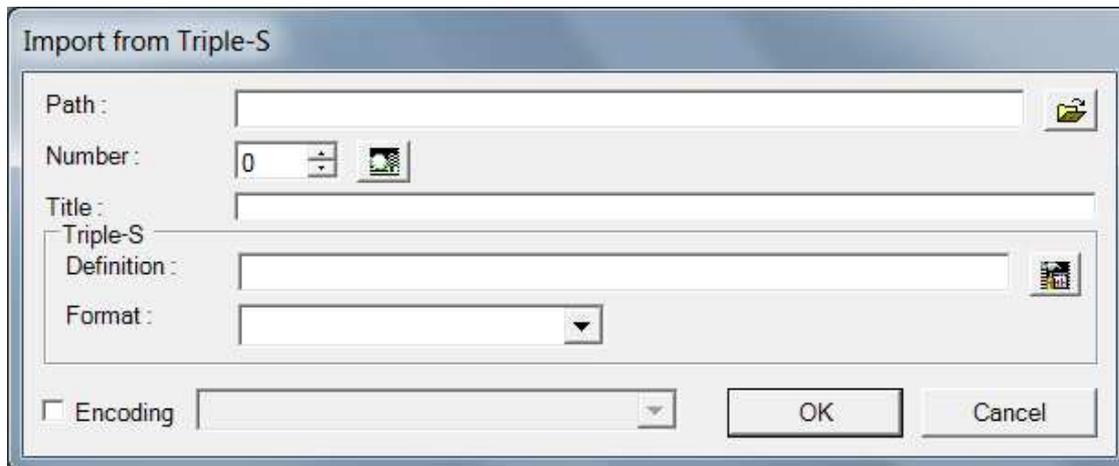
2. Indicate number of the study that you wish to create, together with its title and path.
3. Specify the SPSS files bearing the extension SAV.
4. When you click on OK, the study will be created. You can modify this study by adding variables, deleting source variables etc.

The corresponding data file will also be created under the name E_nnn.asc, where nnn is the study number you specified.

Importing from TripleS

Note: TripleS files must have been created in XML format. TXT files are also admitted (since release 2.8)

1. In the **File>Import** menu, select **TripleS**



2. Indicate number of the study that you wish to create, together with its title and path.
3. Specify the TripleS file.
4. When you click on OK, the study will be created. You can modify this study by adding variables, deleting source variables etc.

Importing Excel (xls orxlsx)

You can import Excel files (xls or xlsx), from the File>Import menu.

Variables are generated from the first line of the Excel file :

The name of the variable is Vnn, where nn is the column number

The type is 'Text'

The title is taken from the column name

The function is Input csv, with the correct column number

a file Ennn_csv is provided as data file, the first line of the Excel file being deleted, and the character « ; » being used as delimiter.

Some advice to improve processing.

The user is free to change the type of any variable, and can use the option « create format from data » to obtain the format for variables that will be used the tables.

Opening a study

When you open a study (**File>Open** menu), the variables dictionary is displayed.

You can:

- Change the title of the study (**Tools>Properties** menu)
- Modify the variables dictionary by adding, changing or deleting variables.

Warning: If you delete variables used in different VoxcoTab modules (tables, export, weighting etc.x), the module in question might fail on execution.

- Create tables (**Tables tab**).
- Request file exportation (**Utilities>Export data** menu)
- Request weighting adjustment (**Utilities>Weighting factor** menu)

File saving: All modifications made in the variables dictionary are saved when you select **File>Save** or **File>Save As** in the main menu. Hence, you can save the study in another directory and/or under another number.

Quitting: If you quit VoxcoTab without using the **File>Save** or **File>Save As** commands, all changes will be lost. As a precautionary measure, you are always presented with a Save prompt when you quit, just in case you have forgotten to do so.

Properties

The Properties window is accessed from the **Tools>Properties** menu.

It enables you to:

- Create or modify the title of the study.
- Specify the field delimiters and text qualifiers used in the CSV data files.
- Specify the name of the data file to be processed. Several data files can be entered in this list, in which case their data will be concatenated virtually for processing as a single file.
- Enable the 'Cache' option. See [Cache File](#).
- Specify the **default font** to be used when creating new labels in 'Formats'. This choice is valid at the application level, not just at the study level.

Deleting a Study

The **File>Delete** menu allows you to choose a study and delete it.

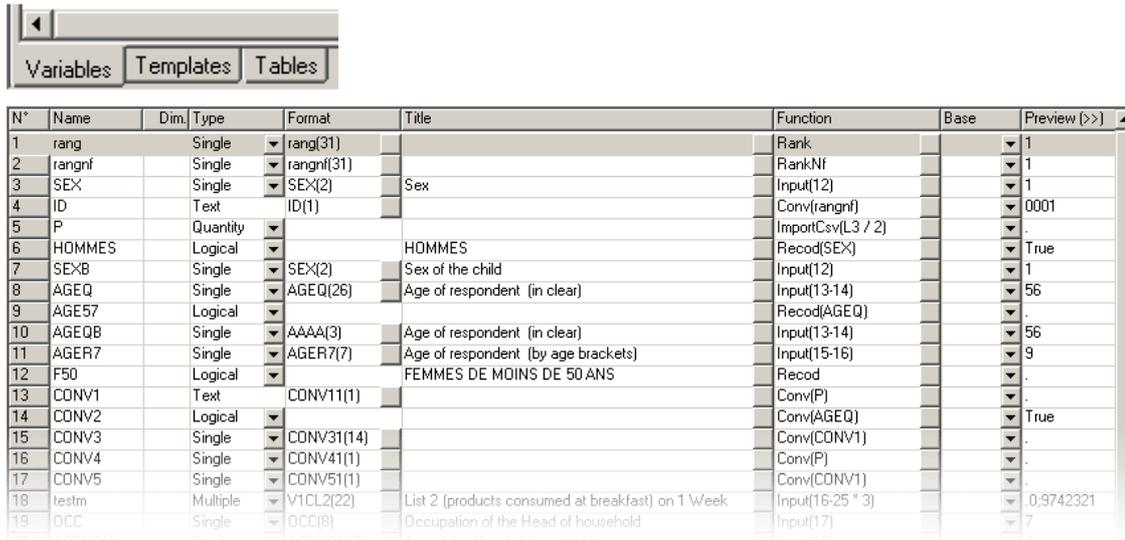
Note: You are prompted to confirm that you indeed wish to delete it

Variables

Overview - Variables

Variables contain source information for statistical tables. Each case in the source data file provides a value for each variable. Variables are created automatically when opening an existing study or importing one (they can, of course, be modified). When you create a new study, you must create its variables.

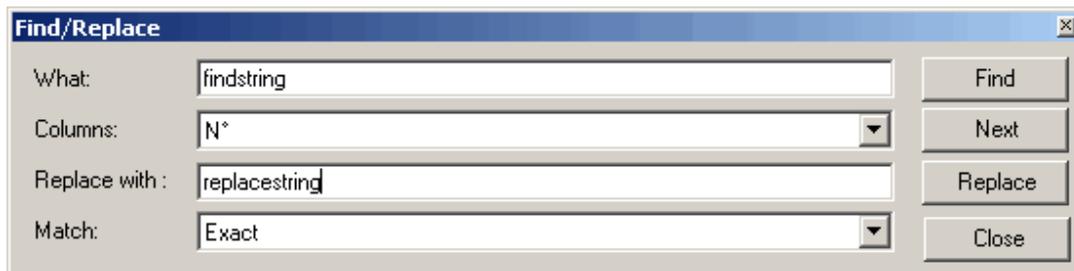
The variables window is brought into view when you click on the Variables tab at the bottom-left of the screen:



You can test for **Errors in the variables list** can be clicking on the Check button in the toolbar:



A **Find/Replace** tool, accessed through the **F3 key** or **Ctrl+F key** combination, operates on any column in the list that you specify:



Note: The tool also operates in the Formats and Functions columns. Note also that search strings are case-sensitive.

Variable Attributes

Each variable is described through a set of attributes:

Attribute	Description												
Name	Up to 16 alphanumeric characters, the _ (underbar) character is permitted. The name must start with a letter												
Level	Applies only when the study was created and declared as a study with levels												
Dimension	Applies when the variable comprises n occurrences i.e. when it represents a series of n questions which are identical apart from their titles.												
Type	<table border="1"> <thead> <tr> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Single</td> <td>The value of the variable for a case is an integer (positive or negative), zero or the conventional default value Missing.</td> </tr> <tr> <td>Multiple</td> <td>A case can have several integer values simultaneously out of the set (-2, -1, 0, 1, 2, 3, ...). By default, if none of these values is present, it has the conventional default value Missing.</td> </tr> <tr> <td>Quantity</td> <td>The value is a real number, or the conventional default value Missing</td> </tr> <tr> <td>Text</td> <td>The value is a string of n characters (n is defined in the variables Function field). The conventional default value (empty field) is Missing.</td> </tr> <tr> <td>Logical</td> <td>The value can be either TRUE or FALSE</td> </tr> </tbody> </table> <p>Note: A variable classified as Missing signifies that its data is missing or non-compliant. For example, alphabetic characters when a numerical value is expected.</p>	Type	Description	Single	The value of the variable for a case is an integer (positive or negative), zero or the conventional default value Missing.	Multiple	A case can have several integer values simultaneously out of the set (-2, -1, 0, 1, 2, 3, ...). By default, if none of these values is present, it has the conventional default value Missing.	Quantity	The value is a real number, or the conventional default value Missing	Text	The value is a string of n characters (n is defined in the variables Function field). The conventional default value (empty field) is Missing.	Logical	The value can be either TRUE or FALSE
Type	Description												
Single	The value of the variable for a case is an integer (positive or negative), zero or the conventional default value Missing.												
Multiple	A case can have several integer values simultaneously out of the set (-2, -1, 0, 1, 2, 3, ...). By default, if none of these values is present, it has the conventional default value Missing.												
Quantity	The value is a real number, or the conventional default value Missing												
Text	The value is a string of n characters (n is defined in the variables Function field). The conventional default value (empty field) is Missing.												
Logical	The value can be either TRUE or FALSE												
Format	<p>The assigned type: Single, Multiple or Text variables may have an assigned format comprising:</p> <ul style="list-style-type: none"> • A list of codes (categories) to appear in tables whenever a frequency distribution is required. • Associated labels • Optionally, values (sometimes referred to as weights) to be used in place of numerical codes when calculating means. • Other formatting options. 												
Title	The title which is to appear in tables.												
Function	The function used to assign a value to the variable (determining the source or content of the variable)												
Base	A logical variable which operates as a case filter whenever the variable is used in a table.												

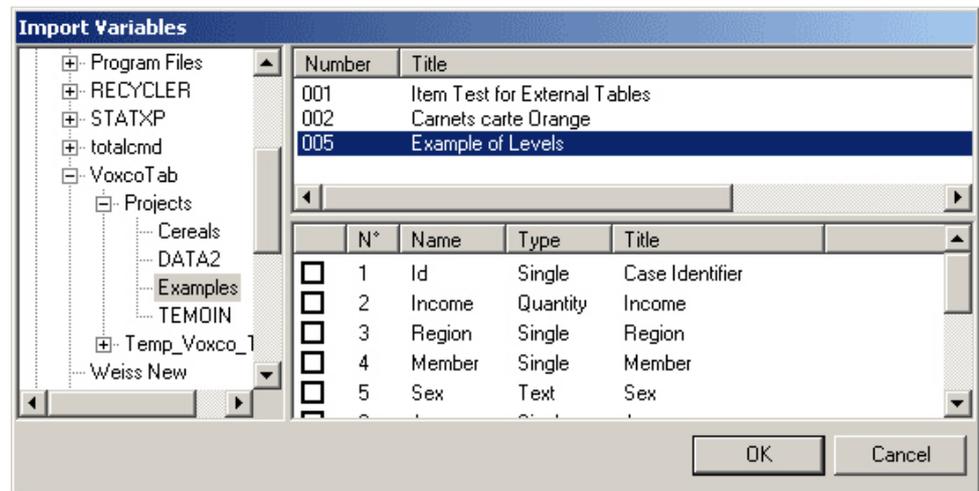
Manipulating variables

When you **right-click** on a variable's attribute, a context-sensitive menu is presented. If the complete line of a variable selected, all of the possible menu items appear:

Note: Some of these commands are also accessible from the Main menu.

Copy
Paste
Insert
Delete
Duplicate
Reset
Import
Link
Frequency table
Topic
References
Referenced by

Function	Main menu command	Comments
Copy	Edit>copy	Copies one or more variables from the same or another study. They are placed on the Clipboard.
Paste	Edit>Paste	Pastes (from the Clipboard) variables copied from the same or another study. When pasting variables from another project, make sure that they do not bear the same names as your current project. If they do, the paste operation will be blocked. Besides, it is better to paste variables into a new line at the end of the variables dictionary to minimize the risk of pasting over existing variables. In any event, if a variable to be pasted has a format which already exists in the study, the existing format will be preserved.
Insert	Edit>Insert	Inserts a new variable before the currently selected variable.
Delete	Edit>Delete	Deletes the currently selected variable. Note: The whole line of the variable must be selected, by clicking in its (left-most) variable number column.
Duplicate	Edit>Duplicate	Duplicates the currently selected variable.
Reset		Clear
Import	Tools>Import	Imports (one or more variables)



You are requested to specify the study, and to check the variables which you wish to import. When a variable to import bears the same name as an existing variable, you will be prompted choose whether to ignore the variable, rename it, or write over the existing variable.

Imported variables come with all of their attributes included their formats. If the name of a variable's format already exists in the study, you will be prompted to choose whether to ignore the format, rename it, or write over the existing format.

Link	This command is used for merging studies. See pour fusionner des études (voir § 3.9
Frequency Table	Produces a pop-up frequency table of each of the variable's categories. See Frequency Tables
Topic	<p>Enables you to classify your variables into topics (which you define). These variables will thereafter be easy to identify in the variables list through the format and colour of the font which you have assigned to the topic.</p> <p>In the Topics window, a single-click enables you to Add/Remove a topic, and to give it your preferred font attributes. When you click on the Apply button, the currently chosen topic will be assigned to the variables currently selected in the variables list.</p>
References	Provides a list of all the variables used in defining the current variable.
Referenced by	Provides a list of all the variables whose definitions use the current variable.

To reposition a variable in the variables list

Select the variable and press the **CTRL+Up/Down arrow** on your keyboard.

Note: This feature may sometimes be limited in order to maintain compatibility with the selected variable's functions and any variables which depend on it.

To find or replace names, text strings

The CTRL+F keys bring up the Find/Replace window where you can find variables by means of their name, title text string etc.

Editing variable attributes

The method for editing a variable's attribute depends on which kind of attribute is selected:

Attribute	Edit method(s)
Name, Dimension	<p>Type directly in the field. The Name attribute may be left blank.</p> <p>Dimension: When you click on the + (plus sign) to the left of a dimension, the variable is expanded to show all of its occurrences. This enables you to assign a title to each occurrence, using copy/paste functions if you wish.</p>
Type	<p>Type the capital letter of the Type i.e. S,Q,M,T,L</p> <p>Click on Edit button to the right of the field.</p> <p>Notes:</p> <ul style="list-style-type: none">• You cannot modify the Type of a variable whose Type is currently Text together with a Format.• A variable which has a Format whose Type is currently Single can only be changed to Multiple, and vice versa.
Format, Title, Function, Base	<p>Click on Edit button to the right of the field.</p> <p>A right-click in the field enables you to Copy, Paste-into and Reset (clear) the selected attributes of the currently selected variable(s).</p>
Level	<p>Type directly in the field.</p> <p>Note: The level you specify must comply with those defined in the Study Structure, or be 0 (zero). A variable set to level 0 includes (by convention) all cases, while any level greater than 0 includes only cases belonging to this level.</p>

Formats

Overview

Formats apply to Single, Multiple or Text variables. They enable you to specify a variable's discrete categorization scheme whenever you need a variable's distribution (by single codes and/or subtotals). They also allow you to assign each code (category) with a numerical value which will be used in calculating means, in place of the numerical code itself.

Formats are automatically generated when importing a study; bearing the name of the corresponding variable.

Assigning a Format

To assign a format to a variable, click on the button to the right of the Format field. You can:

- **Use an existing format** which will then be also associated with the current variable.

Note: A Text variable can only be assigned to a format that is associated with another Text variable. Similarly, a Single or Multiple variable can only be assigned to a format that is associated with another Single or Multiple variable.

- **Duplicate an existing format** under another name. This option applies when you need to modify the format, but only for the currently selected variable. The format you select is presented in the Format definition window where you can modify it.
- **Import a format** from another study (it must conform, of course, to the above restrictions).
- **Create a new format.** When you have provided a name (1 to 8 characters), the Format definition window is presented.

Creating a Format

N°	Code	Label	No Sort	NE	Value
1	1-5	Bread	<input type="checkbox"/>	<input type="checkbox"/>	
2	1-2	<i>Fresh</i>	<input type="checkbox"/>	<input type="checkbox"/>	
3	1	Fresh white bread	<input type="checkbox"/>	<input type="checkbox"/>	
4	2	Fresh speciality bread	<input type="checkbox"/>	<input type="checkbox"/>	
5	3-4	<i>Pre-wrapped</i>	<input type="checkbox"/>	<input type="checkbox"/>	
6	3	Pre-wrapped bread	<input type="checkbox"/>	<input type="checkbox"/>	
7	4	Pre-wrapped speciality bread	<input type="checkbox"/>	<input type="checkbox"/>	
8	5	<i>Bread something</i>	<input type="checkbox"/>	<input type="checkbox"/>	
9	6-7	Viennoiseries	<input type="checkbox"/>	<input type="checkbox"/>	
10	6	Fresh viennoiseries	<input type="checkbox"/>	<input type="checkbox"/>	

Each line (category) in the format will generate an item (row, column or page, according to the circumstances) in a table which uses the associated variable. The item corresponds to the variable's:

- **Code** (an integer for Single or Multiple variables, a string for Text variables)
- **Range(s) of codes** (in which case the item comprises the logical total of these codes)
- **No code**, in which case the item will appear as a 'comment' line in the table.

To each code or range of codes you assign a label (256 character max.) which will be displayed in place of the code in tables whose variables use this format.

You must fill-out at least the Code and Label columns, thus providing a list of those variable's codes to be processed and included in tables requiring frequency distributions, and the labels to be displayed in place of the codes.

Specifying codes and labels

You can do one of the following:

- **Insert. Delete or Duplicate a line**, select and right-click on a line in the table. The pop-up window also allows you Copy, Paste, and Reset.
- **Edit directly in the Code, Label and Value fields**. You can use the arrow keys to navigate around the table.
- **Copy and paste directly in a field**. For example, you can copy a list of codes or labels in an Excel spreadsheet, and paste them into the corresponding column.
- **Specify ranges and sets of codes for sums**
 - The Sum function, available with a right-click of the mouse, will generate a new row in which the specified range of codes (sub-total) reflects the lines selected at the time.
 - To specify a sum consisting of a range or set of codes, you can also type them directly, using the range operators "-" and ";". Thus you can specify any valid codes, even if they don't already exist in the format.
Example: 1-3; 5-22; 43-299
 - Another way to specify a sum consisting of a range or set of codes is to click on the button to the right of the Format field. A code list is presented in which you select the codes you wish by means of the checkboxes provided.

Formatting labels

You can specify the **Font, style, or size** of one or more labels at a time. Select the line(s) and right-click the mouse for access to the Font Configuration window.

Indentation of the selected label(s) to the right or left is also accessed by a right-click of the mouse.

Blank code. If you specify a label, and the accompanying code is blank, this enables an **empty row or column** to appear in a table using the variable.

Line exclusions

The **No Sort** checkbox will exclude its line when a table using the variable is sorted.

The **NE** checkbox will exclude the line in a table.

Note: To choose the options "No sort" and "Hide", you can select several boxes and use the Check All option available with a right-click of the mouse

Values

The **Value** field allows you to assign values (with decimals) to the corresponding codes. When using this feature, means or sums in a table using the variable will be based on these values. Certain conditions apply:

- When assigning values to codes, any codes (categories) for which no value has been specified will be excluded from means or sums. In the event that no code has been assigned a value, means and sums will be computed on the value-equivalent of the codes.
- Values assigned to codes (categories) based on a range of other codes (i.e. more than one code) are excluded from means and sums.
- For a multiple variable, its value will be the sum of the values of the selected codes. If any of the selected codes does not have an associated value, the value of the variable will be classified as Missing.

Formatting tables

You can reposition a selected line (code) in the list by means of the CTRL+Arrow Up or Arrow Down keys. This positioning will be reflected in all tables using this format.

The use of formats enable you to configure the presentation of a variable in tables in categories which can include ranges of codes without the need to create new variables.

The number of lines (categories) thus described in the format is shown in the main Variables screen (in parentheses), following the format name.

If a format is assigned to several variables, it can be modified from any one of these variables.

Automatic format generation

This option, accessible through the **Tools>Create Formats from data...** menu, automatically creates formats for all Single, Multiple or Text variables which do not already have a format. It reads the source data file, generates a line for each unique value for the variable, and copies each value into both the Code and Label field. You can then modify these labels, and create code groupings using the Sum function etc.

You can use this option to obtain a value distribution for all variables whose categorization is unknown.

You can also apply the option to a single variable. To do so, position the cursor on the variable's line; a right mouse-click allows you to select the option "Automatic formation generation for this variable". Note: The option will be ignored if the variable already possesses a format.

Examples of Formats

1. Creation of value ranges (only).

The screenshot shows the 'Formats' dialog box with the following data:

N°	Code	Label	No Sort	NE	Value
1	15-30	15 to 30 years old	<input type="checkbox"/>	<input type="checkbox"/>	
2	31-49	31 to 49 years old	<input type="checkbox"/>	<input type="checkbox"/>	
3	50-99	50 years old or more	<input type="checkbox"/>	<input type="checkbox"/>	

Only the value ranges are included. The variable is not recoded, thus preserving the true values for calculating means.

2. Editing of codes, sub-totals, and means calculated on new values.

The screenshot shows the 'Formats' dialog box with the following data:

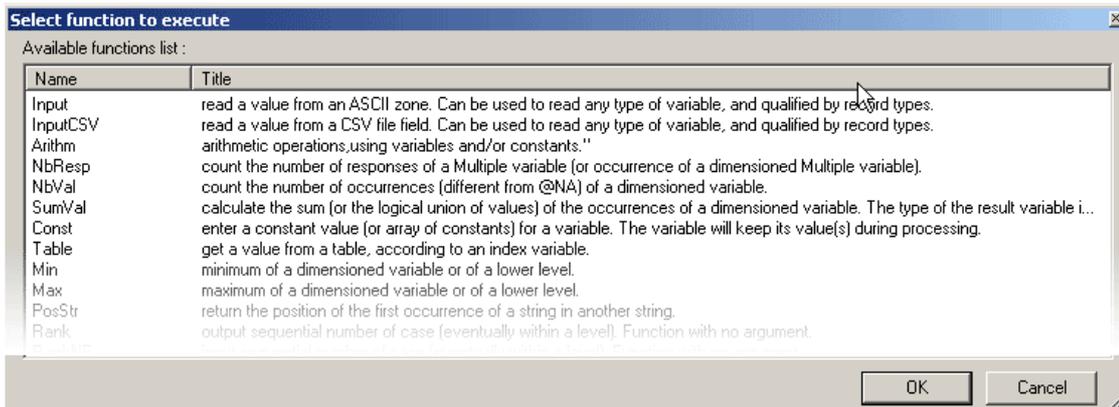
N°	Code	Label	No Sort	NE	Value
1	1-2	Agree	<input type="checkbox"/>	<input type="checkbox"/>	
2	1	Strongly agree	<input type="checkbox"/>	<input type="checkbox"/>	5.000
3	2	Slightly agree	<input type="checkbox"/>	<input type="checkbox"/>	4.000
4	3	Neither agree nor disagree	<input type="checkbox"/>	<input type="checkbox"/>	3.000
5	4-5	Disagree	<input type="checkbox"/>	<input type="checkbox"/>	
6	4	Slightly disagree	<input type="checkbox"/>	<input type="checkbox"/>	2.000
7	5	Strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	1.000

Functions and Operations

Overview - Functions

The role of functions is to provide variables with values whether by reading the source data file, reading an external table, or through any operation which re-codes other already defined variables. The last function listed in this section (CPL function) enables you to write a procedure in the CPL language through which you can assign values to the current variable, or any variable which precedes it. The CPL language is described in the section on [Procedures](#).

When you call a function, by clicking on the button to the right of the Function field, the list presented includes only those functions appropriate for the variable's attributes i.e. type, dimension, level, and whether a format has been specified.



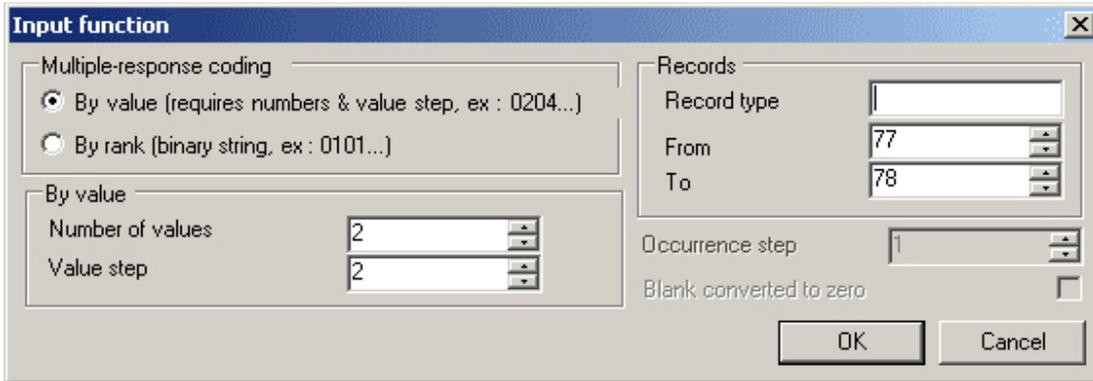
For any given variable, a function can only reference in its argument variables which precede it in the Variables List.

Conversions. Several functions required the conversion of values before they can be executed. For example, an arithmetic operation resulting in a Single type variable (hence an integer), and in which an operand is a Quantity variable (real number), requires that the Quantity variable be truncated to an integer before the function is executed.

Input function (ASCII)

A variable assigned with the **Input** function takes on its value while reading a record in a text input file. Usually each case corresponds to a single data record, and you only have to specify its position in the data record. In other, less usual, circumstances, you must specify both the **record type** in which data is located. and its **position** in this record.

If no type is specified, the data will be read from all types of data record.



The input field is specified through its starting and ending position (From, To). **Note:** The first position in a record is called 1, not 0 (zero).

For a **Multiple variable**, you specify the starting and ending position of its first value. The **Values step**, i.e. the number of positions between each value, and the **Number of values** are optional attributes for this function.

For a **Dimensioned variable**, you specify the starting and ending position of the first occurrence. The **Occurrences step** indicates the number of positions between one occurrence and another.

The values expected in the data records field depend on the variable type:

Variable type	Type of value
Single	Integer. It may carry a sign, and have leading or trailing blanks. Hence [bb][S]nnn[bbb] results in the value nnn (possibly with a sign. Anything else in the input field results in a missing value.
Quantity	A number with a decimal point. It may carry a sign, and have leading or trailing blanks. Hence, [bb][S]nnn.nn[bbb] result in the value nnn.nn (possibly with a sign). An integer without a decimal point is accepted as well, but anything else results in a missing value.

Exception for Single and Quantity variables. If the **Spaces to zero** option is checked, then an empty field results in a value of 0 (zero).

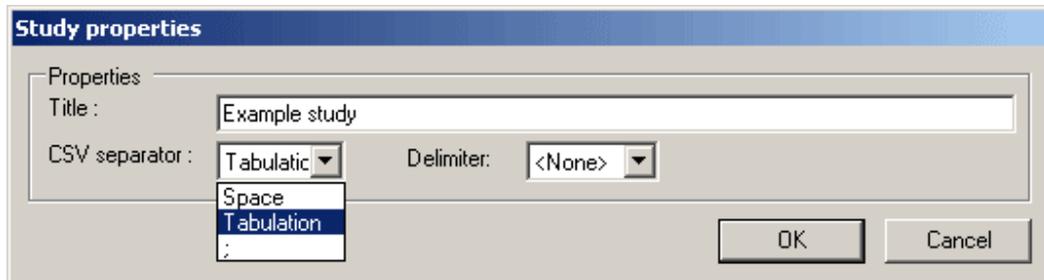
Multiple	There two input modes:	
	Mode	Description
	By Value	<p>A value is taken from each field (the number of fields being the specified Number of values). Each field must contain:</p> <ul style="list-style-type: none"> An unsigned integer, with leading or trailing blanks permitted. Hence, [b]nn[b] results in the value nn. The characters "-" or "&" which are converted, as they are read, into the values -1 and -2 respectively. See Importing from Quancept files. <p>Non-compliant values are simply ignored.</p> <p><i>Example:</i> When reading the 4 values of 6, 35, XY and blank, the result is 2 values i.e. 6 , 35.</p>

	<p>Note: The variable is only classified as missing when none of the values comply.</p>
By Rank	In this mode, the n positions in the data field correspond to the values 1 to n respectively for the variable. Hence, the variable will take on the value j, if, and only if, there is a non-blank and non-zero character in the jth position in the data field.
Text	All text is accepted without any conversion, however leading and trailing blanks are removed. The width of the data entry field will be assigned to the variable as if this were its specified character width.
Logical	A positive integer, optionally with a sign, and permitting leading or trailing blanks. Hence, [bb][S]nnn[bbb] results in a value TRUE. Anything results in the value FALSE.

Input CSV function

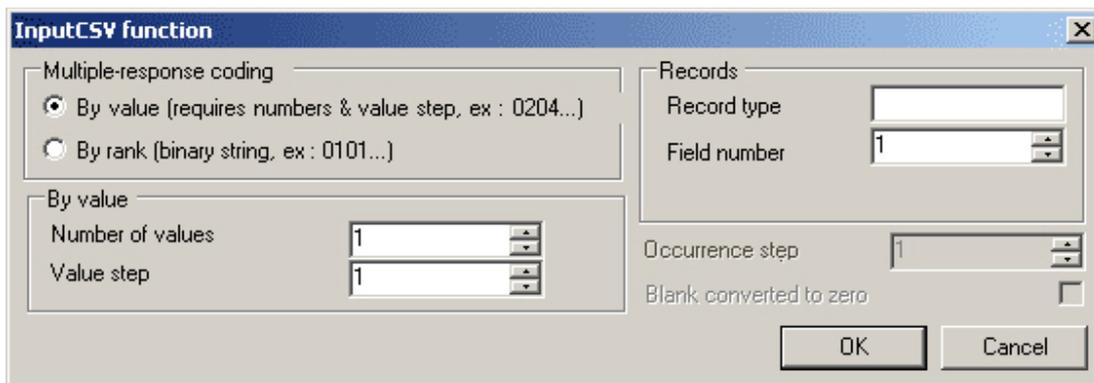
This function allows you obtain a variable's data from a CSV file, provided the study does not have levels, and has a single data record type.

The **CSV field separator** (space, tab, comma (,) or semi-colon (;) is set in the **File>Properties** menu:



The screenshot shows the 'Study properties' dialog box. The 'Title' field is set to 'Example study'. The 'CSV separator' dropdown menu is open, showing options: 'Tabulatic', 'Space', 'Tabulation', and ';'. The 'Delimiter' dropdown menu is set to '<None>'. 'OK' and 'Cancel' buttons are at the bottom right.

To read the correct data into the variable, you simply specify its **Field Number** in the file:

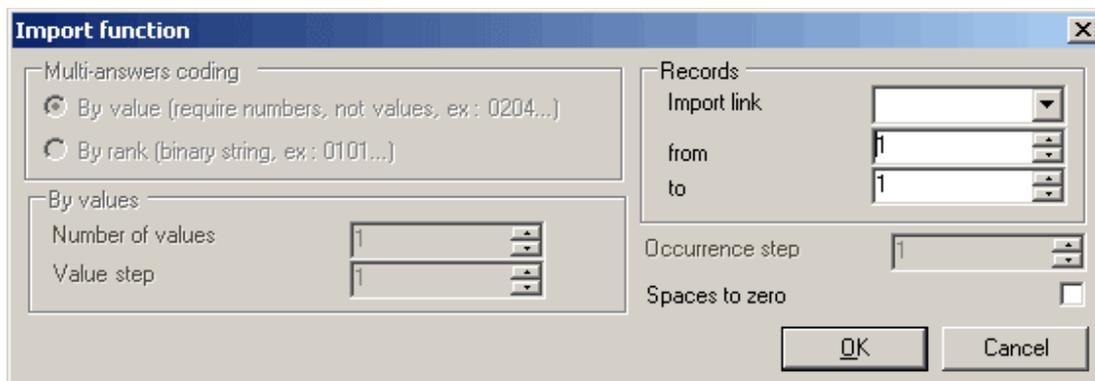


The screenshot shows the 'InputCSV function' dialog box. The 'Multiple-response coding' section has 'By value (requires numbers & value step, ex : 0204...)' selected. The 'By value' section has 'Number of values' and 'Value step' both set to 1. The 'Records' section has 'Field number' set to 1. 'Occurrence step' is set to 1. 'Blank converted to zero' is unchecked. 'OK' and 'Cancel' buttons are at the bottom right.

In all other respects, the options and behaviour are the same as for the [Input Function](#) (ASCII) i.e. for Multiple and Dimensioned variables, and the treatment of non-compliant data values.

Import function

A variable using the Import function obtains its value, not from the principal data file, but from an external table. See [External Tables](#).



The Import Function window is similar to the Input Function window, except that the Record Type field is replaced by a Link in Import File field in which you choose a link from a list of one or more links already defined in the External Tables module.

Once the appropriate record in the external table has been found i.e. whose index key(s) match those for the case, the data is read in exactly the same way as the Input function.

Comment: Loading of data from the external table is performed dynamically whenever VoxcoTab needs to compute a variable using the import function, hence up-to-date index key variables are always used.

Warning: If you change the value of these index keys, (using procedures, for example) you might be reading data from two different records in the External table.

Dimensioned variables - special circumstance

For a dimensioned variable, if the index keys in the external table are not dimensioned, the data from the selected record in the table is still read into the dimensioned variable, and the Step is used to calculate the position of each occurrence (as in the Input function).

If at least one of the index keys in the external table is dimensioned, the data is read differently. Each occurrence will be read from a different record (defined by the Occurrence for the key). Hence, only the Starting position is significant, and any Step that might be specified is ignored.

Import CSV function

This function enables you to obtain variable data from an external table in CSV format. The field separator character can be a space character, tab character, comma (,) or semi-colon (;). You determine the field separator in the File>Properties menu.

The Import CSV definition window is identical to the Import definition window, with the exception that you specify a Field number (rather than the position of the data) in the external file.

In all other respects, the function is identical to the [Import Function](#).

Import CSV function

Multi-answers coding

By value (require numbers, not values, ex: 0204...)

By rank (binary string, ex: 0101...)

By values

Number of values: 1

Value step: 1

Records

Import CSV link: [Dropdown]

Field number: 1

Occurrence step: 1

Spaces to zero:

OK Cancel

Arithmetical Operations

Operands can be:

- Non-dimensioned variables, or occurrences of dimensioned variables
- Dimensioned variables
- Constants

You may use up to 20 operands.

Both the resulting variable and variables used as operands must be numeric i.e. Single or Quantity type.

If the resulting variable is of Single type (S), then the operand values are truncated to integers before calculation. Hence, the calculation is performed in integers, and the result will be an integer.

If the resulting variable is of Quantity type (Q), calculation is floating point. If the resulting variable is dimensioned, then all of the operand variables must be similarly dimensioned or be undimensioned. Calculation is performed for each occurrence.

Operators comprise: * (multiplication), / (division), + (addition), - (subtraction) and ^ (exponentiation).

Priority of execution is ^, followed by * and /, finishing with + and -.

When operators have the same priority, the order of execution is from left to right.

A MISSING result will occur in three situations:

- Division by zero
- Whenever any of the operands has a MISSING value
- The conditional Universe applying to the function has not been verified.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S or Q	S or Q	L
	Level	i	i or <i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned
		n	undimensioned or dimension n	undimensioned or dimension n

If the base is dimensioned, the only occurrences of the resulting variable that will be considered are those whose corresponding occurrence in the base is TRUE. The other occurrences will be classified as MISSING.

Logical Operations

A Multiple variable can be built from other variables (Multiple or Single type variables) using a logical AND, OR, XOR (Exclusive) or EXCEPT operation. Integer constants may be used.

The number of operands may not exceed 20.

Example:

Variable B1 has values 1,2,3,4

Variable B2 has values 3,4,5,6

Operation	Result
B1 OR B2	1,2,3,4,5,6
B1 XOR B2	1,2,5,6
B1 AND B2	3,4
B1 EXCEPT B2	1,2

Rules and behaviour:

- Missing Values

Situation	Result
X AND MISSING	MISSING
X OR MISSING	X
X XOR MISSING	X
X EXCEPT MISSING	X

- There is no priority between operators: operations are performed from left to right:
Example: B1 OR B2 AND B3 is interpreted as (B1 OR B2) AND B3
- The constant MISSING is represented by @NA
- Operand variables must have the same dimension as the result variable, or be non-dimensioned variables, or refer to a specific occurrence.
- The values of Single type variables are converted to multiple type before the operation is executed (0 to n produces 0 to n, other values produce Missing).

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	M	M or S	L
	Level	i	i or <i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned
		n	undimensioned or dimension n	undimensioned or dimension n

- If the base is dimensioned, the only occurrences considered in the resulting variable are those whose corresponding occurrence in the base is TRUE. Conversely, the occurrence becomes MISSING.

Recoding function

This function enables you to define a category of a formatted variable in terms of one or more categories or value ranges of another (source) variable - this is sometimes known as recoding.

The type of variable to which the function is applied i.e. the resulting variable, can be Single, Multiple or Logical; and it must have been formatted to have at least one category.

Note: The conditions concerning [Logical variables](#) are somewhat special, so a special paragraph is devoted to them below.

The Recode function definition screen provides a button for selecting the source variable (applying to the category currently selected in the list), and, if applicable, a checkbox for constraining the function to only one (single) source variable applying to all categories.

Note: Even if the resulting variable's format contains multiple codes (such as sub-totals), they are not presented for recoding.

Code	Label	Source Variable	Grouping
1	Less than 18	AGE "Please tell me your..."	-18
2	19-24	AGE "Please tell me your..."	19 - 24
3	25-45	AGE "Please tell me your..."	25 - 45
4	46-65	AGE "Please tell me your..."	46 - 65
5	Greater than 65	AGE "Please tell me your..."	65 -
6	Male	GEND "Gender"	1
7	Female	GEND "Gender"	2

Specifying source variables

To specify a single source variable for all categories of the resulting variable

1. Make sure that the Single Source option is checked
2. In the first category in the list, select the Variable field.
3. Click on the Select button.
4. In the Variable list that pops up, select the source variable.
This variable will become the source variable for all the categories in the list.

To specify a different source variable for each category of the resulting variable

1. Uncheck the Single Source option.
2. Select the Variable field for one or more categories in the list
3. Click on the Select button.
4. In the Variable list that pops up, select the source variable.
5. Repeat from step 2 for one or more categories until all of them have source variables defined.

Assigning groups of categories or value ranges

The Grouping column is where you set groups of categories or value ranges for each category of the resulting variable. You can type in your specification directly, using the syntax described below:

Source Variable type	Argument	Syntax (examples)
Quantity	Real number	<ul style="list-style-type: none"> • 0-44 signifies 0 to 44, • 18-24 ; 35-44 signifies from 18 to 24 or from 35 to 44,
Single or Multiple	Positive integer or zero	

Text	Character string	<ul style="list-style-type: none"> • 1-3 ; @NA signifies 1 to 3 or MISSING, • 45- signifies 45 or more • -44 signifies 44 or less • A-D signifies from the uppercase ASCII character A to D , in ASCII code order.
Logical	Permitted values are 1 or 0 (representing TRUE or FALSE)	
All	@NA (missing) or - Indefinite (or less or more)	

Using the By Values (F5) and By List buttons (F6)

Instead of typing your grouping or value range specification for a category, you can make use of the **By Values** and **By List** wizards to help you generate it.

To use the By Values wizard

1. Select a category in the Grouping column.
2. Click on the F5 button or press the F5 key. The Value Ranges screen will appear:



3. Fill out the From and To values of the range(s) that are to apply. If you need to specify "or less" or "or more" in the From or To column respectively, you can use the F7 key or click on the F7=Indefinite button. The F7=Indefinite button.

To specify "Missing", use the F8=@NA button or press the F8 key.

4. Repeat from Step 1 for each category for which you need to provide ranges of values.

To use the By List wizard

You can use this wizard on categories whose specified source variable has itself been formatted to have categories.

1. Select a category in the Grouping column.

- Click on the F6=by list button or press the F6 key. The source variable's Recode List screen will appear:



Note: This list only displays the source variable's single code categories for selection. Category code groupings (for nets or subtotals) are excluded.

- Select the codes that you wish to be grouped into the resulting variable's category.
- Repeat from Step 1 for each category for which you need to provide code groupings.

Note: You can use mouse right-click for copy and paste functions

Variable types

Variable Type	Description
Text Source variable	When specifying code strings, quotes should not be used; the quotes visible on subsequent display are automatically inserted by VoxcoTab to highlight the fact that a text variable is involved. Character order conforms to ASCII: 0 to 9, followed by A to Z, followed by a to z.
Logical Source variable	Permitted values are 1 and 2 (TRUE and FALSE)
Dimensioned Resulting variable	If the source variable is also dimensioned (same dimension), recoding is performed occurrence by occurrence. If the source variable is not dimensioned, or if a specified occurrence is concerned, recoding is performed according the values of the non-dimensioned variable for each occurrence of the resulting variable. This rule applies to each category of the resulting variable.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	M, S or L	M,S,Q,T or L	L
	Level	i	i or <i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned
		n	undimensioned or dimension n	undimensioned or dimension n

Execution

For a Single type resulting variable, the first category encountered that meets its grouping specification will be the one selected.

For a Multiple type resulting variable, all categories meeting their grouping specification will be considered.

Warning: When a grouping specification applies to a Text type variable, string matching is limited to the number of characters of the shortest string.

Logical resulting variables

A logical variable cannot have a format i.e. categories; its state can be either TRUE or FALSE. Nevertheless, you can have several grouping specifications, each one resulting in TRUE.

When you launch the Recode function, a single line is presented bearing the label TRUE. The resulting variable will be TRUE if your source variable matches your grouping specification. You can add more lines, each with its own grouping specification. During execution, your resulting variable will be TRUE only if each and every grouping specification is satisfied.

Example:

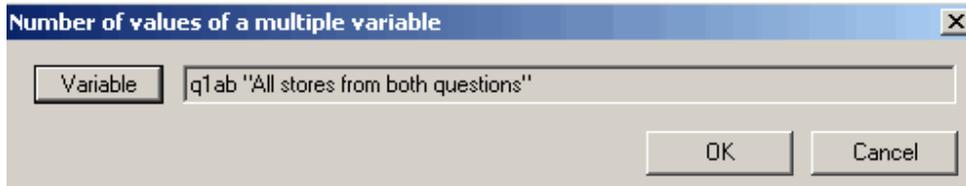
Code	Label	Variable	Recodification
1	true	AGE "Is your age withing..."	18 - 25
1	true	SEX "Sex of Respondent..."	2

The result will be TRUE for women aged 18-25.

NBRESP (no. of responses) function

Counts the number of codes selected in a multiple variable, regardless of its number of formatted categories.

The resulting variable must be Single type, and the argument variable Multiple type.



If the base is dimensioned, only those occurrences in the argument variable whose corresponding occurrences in the base are TRUE will be counted.

The result is a number from 0 to x if the base is true, and Missing if the base is false.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S	M	L
	Level	i	i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned
		n	undimensioned or dimension n	undimensioned or dimension n

NBVAL function - Number of non-missing values for a dimensioned variable

Returns the number of occurrences of a dimensioned variable which are not MISSING.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S	M, S, Q, T, L	L
	Level	i	i	i or <i
	Dimension	undimensioned	Dimension=n	undimensioned or dimension=n

Execution

The result is the a number from 0 to x if the base is TRUE, and MISSING if the base is FALSE.

If the base is dimensioned, only those occurrences in the argument variable whose corresponding occurrences in the base are TRUE will be counted. The result is MISSING if all of the occurrences in the base are FALSE.

SUMVAL (sum of values) function for a dimensioned variable

This function behaves differently depending on whether the type of the resulting variable is Single (or Quantity) or Multiple.

- For a **Single** or **Quantity** resulting variable, the sum of the various occurrences of the source variable is computed.
- For a **Multiple** resulting variable, the union of the various occurrences of the source variable is taken.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	M	M or S	L
	Type	S or Q	S or Q	L
	Level	i	i	i or <i
	Dimension	undimensioned	Dimension=n	undimensioned or dimension=n

Execution :

- As imposed by the type of resulting variable, the necessary conversions are made before performing the sum (Single to Multiple, Quantity to Single etc.)
- If the resulting variable type is Single or Quantity, and if one of the occurrences of the argument variable is MISSING, the resulting variable will also be MISSING.
- The result will be a number between 0 and x if the base is TRUE, and MISSING if the base is FALSE.
- If the base is dimensioned, only those occurrences in the argument variable whose corresponding occurrences in the base are TRUE will be counted. The result will be MISSING if all of the occurrences in the base are FALSE.

Dicho1 function - Distribute a Multiple variable to a dimensioned Single variable

This function will also accept a Single type variable as an argument.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S	M,S	L
	Level	i	i	i or <i
	Dimension	dimension n	undimensioned	undimensioned or dimension n

Execution

- The recognized codes of the argument variable are 1 thru n, where n is the dimension of the resulting variable (codes 0,-1 and -2 are ignored).
- When the argument variable contains the code j, the jth occurrence of the resulting variable is set to 1 (if not, the jth occurrence will be set to MISSING).
- If the base is dimensioned, only those occurrences in the resulting variable whose corresponding occurrences in the base are TRUE will be filled.

DICHO2 function - distribute a dimensioned variable to a Multiple variable

Values of occurrences (lying within a specified range) generate corresponding codes of the Multiple variable.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	M	S,M	L
	Level	i	i	i or <i
	Dimension	undimensioned	dimension n	undimensioned or dimension n

Execution

- When the value of occurrence j of the argument variable is within the specified range, the resulting variable is assigned code j.
- If the base is dimensioned, only those occurrences in the argument variable whose corresponding occurrences in the base are TRUE will be taken into consideration.

NBLEV function

Note: This function only applies to studies with Levels.

The function returns the number of dependents in the case being processed, whose argument variable is not MISSING.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S	M, S, Q, T, L	L
	Level	i	j > i and j depending on its parent level i	j or <j
	Dimension	undimensioned	undimensioned	undimensioned
		dimension n	dimension n	undimensioned or dimension n

Execution

- The notion of dependent derives from the study structure (Level Definition screen)
- The result is a number between 0 and x if the base is TRUE, and MISSING if the base is FALSE.
- If the base is dimensioned, only those occurrences in the argument variable whose corresponding occurrences in the base are TRUE will be taken into consideration.

SUMLEV function

This "Lower Level" sum function applies only to studies with Levels.

For the argument variable, returns the sum of the values of all dependents for the case being processed.

For the result variable, the behaviour depends on whether it is a Single (or Quantity) or Multiple type variable:

- Single (Quantity) type variable: Arithmetical sum
- Multiple type variable: The Logical sum (union) of the argument variable (which is the dependent level) is written to the resulting variable (its parent).

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	M	M,S	L
		S	S,Q	L
		Q	S,Q	L
	Level	i	j > i and j dependent i	j or <j
	Dimension	undimensioned	undimensioned	undimensioned
		dimension n	dimension n	undimensioned or dimension n

Execution

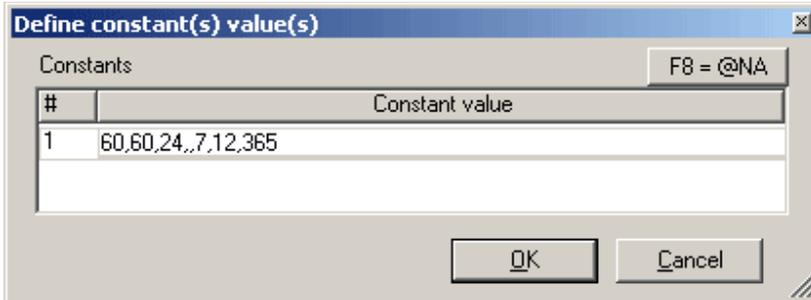
- If the resulting variable is arithmetic (Type S or Q), and if the argument variable has any MISSING value, the result will be MISSING.
- The result is a number between 0 and x if the base is TRUE, and MISSING if the base is FALSE.
- If the base is dimensioned, only those occurrences in the argument variable whose corresponding occurrences in the base are TRUE will be taken into consideration.

CONST function

This function enables you to create a table of constants within a variable (either dimensioned or undimensioned). The variable should not have a Base (even if it does, it will be ignored).

The type of the resulting variable can be Multiple, Single, Quantity or Text.

For a Multiple variable, the different values must be separated by commas:

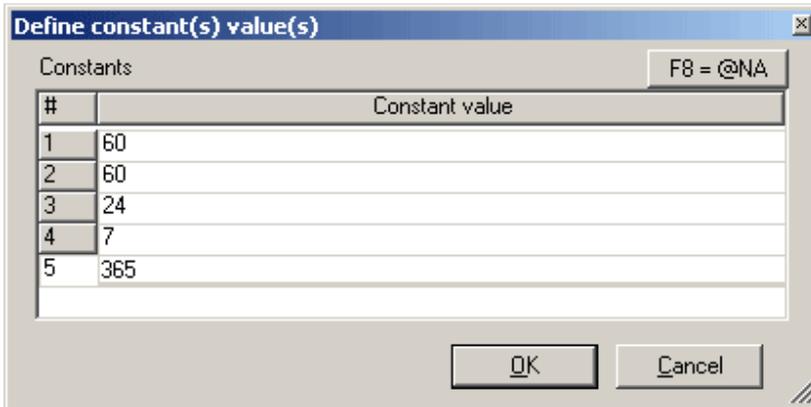


Define constant(s) value(s)

Constants F8 = @NA

#	Constant value
1	60,60,24,,7,12,365

For a Dimensioned variable, the different values are entered on each row:



Define constant(s) value(s)

Constants F8 = @NA

#	Constant value
1	60
2	60
3	24
4	7
5	365

TABLE function

This function enables you to pick a value from a table by means of an index variable.

The resulting variable takes its value from a dimensioned variable (first argument) selected by an index variable (second argument).

RESULT= TABLE [Index]

Conditions

		Resulting Variable	Argument Variable	Index Variable	Base
Variable Attribute	Type	M	M,S	S	L
		S	S,Q	S	L
		Q	S,Q	S	L
		T	T	S	L
	Level	i	i or <i	i or <i	i or <i
	Dimension	undimensioned	dimensioned	undimensioned	undimensioned
		dimension n	dimensioned	dimension n	undimensioned or dimension n

Execution

- Any necessary conversions are made automatically.
- If the resulting variable is dimensioned, it need not have the same dimensions as the table variable: RES [j] = TABLE [INDICE (j)]
- If the base is dimensioned, only those occurrences in the resulting variable whose corresponding occurrences in the base are TRUE will be taken into consideration.
- The final result is MISSING if the base is FALSE.

CONCAT function

This function can assemble a dimensioned variable by grouping together several other variables (dimensioned or not). Several types of source and resulting variable are supported.

The resulting variable can be assembled from several dimensioned variables. In all cases, it must be dimensioned to equal the sum of the dimensions of the source variables; an undimensioned variable having a dimension of 1.

With regard to variable types:

- A Multiple type variable can be assembled from both Multiple and Single variables.
- A Single (or Quantity) type variable can be assembled from Single or Quantity variables.
- A Text variable can be assembled from Text variables.

Note: To select occurrences of a dimensioned variable, select the variable and press SPACEBAR. This will expand the variable to display all of its occurrences for selection.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	M	M,S	L
		S	S,Q	L
		Q	S,Q	L
		T	T	L
	Level	i	i or <i	i or <i
	dimension n	undimensioned or dimensioned (sum of dimensions=n)	undimensioned or dimension n	

If you wish, you can title the occurrences in the resulting variable using the corresponding titles of the source variables. To do this, use the Update occurrence titles checkbox.

Execution

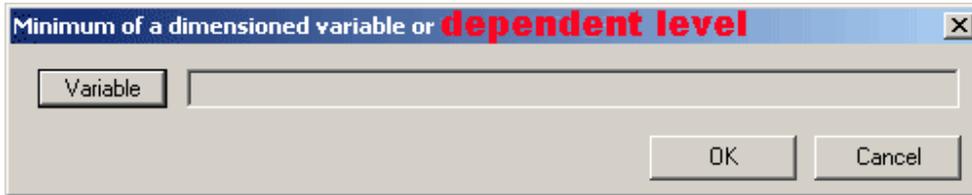
- Type conversions are made as required.
- If the base is dimensioned, only those occurrences in the resulting variable whose corresponding occurrences in the base are TRUE will be counted (all others will MISSING).
- The final result is missing if the base is FALSE.

MIN and MAX functions

Minimum and Maximum in a set of variables

These functions apply to either:

- **A dimensioned variable.** To return the Minimum or Maximum value among its occurrences, or
- **Levels.** To return a variable's Minimum or Maximum within a dependent level.



When you click on the Variable button, the displayed list of variable offers for selection only those which comply with the Conditions table(s) below.

Conditions

1. Maximum or minimum within occurrences of a dimensioned variable. Applies if the level of the resulting variable is the same as the argument level, or if the study does not have levels.

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S	S, Q	L
	Level	i	i	i or <i
	Dimension	undimensioned	Dimension=n	undimensioned or dimension=n

2. Maximum or minimum between levels. Applies if the study has levels, and if level j of the argument variable is greater than the level of the resulting variable.

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S	S, Q	L
	Level	i	j > i and j is a dependent of i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned
		Dimension=n	Dimension=n	undimensioned or dimension=n

Execution (all conditions)

- Type conversions are made as required.
- By convention, a MISSING value is considered to be the smallest value.
- If the base is dimensioned, only those occurrences in the argument variable whose corresponding occurrences in the base are TRUE will be considered.
- The final result is MISSING if the base is FALSE.

RANK and RANKNF functions - Case ranking

Rank refers to the sequence number of a case being processed. The Rank function increments the case sequence number only for cases belonging to a specified universe i.e. those that are TRUE when associated with a logical variable in the universe of the study structure. The RANKNF function, on the other hand, increments the case sequence number for all cases whether or not they belong to the specified universe.

For a study with Levels, the rank of a case belonging to Level $j > 1$ is its rank within its peers.

There is no argument, or supplementary input screen, with this function.

Conditions

		Resulting Variable	Base
Variable Attribute	Type	S	L
	Level	i	i or <i
	Dimension	undimensioned	undimensioned

Execution

With Levels:

The variable using this function will have a value corresponding to the rank position of the case within the group of cases being processed (sequence number possibly limited to the filtered cases)

Without Levels:

If the resulting variable is Level 1, its value will be rank of the case at level 1 with respect to the whole data file.

If the resulting variable is in a dependent level, cases with the same parent will be ranked starting from 1.

Warning: A variable defined using the Rank function must not be used as an argument in another variable's function.

GETLEV function - Get Level

This function returns a value equivalent to the level being processed. Usually, it is used with a variable whose level is 0.

There is no argument, or supplementary input screen, with this function.

Conditions

		Resulting Variable	Base
Variable Attribute	Type	S	L
	Level	i	i or <i
	Dimension	undimensioned	undimensioned

CONV functions

These functions allow you to convert between various value types.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S, T, L	M, S, Q, T	L
	Level	i	i	i or <i
	Dimension	undimensioned dimension n	undimensioned undimensioned or dimension n	undimensioned undimensioned or dimension n

Text-type resulting variable

As a general rule, an argument variable which is MISSING produces MISSING in the resulting variable.

Single to Text

Values can be right-justified packed with either leading blanks or zeros. They can also be left justified.

Warning: If the field width is insufficient to represent a value, the result will be MISSING.

Quantity to Text

As, in Single type to Text type, values can be right-justified packed with either leading blanks or zeros. They can also be left justified. However, the number of decimals must also be specified.

Multiple to Text

Convert a type to another. Multiple to Text (by value), ex.: 2,4,5 => "020405"

Variable: q1ab "All stores from both questions"

Format: By value
 By rank

Length of values of the new variable: 10

OK Cancel

Convert a type to another. Multiple to Text (by rank), ex.: 2,4,5 => "01011"

Variable: q1ab "All stores from both questions"

Format: By value
 By rank

Length of new variable: 10

OK Cancel

Two format options are available:

- **By values** (fixed format): You specify the string length to be used for each value.
- **By rank** (Each category of the source variable is coded 0 or 1): You specify the total length of the resulting variable i.e. the number categories to be taken.

Note: The 1st position in the output string corresponds to the value of 0 of the argument variable. The 2nd position codes the 1st category of argument variable etc. In other words, the character 1 in position j signifies that category j-1 in the argument variable is present.

Example: Take a source variable (Multiple) with codes 2, 4 and 8 selected

The resulting strings would be:

- By values (with string length of 2 for each value specified): " 2 4 8"
- By rank (with total string length of 9 specified): "001010001"

Note: If the string length were 7, then the last "01" representing the codes 7 and 8 would be truncated.

Logical-type resulting variable

There are no options. Whatever be the type of argument variable, the result is FALSE if the argument variable is MISSING, and TRUE otherwise.

Single-type resulting variable

As a general rule, if the argument variable is MISSING then the resulting variable will be MISSING.

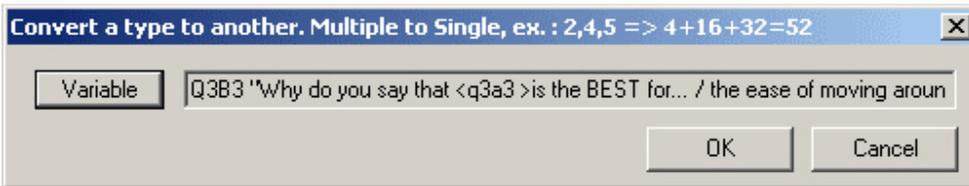
Multiple to Single

There are no options. The result is a number obtained by taking the sum of all the values 2^{**j} , j being one of the argument variable's values, that is to say:

$$j \quad 2^{**j}$$

- 0 1
- 1 2
- 2 4
- 3 8
- 4 16
- 5 32

etc.

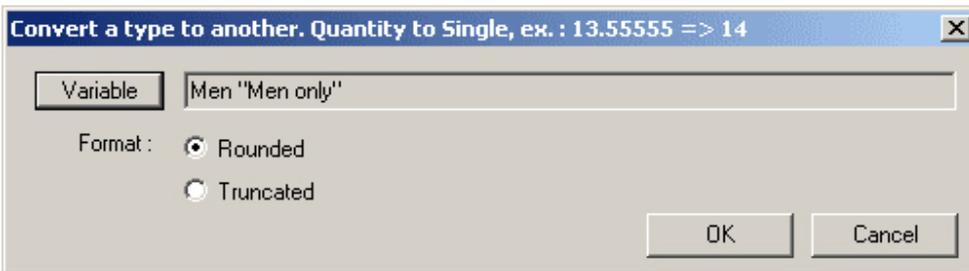


If the argument variable has values (1,2,5), the result will be 2 + 4 +32=38.

The aim of this conversion formula is to uniquely represent all possible combinations of the argument variable's values.

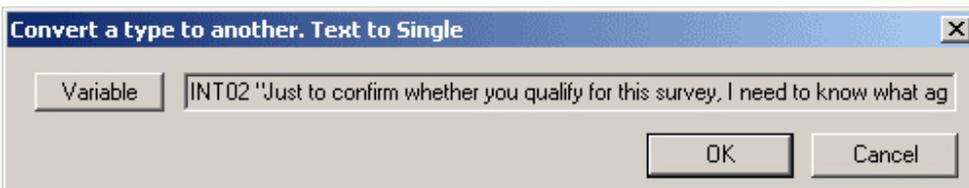
Quantity to Single

You are offered the choice of rounding or truncating the argument variable's value.



Text to Single

There are no options. The argument variable must represent an integer; it may be signed and have leading or trailing blanks. [bb][S]nnn[bbb] results in the value nnn together with its sign, otherwise the result is MISSING.



TIME and DATE functions

In all of these functions, date and time is expressed in a Text type variable whose form is:

- YYYYMMDD : Year, Month, Day (8 characters), or
- YYYYMMDDHHMMSS : Year, Month, Day, Hours, Minutes, Seconds (14 characters)

(Depending on the function, this Text type variable is involved either as an argument or result).

CurDate and CurTime functions

For these functions there is no input screen i.e. there is no argument to supply.

The resulting variable must be an undimensioned Text type variable.

The result is:

- Current Date: A string of 8 characters in the form YYYYMMDD.
- Current Time: A string of 6 characters in the form HHMMSS.

Year, Month, Quarter, DayWeek, DayYear and Week functions

The resulting variable must be an undimensioned Single type variable. The argument variable must be an undimensioned Text type variable in the form YYYYMMDD.

As their title suggests, these functions return respectively (from the argument variable): Year, Month, Quarter, Day of the Week, Day of the Year, and Week of the Year.

For the Month and DayWeek functions, a format is automatically created for the resulting variable comprising the labels January thru December, and the labels Monday thru Sunday.

Note: The Week function returns a number comprising the Year and the Week number in the format YYYYWW.
The calculation is $\text{Year} * 100 + \text{Week No.}$

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S	T	L
	Level	i	i or <i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned

Difference between dates

Two date/times in the form of undimensioned text variables or constants are compared. You can choose the units in which the difference is computed.

Conditions

The resulting variable must be Single type and undimensioned

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S	T	L
	Level	i	i or <i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned

AddTime function : Add time duration to a date

The function takes a start date and time (constant or undimensioned variable), a time duration to be added (constant or Single type variable, negative or positive), and the unit in which the time duration is expressed.

The result is a date (Text variable, width 8 or 14 depending on the start date).

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	T	T, S	L
	Level	i	i or <i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned

CHAIN function

This function will concatenate (join together) several character strings.

Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	T	T	L
	Level	i	i or <i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned
		undimensioned	undimensioned or dimension n	undimensioned

Execution

- The length of the resulting variable is the sum of the lengths of the source (argument) strings.
- Any occurrences of the source variable(s) will be concatenated sequentially.
- If the base is FALSE, the final result will be MISSING.

SUBSTR function

Extracts a sub-string from a Text type variable.

The arguments are the length and starting position of the sub-string in the source variable. They can be specified either as variables (Single type) or constants.

Conditions

		Resulting Variable	Source Variable	Variable indicating position and length	Base
Variable Attribute	Type	T	T	S	L
	Level	i	i or <i	i or <i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned	undimensioned
		dimension n	dimension n	undimensioned	undimensioned or dimension n

Execution

- The indicated number of characters starting at the specified position in the source variable are copied into the resulting variable.
- The length of the sub-string to be extracted is limited to the length of the source string.

POSTR function

Returns the position of a specified sub-string first encountered within a character string.

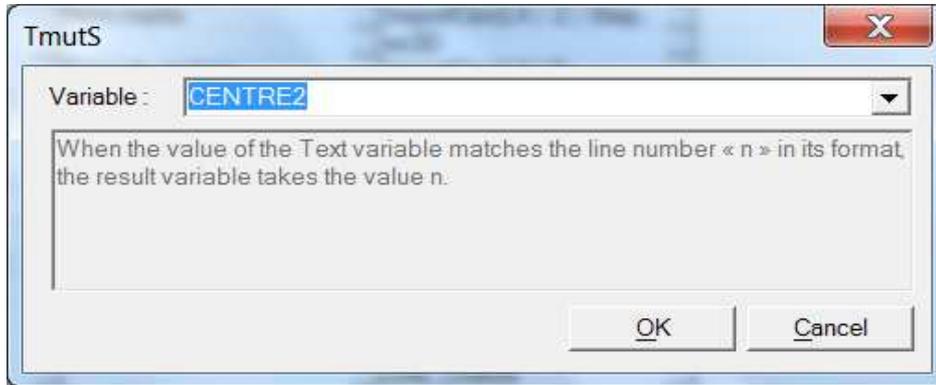
Both the string and sub-string can be specified either as variables or constants.

Conditions

		Resulting Variable	Argument Variable 1	Argument Variable 2	Base
Variable Attribute	Type	S	T	T	L
	Level	i	i or <i	i or <i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned	undimensioned
		dimension n	dimension n	undimensioned	undimensioned or dimension n

TmutS function

With function TmutS, a variable of type Single obtains its value from a variable of type Text : resulting values 1 to n corresponding to the n lines of the format of the variable text:



Conditions

		Resulting Variable	Argument Variable	Base
Variable Attribute	Type	S	T xith format	L
	Level	i	i	i or <i
	Dimension	undimensioned	undimensioned	undimensioned
		Dimension n	dimension n	Undimensioned or dimension n

When the value of the argument variable matches the line number « n » in its format, the resulting variable takes the value n.

Comments :

- 1) The format of the argument variable should contain only unique values (no sub-totals), otherwise there could be inadequate results.
- 2) Of course, it is useful to give a format to the resulting variable, with the codes 1 to n in lines 1 to n, and by copying the labels of the argument variable.
- 3) Using TmutS allows to work on a Single variable rather on a Text variable, which presents huge advantages : sub-totals are easier to build in the format, recoding functions are easier and numerous, etc. Hence, it is highly recommended to use it, as long as the use of the text variable is not limited to a single tabulation, without any recoding.

CPL function (procedure)

A CPL function is a procedure written in the CPL (Cotab Procedural Language) language. This language allows you to perform all sorts of arithmetical, logical, and test operations including iterative loops See Procedures.

Note: CPL function must not be assigned to a dimensioned variable.

External Tables

Concept

Instead of obtaining data directly from a data file, a variable can be configured to obtain it from an External Table. External Tables are single-record type ASCII files in standard or CSV format.

For standard formatted files the records must have a fixed record length.

For CSV formatted files, the **field delimiter** (space, tab, or ; (semi-colon)) is set in the **File>Properties** menu

Each case in the study will be associated with a record in the external table by means of an index key.

You configure a variable to obtain its data from an external table by means of the Import function. There is no limit on the number of variables that an external table can support.

The first step is to create a link, by specifying the external table and index keys.

To create a link

1. Select the **Tools>External Table** menu.
2. Click on the New Std (standard) or the New CSV button as required. The link number is automatically created.
3. Specify the data file to be used as an external table.
4. In Index Key table, for each index key, specify the positioning of the data field in the external table (From/To for a standard ASCII file, and Field no. for a CSV file) and select its associated variable in the study from the drop-down list.

Note: The variable type must be Text, and it can be dimensioned. If several keys are dimensioned for the same table, they must all have the same dimension.

To modify external links

1. Select the **Tools>External Table** menu.
2. Select the desired link number in the Import Link drop-down list.
3. In the Index Key table, make any necessary changes. See [Step 4](#), creating a link above.

To delete a link

1. Select the **Tools>External Table** menu.
2. Select the desired link number in the Import Link drop-down list.
3. Click on the Suppress button.

Reading data

You read data from an external table into a variable by means of the [Import](#) or [ImportCSV](#) function. These functions are similar to the Input functions except that they access external tables instead of the main data file.

If one of the link keys is dimensioned, each case in the data file will be associated with several records (not one record) in the external table. The number of associated records will equal the dimension of the link key.

Bases

This "Base" field for a variable in the Variables Dictionary may be used to reference an existing logical variable positioned earlier in the list.

When a variable is thus conditioned by a "Base" variable:

- The function will not be applied to the variable if the base logical variable has the value "false".
- The variable will be automatically filtered according to this Base when it appears in any table.

The logical variable specified in the Base field may be dimensioned, but only if the calling variable (or an argument variable in some cases, see the Limitations tables) bears the same dimension.

Notes

- For an Input or InputCSV function, the base will not be taken into account at the time that it is executed ; it will be executed even if the logical base variable has the value "false".
- The base will only be taken into account (and the value of the variable set to MISSING if the logical variable has the value "false) after all the variables have been computed i.e. at the moment that control is passed to the calling application (tables, export etc.)
- A variable possessing an Input or InputCSV function may therefore have a value other than MISSING at the moment that all the other variables are calculated, and yet end up with the value MISSING.

Execution

General

Variables are computed sequentially, in first to last order, whatever be their level or type, with the following conditions applying:

- Input functions are executed first, after having read a complete data record from the input file.
- The other functions are executed in a 2nd pass (when all the data records for the case have been processed as above).
- The Rank function is executed last, since all variables have to have been processed before VoxcTab can decide if a case currently being handled will be accepted. Hence, a variable created with the Rank function must never be used as an argument to a function used in another variable.
- For a study with levels, functions are executed level by level. This means that for a case at level *i*, only functions relating to resulting variables at level *i* will be executed.

Processing of missing values (NA)

During the variables creation phase, a variable is classified Missing if it does not have a value, or if its value is incompatible with its type (for example, a letter in a Single type variable). This applies regardless of the values specified in the Format attached to the variable, and during all other phases such as creation of tables, exportation etc.

This definition changes for Single, Multiple or Text variables which have a Format: in fact, in tables it is normal to expect that all cases which do not belong to any of the formatted categories be grouped into the N option line (No Response). The procedure is as follows:

After all variables have been calculated for a case, and before handing control over to the program being executed (tables, weighting, exportation etc.), for all variables which have a Format, the program verifies that their values correspond to their Format definitions (i.e. values or value ranges). Whenever value does not correspond, the variable in question is classified as Missing.

For example, take a variable X that has a Format comprising values 1 to 10. If the variable takes on the value 99, this value will be available to a variable Y which is based on variable X; however, it will be replaced by Missing when the set of variables is passed to the program which calls them. Hence, this value of 99 will not be visible in Preview, but can be used in the construction of variable Y.

Thanks to this procedure, in tables, the N option in templates encompasses all cases that are not catered for in the Format.

In summary, the classification as Missing of a value not prescribed in a variable's Format takes place after all the variables have been calculated for a case; and remains so for all programs: preview, frequency table, export, weighting, and tables.

Process

Study without levels, and single data record per case

1. Reads a record
2. Executes all input functions
3. Executes all other functions, according to their bases
4. Executes the Rank function
5. Applies the Base to variables using an input function.
6. Verifies the values of all variables possessing a Format, and sets to MISSING all those whose value does not conform to the format.
7. Sends the case to the module being run (tables, weighting etc.) provided any universe defined in the Study Structure has been verified.
8. Repeats from Step 1 until end of the file has been reached.

Study without levels with multiple data records

1. Reads a record
2. Executes all input functions for the record.
3. Repeats from Step 1 until all of the records for the current case have been processed.
4. Executes all other functions, according to their bases

5. Executes the Rank function
6. Applies the Base to variables using an input function.
7. Verifies the values of all variables possessing a Format, and sets to MISSING all those whose value does not conform to the format.
8. Sends the case to the module being run (tables, weighting etc.) provided any universe defined in the Study Structure has been verified.
9. Repeats from Step 1 until end of the file has been reached.

Study with levels

1. Reads a record
2. Executes all input functions for the record.
3. Repeats from Step 1 until all of the records for the current family case have been processed. A family is defined as a case at Level 1 and all its dependents.
4. Executes all other functions, according to their bases, in sequential order. For a variable at Level 1, since there is only one case per family, each function will only be executed once. For a variable at a lower level (higher level number (2, 3 etc.) since there can be several cases at this level, the function will be executed for each case at this level, before proceeding to the next variable.
5. Executes the Rank function for each case.
6. Applies the Base to variables using an input function.
7. Verifies the values of all variables possessing a Format, and sets to MISSING all those whose value does not conform to the format.
8. Sends the Family to the module being run (tables, weighting etc.). VoxcoTab sends the cases one at a time, starting with the case at level one, followed by each dependent at the next level in the dependence tree etc. This procedure ensures that each parent will always precede its dependents, and that cases from different dependence chains will never be found together i.e. cousins will never be found together.
9. Repeat from Step 1 until end of the file has been reached.

Special Case for variables at Level 0

In a study with levels, variables usually belong to one of the levels defined in the Study Structure; however, they can also belong to level 0. By convention, a variable belonging to level 0 is considered to belong to all levels, meaning that it will be computed for each case in every level.

Generally, level 0 is used to define level identifiers such as sequence numbers computed with the Rank function, or to explicitly register the case's level (for inclusion in an exported file, for example).

Execution of multiple files

All VoxcoTab operations (tables, exportations etc) can be performed on one or several files (several survey waves, for example). The files are processed sequentially as one file, and all you have to do is name them all in the file specification dialog window concerned.

External tables

Every time an import function is executed, VoxcoTab looks in the specified external table for the record which matches the assigned index keys, given their current value (if more than one record matches, the first one will be picked).

The variable to be imported is read from this record. It is possible, therefore, that for the same individual, 2 variables using the import function to access the same external table might obtain data from 2 different records. This could happen if the value of the index key variable is changed between reading the first and the second variable (through a procedure).

Note: If the index key is dimensioned, the variables read through importation from a corresponding external table must also be dimensioned; each occurrence is read from the external record which corresponds to the occurrence of the matching index of the key.

Cache file

This option is designed to speed up the execution of resource intensive studies, for example comprising tens of thousands of cases.

In fact, running any application (tables, exportation etc.) consists, for each case, of first calculating its set of variables and secondly executing the application (tables, exportation etc.); this process is repeated for each case in the data file.

Sometimes the first phase (calculation of variables) can take much longer than the second (if there are many variables, or if procedures for calculating variables are particularly complex). When executing several table commands, one after the other, this can be costly in time since the variables calculation is repeated each time.

When you use the Cache option (Tools>Properties>Use Cache) the results of variable calculation when executing the first set of cases will be written to the Cache file. When the next command is executed (tables, exportation etc.) the program will automatically read the Cache file instead of reading the data file and recalculating all the variables again.

The Cache file will remain valid so long as the variables dictionary has not been modified. If the variables dictionary is modified (new variable, change made to a function etc.) the Cache file will be automatically erased and re-built.

Warning: Neither the Bases of a variable, nor the use of Universes in study structures, are taken into account to assess whether the variables dictionary has been modified. In this situation you must erase/reset the Cache file yourself, using the **Tools>Reset Cache** command.

Execution of several data files. Whether the files are called in several consecutive executions or in a single execution, the Cache file contains a sub-file for each data file. Hence, in a subsequent execution, the program will access the appropriate cache file(s).

Multiple wave study. When executing a command comprising several waves, the program will use existing cache sub-files, and create a new one for each new wave.

Preview

The Preview command, accessed through the Tools>Preview menu or the Execute button  in the Toolbar starts reading the source data file, and calculates the variables for the first case. The values of these variables are displayed in the Preview column in the Variables screen. When you click on the header of the Preview column the next case will be read, calculated and displayed.

N°	Name	Level	Dim.	Type	Format	Title	Function	Base	Preview (>>)
1	Id	1		Single	Id1(3)	Case Identifier	Input(T=3/1-2)		2
2	Income	1		Quantity		Income	Input(T=1/4-8)		78000
3	Region	1		Single	Region1(3)	Region	Input(T=1/10)		1
4	Member	2		Single	Member1(3)	Member	Input(T=2/12)		1
5	Sex	2		Text	Sex(2)	Sex	Input(T=2/14)		F
6	Age	2		Single	Age(8)	Age	Input(T=2/16-17)		19
7	Trip	3		Single	Trip1(3)	Trip	Input(T=3/19)		1
8	Services	3		Multiple	Services1(4)	Services used	Input(T=3/21 * 4)		.1,3
9	Satisfaction	3	3	Single	Satisfaction1(7)	Level of satisfaction with:	Input(T=3/29 / Step=2)		.
...	L<1>					Speed			5
...	L<2>					Service			2
...	L<3>					Comfort			5

Comment: When working with a dimensioned variable, if you click on the + sign on the left of the dimension field or press the space bar when it is selected, it will be expanded to display all of its occurrences.

Frequency Tables

You can quickly display a variable's frequency distribution, regardless of its type.

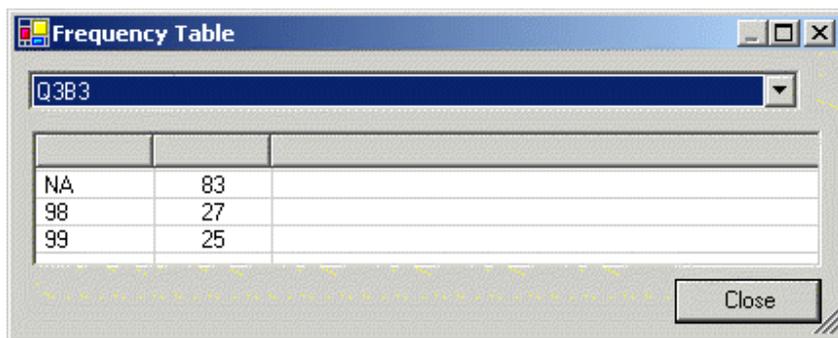
To obtain frequency distributions

Do one of the following:

- Select **Utilities>Frequency table** in the main menu, and then choose the variables you are interested in.
- **Right-click** on a variable in the variables list, and select Frequency table in the pop-up menu.

Once you have chosen the source data file when prompted, you will be presented with the frequency distribution (s) of the chosen variable(s).

Each value encountered in the variable is shown in the left-hand column together with its frequency in the right-hand column, regardless of any categories which may have been defined in the variable's Format. However, when the variable does have a Format, any values which are not defined in the format are categorized as NA (Missing), and will not appear in the displayed list.



NA	83	
98	27	
99	25	

For Quantity variables, the mean, standard deviation, median and 1st and 3rd quartiles are shown as well.

Templates

Overview - Templates

In tables, each variable invoked is associated with a Template, whose role is to define which Options (counts, statistics etc.) of the variable are to be applied/displayed.

These Template Options are chosen out of the following list (the default labels describing them can be customized in the Tools>Preferences menu):

	Option	Description
T	Total	The total for the variable taking into account any base that might be associated with it
N	No response	Corresponds to the number of MISSING values for the variable, as defined in the description of Variable Attributes . This applies to: <ul style="list-style-type: none">• Variables without a Format specification: all true Missing values are included.• Variables with a Format specification: all values not described in the category values or value ranges.
D	Distribution	Refers to the list of categories in the variable's Format . This option requires that the variable has a Format.
A	A least one response	The total number of respondents to the variable. Hence, the opposite of the N option (no response).
C	Sum	The sum of a variable's values (real numbers for Simple or Quantity type variables, or the value that might be assigned to the variable's categories in an associated Format.
M	Mean	The mean of the variable's values which qualify for the Sum (see above).
Mi	Minimum	The minimum value of the variable which qualifies for the Sum.
Ma	Maximum	The maximum value of the variable which qualifies for the Sum.
NB	Average number of responses	Average number of responses for a multiple variable
NT	Number of responses	The number of responses for a multiple variable
LO	List of occurrences	For a dimensioned variable, this list allows each occurrence to be customized, by editing them in various positions (rows, columns or pages).
W	Weighting	

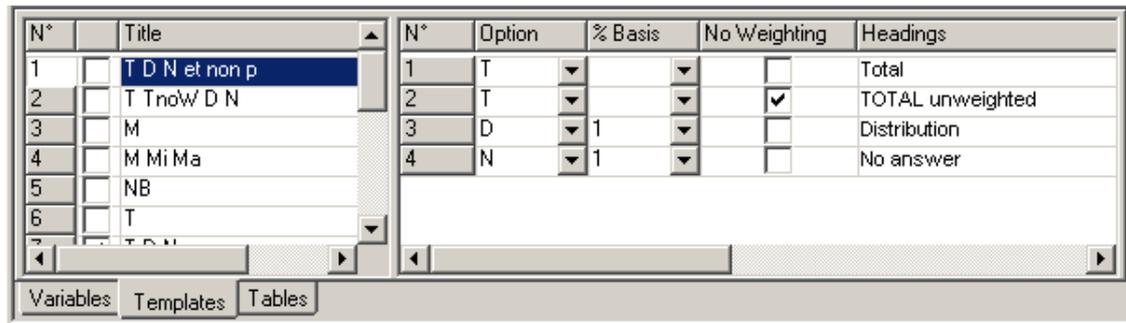
A template comprises one or more of these Options, each comprising several attributes or settings.

The Check button in the Toolbar allows you to check that there are no errors in the variables or templates.

Managing Templates

The Templates window is brought into view when you click on the Templates tab at the bottom-left of the screen. It is divided into 2 panes.

- The left-hand pane displays the list of the current study's templates to which you can add and delete.
- The right-hand pane shows the attributes of the template currently selected in the left-hand pane.



You can perform the following actions in the templates list when a template row is selected (in the left-hand pane):

Operation	Main Menu	Mouse click	Toolbar
Add (Insert) Note: Specify a template number (1 to 999) and title	Edit>Insert	Right-click	
Delete	Edit>Delete	Right-click	
Duplicate	Edit>Duplicate	Right-click	
Rename		Right-click, Double-click	
Import Note: If the template number already exists, you are prompted to change its number or substitute the existing template.		Right-click	
Copy from another study	Edit>Copy from the source study Edit>Paste into the current study	Right-click	

Default template: One of templates in the list must be chosen as the default template, by clicking in the box situated to the left of its title. This will be the one applied to variables to which no template has been specified.

Note: When the Title field is selected, with a right-click you can copy, paste or clear the title.

Template Options

In the right-hand pane, when you have selected a template in the list, you can add as many rows as you wish; each row representing an Option. Managing Options

When a row is selected, a right-click provides access to standard edit functions: insert, delete, duplicate, copy/paste (rows), and font selection.

Operation	Main Menu	Mouse click	Keyboard	Toolbar
Add (Insert)	Edit>Insert	Right-click		
Delete	Edit>Delete	Right-click		
Duplicate	Edit>Duplicate	Right-click		
Fonts		Right-click		
Copy/Paste (rows)	Edit>Copy Edit>Paste	Right-click		 
Move row	Edit>Move Up Edit>Move Down		Ctrl+Up arrow Ctrl+Down arrow	

Managing Option fields

The fields, and edit methods for defining each option are:

Field	Comment	Mouse clicks	Keyboard
Option	See Templates Overview	Double-click in the field, or click on the down-arrow on the right of the field to access the Option list	Type the Option's first letter (uppercase)
% Basis	You can designate an existing Option (by specifying its line number) which is to be used as the base in calculating the percentage. The base option must only be of type T, C, NT or A. Note: A % Basis cannot be used for Option types M, NB, Mi and Ma.		
Suppress Weighting	With this checkbox selected, corresponding items will not be subjected to any weighting.		
Label	This is the Label that will appear in tables using this Option, except for the Options D and LO. Note: The default Label, unless you modify it, is taken from the Label as set in Preferences.	Double-click in the field to edit the Label. The font settings for one or more selected Labels are accessed with	Type in the field to overwrite its contents

		a right-click of the mouse.	
Decimals	The number of to be displayed in percentages or means (depending on the option).		
Place	Note: This field is reserved for future use		
Hide	With this checkbox selected, the corresponding item will not be displayed. It will simply be used in the calculation of other items.		

Notes:

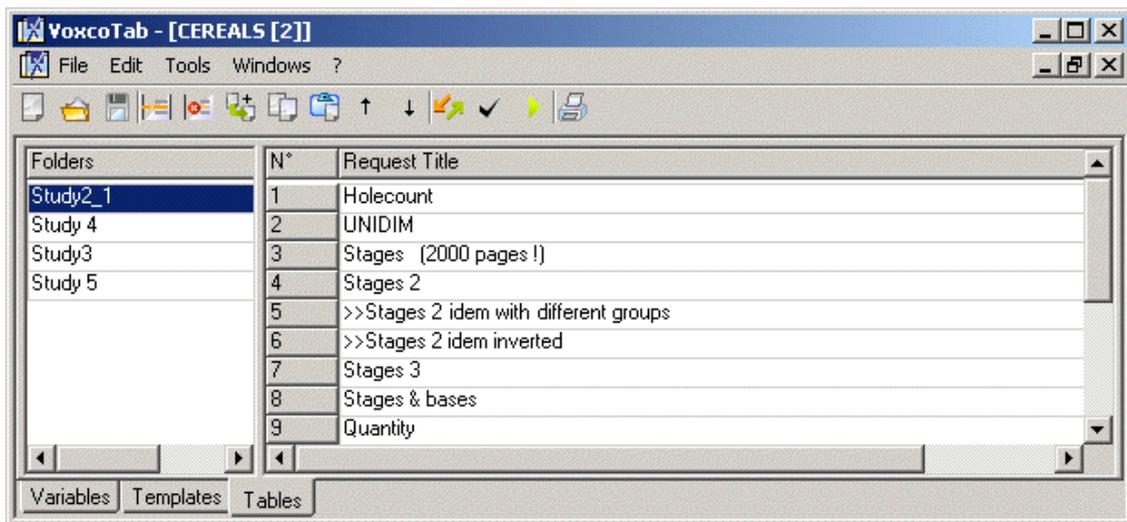
- Copy, Paste and Clear functions apply to one or more cells selected in a column.
- The Font settings for one or more selected Labels are accessed with a right-click of the mouse.

Tables

Managing Table Requests

The Tables window is brought into view when you click on the Tables tab at the bottom-left of the screen. It is divided into 2 panes.

- The left-hand pane displays the list of the current study's Folders to which you can add and delete.
- The right-hand pane shows the Table Requests contained in the Folder currently selected in the left-hand pane. You can add any number of requests to this table.

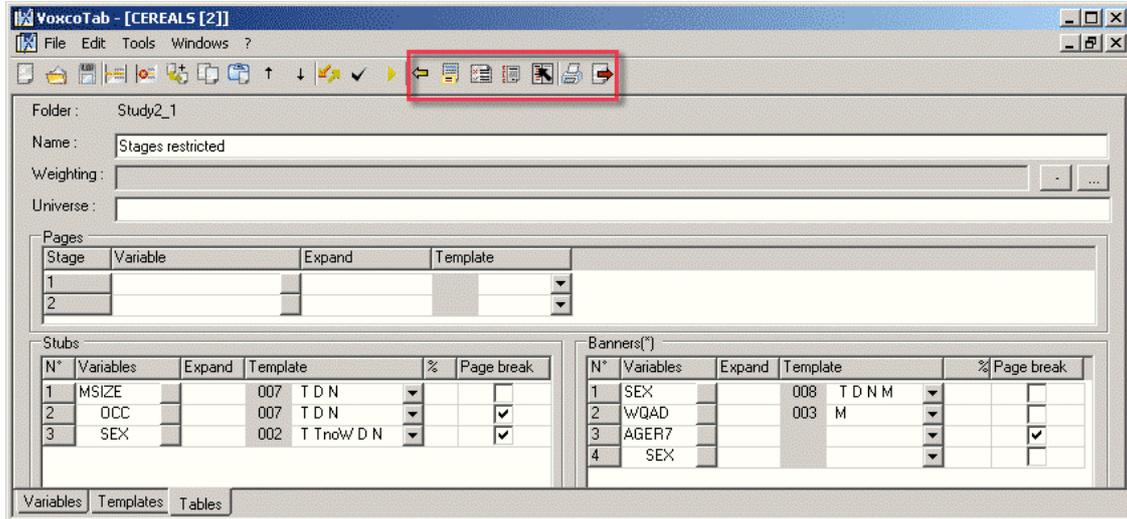


In both panes, you Insert (add), Delete, Duplicate, Import, Copy and Paste Folders and Requests respectively. Import, Copy and Paste functions also work from one study to another. Both for Folders and Requests, the only items to edit are their Titles.

The Check icon in the toolbar tests the currently selected table request, or, if a folder is selected, all the table requests in that folder.

Table Request

The Table Request configuration window is displayed when you double-click on a Table Request (in the Requests pane). Note that the toolbar has new buttons:



Note: Table Request buttons are added on the right-hand side of the Toolbar.

Table Request configuration settings

Setting	Description								
Weighting	An undimensioned quantity variable								
Universe (Filter)	<p>An expression using logical variables exclusively connected with the Boolean operators AND, OR and EX (exclusive). If the result of this expression is TRUE, for any case, then that case will be included in the table. Otherwise, the case will be excluded.</p> <p>The default order of execution is from left to right, but you can use parentheses to specify priority amongst the operators.</p> <p>If using dimensioned variables, they must all have the same dimension.</p> <p>Operator rules</p> <table border="1"> <thead> <tr> <th>Expression</th> <th>Result</th> </tr> </thead> <tbody> <tr> <td>A AND B</td> <td>TRUE if the logical variables A and B are both TRUE</td> </tr> <tr> <td>A OR B</td> <td>TRUE if at least one of the logical variables A and B is TRUE</td> </tr> <tr> <td>A XOR B</td> <td>TRUE provided that only one of the logical variables A and B is TRUE.</td> </tr> </tbody> </table> <p><i>Example of a universe: (Q18 AND MALE) OR (Q25 AND FEMALE)</i></p>	Expression	Result	A AND B	TRUE if the logical variables A and B are both TRUE	A OR B	TRUE if at least one of the logical variables A and B is TRUE	A XOR B	TRUE provided that only one of the logical variables A and B is TRUE.
Expression	Result								
A AND B	TRUE if the logical variables A and B are both TRUE								
A OR B	TRUE if at least one of the logical variables A and B is TRUE								
A XOR B	TRUE provided that only one of the logical variables A and B is TRUE.								

Table unit windows: You can select any of the three windows (Pages, Stubs or Banners) by clicking on it. Its title will have * appended, to indicate that it is the one currently selected.

You can perform the following actions when a row is selected in the Banners and Stubs windows:

Operation	Main Menu	Mouse click	Toolbar
Add (Insert)	Edit>Insert	Right-click	
Delete	Edit>Delete	Right-click	
Duplicate	Edit>Duplicate	Right-click	
Copy/Paste	Edit>Copy Edit>Paste	Right-click	 
Move a row	Edit>Move Up Edit>Move Down	Ctrl+Up arrow Ctrl+Down arrow	 

Pages	<p>A First Stage and/or Second Stage variable within whose categories the Stub and Banner will be crossed.</p> <p>If a Second Stage variable is specified, it will be crossed with the First Stage variable i.e. each of its categories will be broken out within each category of the First Stage variable.</p> <p>See Table request element - attributes</p>
Stubs	<p>Any number of variables to be used for creating rows in the table.</p> <p>Any variable may set to Stage 1 or Stage 2. However, the first one must set to Stage 1. A variable set to Stage 2 will be crossed with the Stage 1 variable which comes before it.</p> <p>See Table request element - attributes</p>
Banners	<p>Any number of variables to be used for creating columns in the table.</p> <p>Any variable may set to Stage 1 or Stage 2. However, the first one must set to Stage 1. A variable set to Stage 2 will be crossed with the Stage 1 variable which comes before it.</p> <p>See Table request element - attributes</p>

About Stages and cross-tabulation

Stages is a feature which enables you to add a further breakdown dimension to a cross-tabulation.

Any variable selected in a table request can be specified as Stage 1 or Stage 2. A Stage 2 variable (which must be preceded by a Stage 1 variable) will be broken down (within its own Banner or Stub) by each category of the Stage 1 variable which precedes it.

When you add a variable to a Stub or Banner, you can add it as a Stage 1 or Stage 2 variable. Stage 2 variables are easy to identify because they are indented in the list. You can change a variable from Stage 1 to Stage 2 and vice-versa by means of the Ctrl+Right/Left arrow keys.

Example: In the screen-shot above, the Stage 2 variables OCC and SEX will be broken down in the Stub by the Stage 1 variable MSIZE. The process is a cross-tabulation in which the category combinations are reflected in the Stub.

The final table comprises a cross-tabulation of all the Page variables, Stub variables and Banner variables.

To execute a request

To execute a request, click on the Execute button in the toolbar .

To return to the Requests folder

To return to the Requests folder, click on the Return button in the toolbar .

Table Request variables - attributes

Each unit (row) in a table request consists of a selected variable together with several attributes:

N°	Variables	Expand	Template	%	Page break
1	MSIZE		007 T D N		
2	OCC		007 T D N		

Variable

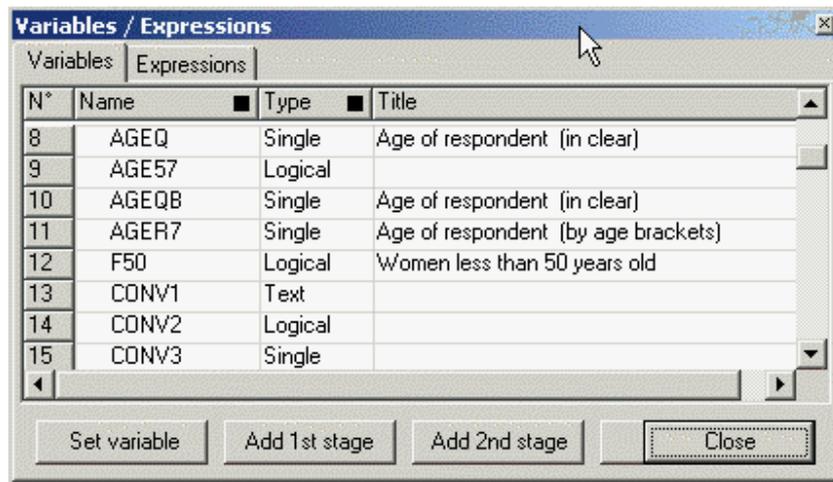
A variable used in the definition of a Page, Stub or Banner can be of any type (single, multiple, text, quantity or logical) or level (if the study has levels), and they may be dimensioned.

If the only dimensioned variables are in a Stub (and they are not expanded), they can have different dimensions. An exception to this however, is for a pair of Stage 1 and Stage 2 variables; they cannot have different dimensions. The same applies to dimensioned variables in a banner i.e. provided there are no dimensioned variables in the Page, Stub or Universe.

In all other situations, all un-expanded dimensioned variables specified in the table request must have same dimensions.

To add a new variable

To add a new variable, or assign a new variable, click on the button on the right of the field. This brings up the Variables Selection List (the Variables Tab in the Variables/Expressions window):



Button	Description
Set variable	Sets the chosen variable (as a 1st Stage variable) for the currently selected unit (line)
Add 1st stage Add 2nd stage	Adds the chosen variable(s) as 1st or 2nd Stage variables after the currently selected unit (line)

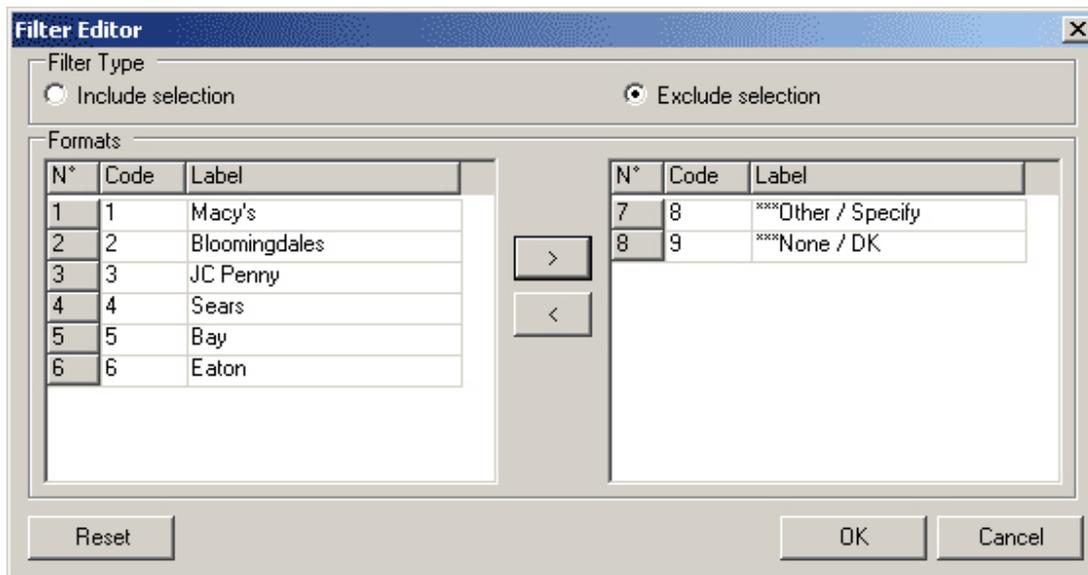
Note:

- To assist in choosing your variable, you can narrow down the variables by Type or show variables in alphabetical order. Click in the Name or Type column header.

- You can display all the occurrences of a dimensioned variable by clicking on its + sign prefix. This allows you to choose one or more of them.

To filter (customize) the selection of a variable's categories

When you have a variable selected (whether in the Pages, Stubs or Banners area), a right-click provides access to the Filter Editor (in addition to Copy, Paste and Reset functions). The Filter Editor allows you to include or exclude specific category codes for the variable. **Note:** It only applies to variables using the template option D (Distribution).



The full set of the variable's formatted category codes are initially displayed in the left-hand list. The codes that you wish to include or exclude, you then move to the right-hand list by using the arrow buttons between the lists. The radio buttons Include/Exclude selection determine whether the codes in the right-hand list will be included or excluded respectively.

This option operates on display only, which means that you can modify it after a table has been executed. It applies only to the situation in which it is specified i.e. even if the same variable appears elsewhere in the same table, it will not be affected by this filter.

Expand

A dimensioned variable which has been selected globally can be expanded by clicking on its Expand checkbox. The result is equivalent to having selected each occurrence individually. Hence, the constraints on using dimensioned variables described above do not apply when the variable is Expanded.

Stubs			
N°	Variables	Expand	Template
1	S1S[12]	<input checked="" type="checkbox"/>	
2	S2A[7]	<input checked="" type="checkbox"/>	
3	S2B[7]	<input checked="" type="checkbox"/>	
4	S4A[18]	<input type="checkbox"/>	
5	S5AA		
6	S5AB		

Template

If no template is assigned, the default template will be used.

To assign a template to a variable

- Type the template number in the variable's template field, or
- Select a template from the list that appears when you double-click on the variable's template field, or simply click on the button to the right of it.

Note: A template assigned to a logical variable can only contain the following options:

Option	Description
T	Total
A	Value TRUE
N	Value FALSE
LO	Applies to a list of occurrences (dimensioned variable)

Weighting - W option

This options enables you to create tables using several weighting variables. It can be applied to pages, columns or rows in the table.

Use of the W option is equivalent to specifying a weighting variable in the table request.

Comments:

1. Regardless of whether the W option is applied to a rows, columns or pages, it must be accompanied by another option - at the 1st stage if W is at the 2nd stage, and at the 2nd stage if W s at the 1st stage. Hence, if there is only one stage, W should not be used.
2. If there are several W options in the table, only one is applied in each component table, in the following priority order: column (stage 2, if not stage 1); if not, in rows; if not, in pages; if not, in the table's weighing field.

Percentages (%)

This field is generally not used. It allows you to modify the definition of the base for percentages contained within the template associated with the variable. If you enter a number, the base for percentages will be taken from the variable/template pair bearing this number. This enables you to apply a percentage base which is totally independent of the variable using it.

Nevertheless, certain restrictions need to be respected:

When a variable A references, for a percentage base, a line in the table using variable B, then...

- If variable A is Stage 1, then variable B must also be Stage 1.
- If variable A is Stage 2, either variable B is Stage 2 in the same group i.e. they must both depend on the same Stage 1 variable, or variable B is Stage1, hence in a group other than A.

Moreover, in the template associated with variable A, if the percentage base is line j, then line j of the template associated with variable B must be of type T,C or A only.

Generally speaking, this Percentages field is used simply to refer to a line containing a logical variable associated with a template comprising a single option. Hence its use is very straightforward.

Note concerning percentages: Percentages are always computed at the last Stage. This means that when there are 2 stages, any percentages requested with the variable at the 1st stage are not taken into account. Similarly, a request for a different base (see above) will not be taken into account if the line referenced is Stage 1 and if it is followed by a 2nd stage.

Page Break

When this checkbox is selected, a page break will be included in the output before the tables using this variable. This applies for PDF and Excel-XML output.

You can select/deselect the page break option for several variable at a time in a table request. Simply select the variables, and use the **Check All** option available with a right-click of the mouse.

Note: Copy, Paste and Clear functions can be used on multiple fields in the Variable and Template columns.

Execution of a table request

Constraints on a table request

When you click on the Check button  or the table request Execute button , VoxcoTab checks the Variables and Templates specified in the table request. If errors are encountered they are reported, and the execution does not take place. Checks include the following:

- Referenced variables must exist.
- Weighting variables must be Quantity type, and undimensioned.
- Universe (filter) variables must be Logical type.
- Page, Stub or Banner variables whose template includes a D option (distribution) must have a Format.
- The options C, M, Mi, Ma in a template imply that the variable applying them is Single or Quantity type. Or, if they are Multiple or Text type, their categories must have been assigned values in their corresponding Formats.
- The W option in a template implies that the calling variable is of Quantity type, and that it is associated at the 1st stage (or 2nd stage, depending on the circumstances) with another option (other than W).
- The NB and NT options imply that the variable calling it is Multiple type.
- The LO option implies that the variable calling it is dimensioned.
- Unexpanded dimensioned variables may be used in a Page, Stub, Banner or Universe, provided they have the same dimension. However, if the only dimensioned variables are in a Stub (and they are not expanded), they may have different dimensions. An exception to this however, is for a pair of Stage 1 and Stage 2 variables; they must not have different dimensions. The same applies to dimensioned variables in a Banner i.e. provided there are no dimensioned variables in the Page, Stub or Universe. In all other situations, all un-expanded dimensioned variables specified in the table request must have the same dimensions.

Errors are displayed in the new window below the requests. When you click on the error message, the request line at fault is highlighted in red.

Execution.

If no errors are detected, the data file is read case-by-case. Provided that the conditions for inclusion in the tables are satisfied i.e. compliance with the expression used in the Universe (filter), the data are broken out into a table comprising a set of component tables each of which is defined by a Page variable (or pair of them if there is a 2nd Stage variable), a Stub variable (or pair of them if there is a 2nd Stage variable), and a Banner variable (or pair of them if there is a 2nd Stage variable), taking into account the various Options and settings of their associated templates.

The breakdown in a component table depends, however, on the bases that might be applied to the Stub, Banner or Page variables according to the following rules:

- A base assigned to a cross-tabulation variable (Stub, Banner or Page) will be taken into account. There are, however, two exceptions:
 - The dimension of the base is greater than the dimension of the variable to which it is applied; which is to say that the base is dimensioned while the variable is undimensioned. This can occur when using the NbVal function, for example.
 - The level of the base is greater than the level of the variable.
- If a base applies, and if its value is FALSE, the component table will not include the case.

If there is Weighting specified, a MISSING value in the referenced weighting variable limits the case from being counted in a table cell to when it is set to "unweighted" i.e. for those whose template (Stubs, Banners or Pages) includes the "unweighted" option.

Component tables

The set of component tables (which make up a complete table) is determined by the crossing of Page, Banner and Stub **Chains**.

Stage 1 variables only: When variables are Stage 1 only, a **Chain** is the same as an **Element**, which is the union of a **Variable** and an **Option** in its associated Template. There are as many Chains as there are Variable/Option combinations.

Stage 1 and Stage 2 variables: When there are Stage 1 and Stage 2 variables involved (in a Page, Banner or Stub), the result is equivalent to a cross-tab within the Page, Banner or Stub. Hence, there is a **Chain** for each Stage 2 Element paired with each of its Stage 1 Elements.

Example without Stages:

	Variable	Template Options	Element/Chain
Page Variable	SEX	T (Total) D (Distribution)	SEX.Total SEX.Distribution
Banner Variable	REG	T (Total) D (Distribution)	REG.Total REG.Distribution
Stub Variables	Q5	T (Total) D (Distribution) N (No Response)	Q5.Total Q5.Distribution Q5.No Response
	Q6	T (Total) D (Distribution) N (No Response)	Q6.Total Q6.Distribution Q6.No Response

The total number of component tables will be 24 i.e. the product of the Page, Banner and Stub Element/Chains.

Example with Stages

		Variable	Template Options	Element	Chain
Page Variable		SEX	T (Total) D (Distribution)	SEX.Total SEX.Distribution	
Banner Variable		REG	T (Total) D (Distribution)	REG.Total REG.Distribution	
Stub Variables	Stage 1	Q5	T (Total) D (Distribution) N (No Response)	Q5.Total Q5.Distribution Q5.No Response	Q5.Total-Q6.Total Q5.Total-Q6.Distribution Q5.Total-Q6.No Response
	Stage 2	Q6	T (Total) D (Distribution)	Q6.Total Q6.Distribution	Q5.Distribution-Q6.Total Q5.Distribution-Q6.Distribution

			N (No Response)	Q6.No Response	Q5.Distribution-Q6.No Response Q5.No Response-Q6.Total Q5.No Response-Q6.Distribution Q5.No Response-Q6.No Response
--	--	--	-----------------	----------------	--

The total number of component tables will be 36 i.e. the product of the Page, Banner and Stub Element/Chains.

Component table breakdowns

Non-quantitative component table: None of the options C, M, Mi, Ma, NT or NB appear in any of the chains. A case (optionally weighted) will be broken down into all the cells of the table for which the case applies i.e. as defined by the crossing of all the pages, columns and rows, and for which any associated bases are true.

Quantitative component table: At least the one of the options C, M, Mi, Ma, NT or NB appears in one of the chains. In this case, the value of the corresponding variable will be broken down into all cells as defined above.

- C option. The value of the corresponding variable is simply summed (multiplied by a weighting factor, if applicable) in the cells.
- M option. The value of the variable is summed (weighted if applicable), the squared values are summed (weighted if applicable), and the number of cases (weighted if applicable) is computed for eventual calculation of the mean and standard deviation.
- Mi or Ma option. The minimum or maximum (excluding missing values) is displayed in each cell that applies.
- NT option. The number of values of the corresponding variable are cumulated (weighted if applicable) in the cells.
- NB option. The number of values (weighted if applicable) of the associated multiple variable is summed for calculation of the mean.

Component table with several quantities: The options C, M, Mi, Ma, NT or NB appear more than once in the six possible elements constituting a component table. Since a cell in a component table can only involve a single quantity, it is chosen in the following priority order: columns, by default in rows, by default in pages, and in each of the dimensions, firstly at the 2nd stage, if it exists, if not by the 1st stage.

Dimensioned component table: This occurs where at least one of the variables is dimensioned and unexpanded i.e. a universe or chain variable.

In this case, let n be the number of occurrences of the dimensioned variable (remembering if there are several dimensioned variables constituting a component table, they must have the same dimension). Each case will be broken down n times in the component table successively for occurrence 1, occurrence 2 etc.

The LO option in a template enables the breakdown and display of each occurrence.

Studies with Levels

The breakdown depends both on the levels defined for the study, and the different variables specified for the table. The level of a component table, as defined by its weighting, universe (filter), page, row and column variables, is equal to the lowest level encountered in this combination (i.e. the level bearing the highest number).

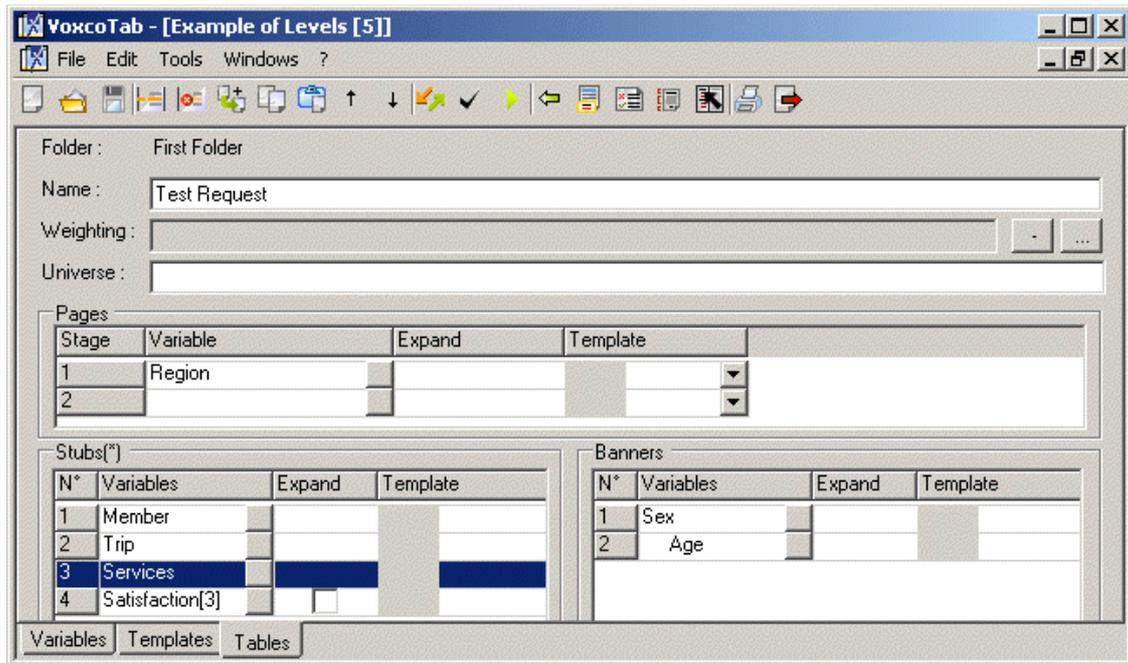
The general rule is that a component table at level j is populated by the data gathered at level j , but this does not prevent the inclusion of data from a higher level (i.e. from a level bearing a small number).

Example:

Assume that we have a study with the following variables:

Level	Variables
Level 1 (Households)	REGION, INCOME
Level 2 (Individuals)	MEMBER, SEX, AGE
Level 3 (Trips)	TRIP, SERVICES, SATISFACTION

and that we issue the following Table Request:



The component table (REGION, SEX-AGE, MEMBER) will contain Members

The component tables (REGION, SEX-AGE, TRIP), (REGION, SEX-AGE, SERVICES), (REGION, SEX-AGE, SATISFACTION) will contain Trips.

Percentages (example)

This example is based on the following Template and Table Request:

Template

N*	Title	N*	Option	% Basis	No Weighting	Headings	Decimals	Place	No Edition
1	<input checked="" type="checkbox"/> T D A N and T unweighted	1	T	1	<input type="checkbox"/>	TOTAL	0	NONE	<input type="checkbox"/>
2	<input type="checkbox"/> T D N and A	2	D	3	<input type="checkbox"/>	Distribution	0	NONE	<input type="checkbox"/>
3	<input type="checkbox"/> A	3	A	1	<input type="checkbox"/>	Respondents	0	NONE	<input type="checkbox"/>
		4	N	1	<input type="checkbox"/>	No response	0	NONE	<input type="checkbox"/>
		5	T		<input checked="" type="checkbox"/>	Unweighted Total	0	NONE	<input type="checkbox"/>

Table Request

Folder : Tables 1
 Name : New request
 Weighting :
 Universe :

Pages

Stage	Variable	Expand	Template
1			
2			

Stubs

N*	Variables	E...	Template	%	Page break
1	STATE	001	T D A N and T unweighted		<input type="checkbox"/>
2	SEX	001	T D A N and T unweighted		<input type="checkbox"/>
3	TYPE	002	T D N and A	4	<input type="checkbox"/>
4	SHOP	003	A	4	<input type="checkbox"/>

Banners(*)

N*	Variables	E...	Template	%	Page break
1	Q1A	001	T D A N and T unweighted		<input type="checkbox"/>
2	Q3B1	002	T D N and A	3	<input type="checkbox"/>
3	Q4A	003	A		<input type="checkbox"/>
4	SHOP	001	T D A N and T unweighted	1	<input type="checkbox"/>

Column Percentages

Q1A: The columns Total, No Response and At Least One Response use the column Total of Q1A as the base for percentages. The columns for the distribution (categories) of Q1A use as the base for their percentages the column At Least One Response for Q1A.

Q3B1: All the columns use the column Q4A as the base for their percentages.

SHOP: The columns Total, No Response and At Least One Response use the Total Column of Q1A as the base for percentages. The columns for the distribution of SHOP use as the base for their percentages the column At Least One Response for Q1A.

Row Percentages

STATE: The rows Total, No Response and At Least One Response use the row Total of STATE as the base for percentages. The rows for the distribution (categories) of STATE use as the base for their percentages the column At Least One Response for STATE.

TYPE: All the rows requiring a percentage, i.e. Total, No Response, At Least One Response (TRUE), and Distribution, use the row SHOP as the base for the percentages. This applies to all the categories belonging to SEX i.e. Total, Male, Female, No Response, At Least One Response, and Unweighted Total.

SHOP: The percentage base is the row for SHOP itself.

Presentation of Results

Preview

The **Preview button**  in the Toolbar displays a Preview of the table. It also serves to toggle back and forth between the Table Preview and the Table Request that produces it.

If **Page variables** have been specified in the Table Request, you can select any page from the Page drop-down menu in the upper part of the window.

You can **re-size column widths** and row heights by click-dragging on any cell border.

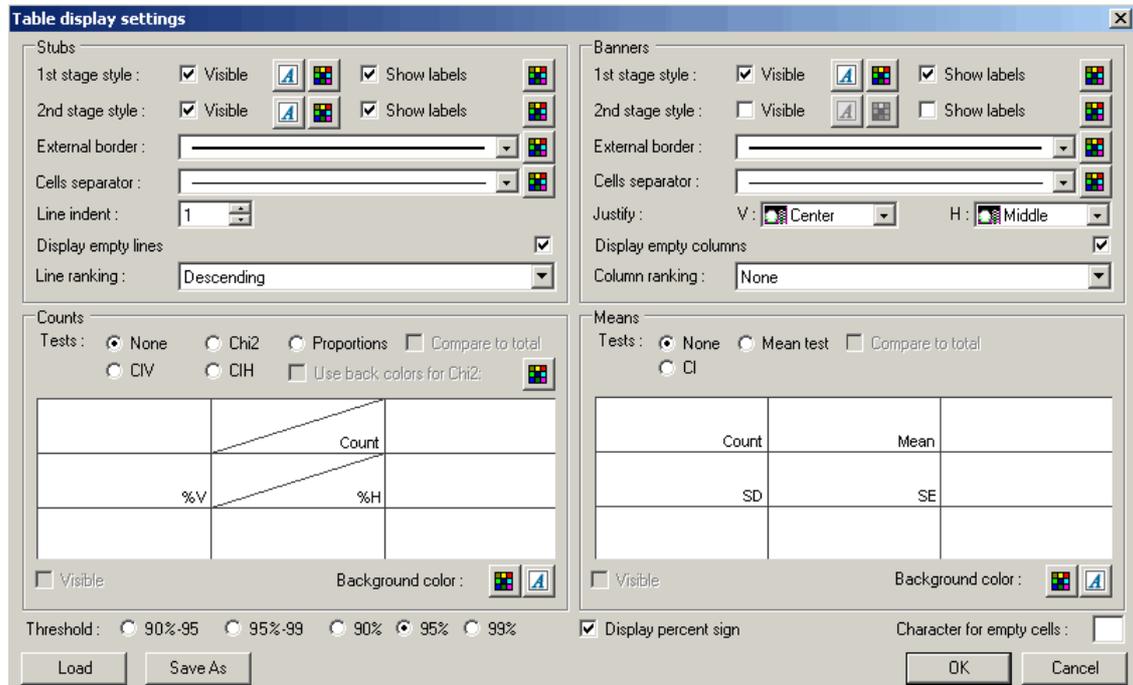
Folder : Tables 1
Request : Table 1a
Page :

	Sex of Respondent						Is your age within the following range...											
	TOTAL	Male	Female	Respondents	No response	Unweighted Total	TOTAL	18-25 years	26-45 years	46-								
All stores from both questions																		
TOTAL	118 100	100	14 100	36 100	25 100	64 100	39 100	33 100	79 100	67 100	118 100	0	118 100	100	8 100	15 100	7 100	13 100
Macy's	28 52	100	6 43	35	11 44	65	17 44	61	11 73	39	28 52	0	28 52	100	8 100	30 100	3 43	11 43
Bloomingdales	19 35	100	5 36	38	8 32	62	13 33	68	6 40	32	19 35	0	19 35	100	2 25	11 100	2 29	11 100
JC Penny	18 33	100	2 14	14	12 48	86	14 36	78	4 27	22	18 33	0	18 33	100	2 25	11 100	3 43	17 100
Sears	19 35	100	5 36	33	10 40	67	15 38	79	4 27	21	19 35	0	19 35	100	1 13	6 100	2 29	10 100
Bay	20 37	100	4 29	31	9 36	69	13 33	65	7 47	35	20 37	0	20 37	100	1 13	5 100	10 100	
Eaton	24 44	100	4 29	25	12 48	75	16 41	67	8 53	33	24 44	0	24 44	100	2 25	8 100	5 43	10 100
***Other / Specify	20 37	100	3 21	21	11 44	79	14 36	70	6 40	30	20 37	0	20 37	100	3 37	10 100	7 43	10 100
***None / DK	14 26	100	5 36	42	7 28	58	12 31	86	2 13	14	14 26	0	14 26	100	2 25	8 100	5 43	10 100
Respondents	54 46	100	14 100	36	25 100	64	39 100	72	15 19	28	54 46	0	54 46	100	8 46	30 100	3 43	11 43
No response	64 54	100							64 81	100	64 54	0	64 54	100				
Unweighted Total	118 100	100	14 100	36	25 100	64	39 100	33 100	79 100	67 100	118 100	0	118 100	100	8 100	15 100	7 100	13 100
Over the last month, about how many times did you go shopping at ... / Macy's																		
TOTAL	118 100	100	14 100	36	25 100	64	39 100	33 100	79 100	67 100	118 100	0	118 100	100	8 100	15 100	7 100	13 100
Once	11 100	100	3 100	50	3 100	50	6 100	55	5 100	45	11 100	0	11 100	100				

A right-click in the table window pops-up a menu from which you can Execute a Table Request, access Table and Page Settings, view the Page Index, perform a Gamma Test, and Refresh the screen.

Table Display Settings

The Table Display Settings window is accessed through its button  in the toolbar:



The various settings presented in this window, which allow you to customize the table display, apply equally to screen display, printed display, or display exported to Excel.

The combination of settings in this window can be given a name and saved as a Style (see the Save As button above). You can Load the settings of any existing Style (see the Load button above).

Note: In the **Tools>Table/Page Settings** menu, you can add, delete and edit Table Settings Styles.

Stubs

The settings for Stubs are presented in the upper-left area of the window:

Setting	Description
1st and 2nd Stage Style	<p>Show Titles: This checkbox allows you to show/hide the variable titles.</p> <p>: These buttons access dialog windows where you can set Font and Background colour for Titles</p> <p>Show Labels: This checkbox allows you to show/hide category labels</p> <p>: This button accesses the dialog window where you can set the Background colour for category labels</p> <p>Note: Font style for a category label is set in the variable's Format.</p>
Borders	<p>External border applies to the Stub variable's title.</p> <p>Cell separator applies to the cell borders.</p>

	Borders settings are: thick, thin or no border, and you can set their colours as well.																																																																																		
Line Indent	Indentation of Titles and Labels in number of characters.																																																																																		
Display Empty Lines	This checkbox allows you to display empty rows. When not checked, empty rows are not displayed, and a table consisting of empty rows will not be displayed either.																																																																																		
Row Ranking	<p>You can sort rows in ascending or descending order. The sort is applied only within the same Chain at the last stage of a D (Distribution) or LO (List of Occurrences) Option.</p> <p>The sort is performed always on the values in the 1st column; this will be the counts, or means if applicable.</p> <p>When there is a Distribution, the level of indentation of category labels in the Format will be used as a further primary sort key.</p> <p><i>Example:</i></p> <table border="1"> <thead> <tr> <th rowspan="2"></th> <th colspan="4">Sex of Respondent</th> <th colspan="2">Is your age with</th> </tr> <tr> <th>Total</th> <th>Male</th> <th>Female</th> <th>Total</th> <th>16</th> </tr> </thead> <tbody> <tr> <td colspan="7">At which one of these department stores do you most often do your shopping</td> </tr> <tr> <td>Total</td> <td>118</td> <td>0</td> <td>14</td> <td>0</td> <td>25</td> <td>0</td> </tr> <tr> <td>Group 1</td> <td>21</td> <td>0</td> <td>5</td> <td>0</td> <td>10</td> <td>0</td> </tr> <tr> <td>Macy's</td> <td>9</td> <td>0</td> <td>3</td> <td>0</td> <td>2</td> <td>0</td> </tr> <tr> <td>JC Penny</td> <td>7</td> <td>0</td> <td>1</td> <td>0</td> <td>6</td> <td>0</td> </tr> <tr> <td>Bay</td> <td>5</td> <td>0</td> <td>1</td> <td>0</td> <td>2</td> <td>0</td> </tr> <tr> <td>Group 2</td> <td>19</td> <td>0</td> <td>4</td> <td>0</td> <td>7</td> <td>0</td> </tr> <tr> <td>Bloomingdales</td> <td>10</td> <td>0</td> <td>2</td> <td>0</td> <td>4</td> <td>0</td> </tr> <tr> <td>Sears</td> <td>5</td> <td>0</td> <td>2</td> <td>0</td> <td>3</td> <td>0</td> </tr> <tr> <td>Eaton</td> <td>4</td> <td>0</td> <td></td> <td></td> <td>4</td> <td>0</td> </tr> </tbody> </table>		Sex of Respondent				Is your age with		Total	Male	Female	Total	16	At which one of these department stores do you most often do your shopping							Total	118	0	14	0	25	0	Group 1	21	0	5	0	10	0	Macy's	9	0	3	0	2	0	JC Penny	7	0	1	0	6	0	Bay	5	0	1	0	2	0	Group 2	19	0	4	0	7	0	Bloomingdales	10	0	2	0	4	0	Sears	5	0	2	0	3	0	Eaton	4	0			4	0
	Sex of Respondent				Is your age with																																																																														
	Total	Male	Female	Total	16																																																																														
At which one of these department stores do you most often do your shopping																																																																																			
Total	118	0	14	0	25	0																																																																													
Group 1	21	0	5	0	10	0																																																																													
Macy's	9	0	3	0	2	0																																																																													
JC Penny	7	0	1	0	6	0																																																																													
Bay	5	0	1	0	2	0																																																																													
Group 2	19	0	4	0	7	0																																																																													
Bloomingdales	10	0	2	0	4	0																																																																													
Sears	5	0	2	0	3	0																																																																													
Eaton	4	0			4	0																																																																													

Banners

The settings for Banners are presented in the upper-right area of the window:

Setting	Description
1st and 2nd Stage Style	<p>Show Titles: Allows you to show/hide the variable titles.</p> <p>: For setting Font and Background colour for Titles</p> <p>Show Labels: Allow you to show/hide category labels</p> <p>: For setting the Background colour for category labels</p> <p>Note: Font style for a category label is set in the variable's Format.</p>
Borders	<p>External border applies to the Banner variable's title.</p> <p>Cell separator applies to the cell borders.</p>

	Borders can be thick, thin or no border, and you can set their colours as well.
Justify	Justification applies to Banner Titles and Labels as a whole. You can justify them both vertically and horizontally.
Display Empty Columns	This checkbox allows you to display empty Columns. When not checked, empty Columns are not displayed, and a table consisting of empty Columns will not be displayed either.
Column Ranking	<p>You can sort columns in ascending or descending order. The sort is applied only within the same Chain at the last stage of a D (Distribution) or LO (List of Occurrences) Option.</p> <p>The sort is performed always on the values in the 1st row; this will be the counts, or means if applicable.</p>

Counts

The settings for Counts are presented in the lower-left area of the window:

Setting	Description
Tests	<p>The choice is Chi2 Test or Proportion Tests.</p> <p>Test are performed on each table defined by the Stub and Banner variables used in the last stage. It is assumed that the last stage variables in the Stub and Banner are using a Template which includes the T (Total) Option, or by default the A (Respondents) Option i.e. that there is both a Total row and Total column.</p> <p>If the table contains both weighted and unweighted parts, tests are always conducted separately for each part. Hence, if the table presents a weighted distribution for a variable, there must be a weighted total row (specified with a T or A option in the Template).</p> <p>These tests make the assumption that the table does not include any M (mean), Mi (minimum), Ma (Maximum), NB (average number or responses) Options. In view of the above, a table can include tests for some parts, and exclude tests from other parts.</p> <p>Chi2 Test</p> <p>The Chi2 test enables you determine if the count in a cell is over or under-represented. A count which is over-represented is indicated by "+" signs, a count which is under-represented is indicated by "-" signs. Display of one, two, or three of these signs indicates a confidence level of 90%, 95% and 99% respectively.</p> <p>The Chi2 test assumes that both the Total row and Total column (corresponding to the T or, by default the A option in the Template) are either weighted or unweighted in the cell being tested.</p> <p>Proportions Test</p> <p>The Proportions test compares counts between each pair of columns (including the Total column if requested). Columns are labelled A, B, C etc; the Total column, if requested, is labelled Z. The column corresponding to the Template option A (respondents) is labelled Y. The test result for a cell is displayed, in the cell itself, by a listing of the labels of all the columns against which the test is positive. However, the result of the test is displayed only in the column with the highest count (except when comparison is against the Total column, in which case the test is displayed in the other column.)</p> <p>For any given column, comparison tests are made only with columns to the left of it in the display.</p>

The Proportions test assumes that there is a Total column (Template option T, or by default option A, if comparison with the Total is requested) and a Total row ((Template option T, or by default option A) which are homogeneous i.e. both weighted or unweighted for the cell being tested

- Notes:**
- When there is both a weighted and unweighted Total column, they will each nevertheless be labelled Z (or Y) since column testing is always made against the only Total column which is appropriate.
 - When the table presents a weighted distribution, if the test has to be made on the **unweighted** frequencies, then there must be both a weighted T and unweighted T option in the Template (or by default a weighted A and unweighted A option).

Confidence intervals.

This option provides confidence intervals in each cell of a table.

In the 'Table Settings' screen, the user is given the following options, in the 'Tests' area.

In the lower-left area, CIV and CIH provide confidence intervals for vertical and horizontal percentages.

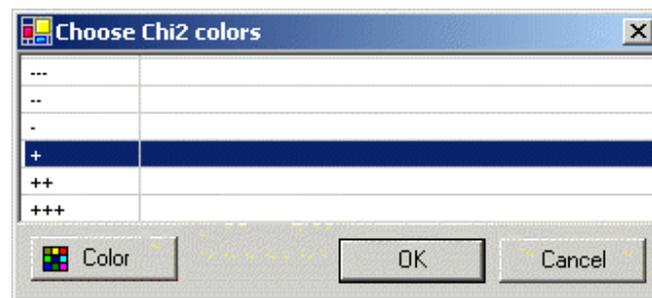
In the lower-right area, CI provides the confidence interval for means.

In both cases, the user must choose one of the options : 90, 95 or 99% (elsewhere, the 95% option is applied by default).

Background Colour

The Background Colour  button applies to cells with "counts".
Use Background colour for Chi2 checkbox (which applies if the Chi2 test is selected) allows you to use colours to indicate cell significance corresponding to +++, ++, ... as described above.

The  button to the right of the checkbox, allows you to select a colour for each significance level.



Cell layout

Note: These settings only apply to cells without Means.
 Each cell in a table has a choice of 9 available sub-cells (positions) for displaying Counts, Row Percentages, Column Percentages and Tests. You can "**Drag and Drop**" any item from one position to another.

%H	Count	Chi2
%V		

Visible Adjust to 100%

Once you have selected an item (by clicking on it), you can:

- **Make an item invisible.** When you un-check the Visible checkbox, a line will be drawn through the cell to indicate that its contents will not be displayed in the table (see above). If the item is the Chi2 test, this will suppress the signs +++,...but have no effect on any [background colours](#) that you may have specified.
- **Adjust to 100%.** This option allows you ensure that the displayed percentages add up to 100% exactly. To make the option appear, click on a %H or %V cell..

The option applies only to type D templates:

- For the %H percentages, the column variable at the last stage must be of type S or T.
- For the %V percentages, the row variable at the last stage must be of type S or T.

Notes:

- The % concerned must have no decimals, otherwise this option will be ignored.
- If the % is based on T, the base will consist of all simple codes in the variable's Format plus those in the N (No code) category, if it exists.
- If the % is based on A, the base will consist of all simple codes in the variable's Format.

The program will take the sum of the percentages described above. If the sum differs from 100, one or more percentages will be adjusted to make the sum add up to 100.

- **Set Font attributes for an item.** Double-click on the cell, or click on the Font button  to access the Font attributes dialog window. Attributes include Font, size, style, colour etc.

Display of Counts

Points to note concerning the display of counts:

- If the cell contains a C (Template) Option, the results will always be displayed without decimals, since the decimals setting in the Template only concerns percentages. Nevertheless, the decimal part will have been taken into account when performing the breakdown for this variable using the C Option. If you wish to display the result with decimals, you could achieve this by introducing another variable using arithmetic functions.
- When you make counts invisible, the positions in the table will only contain percentages. If there are positions for which no percentages are requested, the position should be empty. To avoid this situation, by convention, counts will be displayed in empty cells in place of the missing percentages. This enables you to have tables which present in the same column: a count representing the base total, and percentages in the category cells.

Threshold	When Proportion tests are requested, you can choose the threshold value: 90/95, 95/99, 90, 95 or 99%. Results will be shown in upper-case if the highest end of the range satisfied, and in lower-case if the lowest end of the range is satisfied. For single threshold values (90 for example), the letter will be shown in upper-case.
Display percent sign	When this setting is checked, each percentage will be followed by the percent sign %.
Character for empty cells	You can specify a character to be displayed in empty cells.

Means

The settings for Means are presented in the lower-right area of the window.

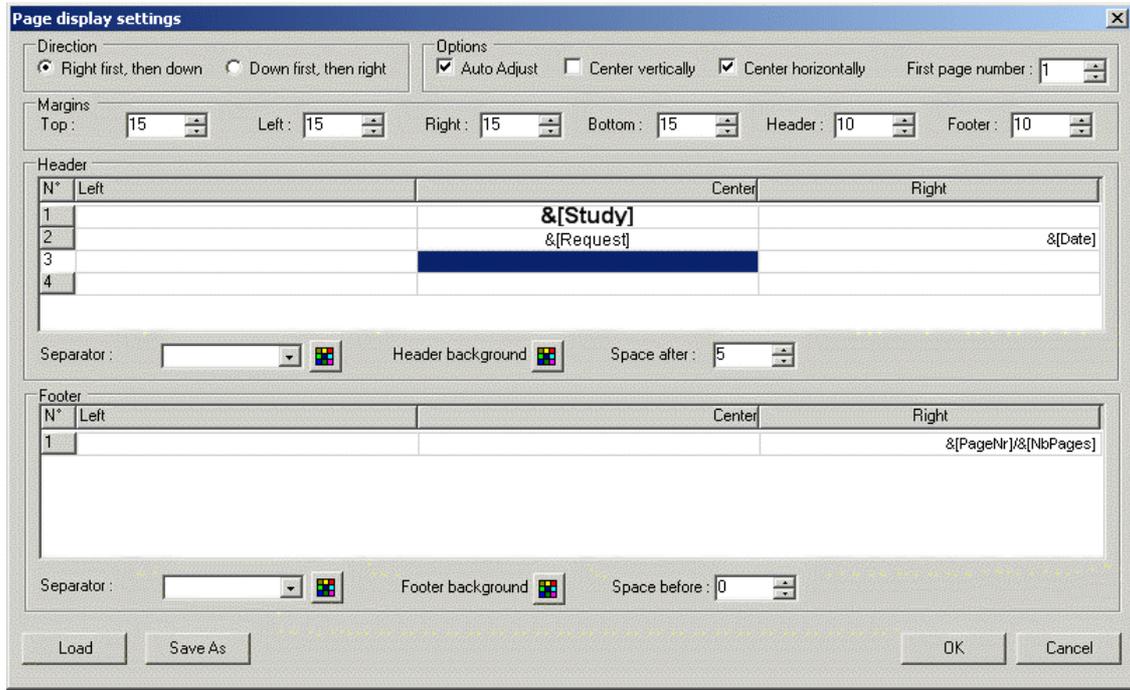
Setting	Description									
Means Tests	<p>If any one of the counts is less than 30, the Student's T test will be used.</p> <p>The test is performed by comparing the results in 2 columns. The columns are labelled A, B, C,... as in the Proportions tests, the Total column labelled Z, and the "respondents" column (Template Option A) labelled Y. For any given column, comparison tests are made only with columns to the left of it in the display. However, the result of the test is only displayed in the column with the highest mean (except when comparison is made against the Total column, in which case the test is displayed in the other column.)</p>									
Background Colour	<p>The Background Colour  button allows you to set the background colour for cells with "means". This enables you to make them easy to distinguish visually from cells with counts.</p>									
Cell layout	<p>Note: These settings apply only to cells with Means.</p> <p>Each cell in a table has a choice of 9 available sub-cells (positions) for displaying a Count, Mean, Standard Deviation, Standard Error and Tests. You can "Drag and Drop" any item from one position to another.</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; text-align: center;">Count</td> <td style="width: 33%; text-align: center;">Mean</td> <td style="width: 33%;"></td> </tr> <tr> <td style="text-align: center;">SD</td> <td style="text-align: center;">SE</td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table> <p><input checked="" type="checkbox"/> Visible </p> </div> <p>Once you have selected an item (by clicking on it), you can:</p> <ul style="list-style-type: none"> • Make an item invisible. When you un-check the Visible checkbox, a line will be drawn through the cell to indicate that its contents will not be displayed in the table (see above). • Set Font attributes for an item. Double-click on the cell, or click on the Font button  to access the Font attributes dialog window. Attributes include Font, size, style, colour etc. 	Count	Mean		SD	SE				
Count	Mean									
SD	SE									

Example of a table using Background colours

		PRODUIT										
		TOTAL	VIALEBEX 4% 100 ml		VIALEBEX 4% 250 ml		VIALEBEX 4% 500 ml		VIALEBEX 20% 50 ml		VIALEBEX 20% 100 ml	
Nombre d'unités par usage												
Cumul												
Nombre d'unités par usage												
Total	381,741 100.0	100	787 0.2	100	3,594 0.9	100	20,784 5.4	100	129 0.0	100	72,827 19.1	100
1 - Réa polyvalente-	41,039 100.0	11			1,483 3.6	41	1,697 4.1	8			8,349 20.3	11
2 - Réanimation médicale	26,860 100.0	7	12 0.0	2	205 0.8	6	1,305 4.9	6			5,604 21.0	8
3 - Réa chir. cardiaque -	28,009 100.0	7	2 0.0	0	102 0.4	3	834 3.0	4			1,795 6.4	2
4 - Réa chirurgicale -	31,305 100.0	8			42 0.1	1	3,792 12.1	18	1 0.0	1	7,481 23.9	10
5 - Urgences -	27,645 100.0	7	27 0.1	3	57 0.2	2	727 2.6	3	1 0.0	1	909 3.3	1
6 - Pédiatrie -	5,317 100.0	1	110 2.1	14	142 2.7	4	277 5.2	1	92 1.7	71	1,080 20.3	1
7 - Chir gén. et dig.-	79,161 100.0	21	1 0.0	0	479 0.6	13	5,069 6.4	24			8,494 10.7	12
8 - Chirurgie urologique	3,775 100.0	1			31 0.8	1	392 10.4	2	1 0.0	1	105 2.8	0
9 - Chirurgie thoracique-	19,419 100.0	5	8 0.0	1			804 4.1	4			412 2.1	1
10 - Autres chirurgie -	20,444 100.0	5			257 1.3	7	547 2.7	3	9 0.0	7	1,197 5.9	2
11 - Gastro-entérologie -	36,493 100.0	10	91 0.2	12	10 0.0	0	394 1.1	2			23,382 64.1	32
12 - Néphrologie -	13,049 100.0	3	3 0.0	0	163 1.2	5	1,282 9.8	6			2,826 21.7	4

Page Settings

The Page Settings window is accessed through its button  in the Toolbar. **Note:** Page Settings can be saved under a Stylesheet name which can be later Loaded into this window:

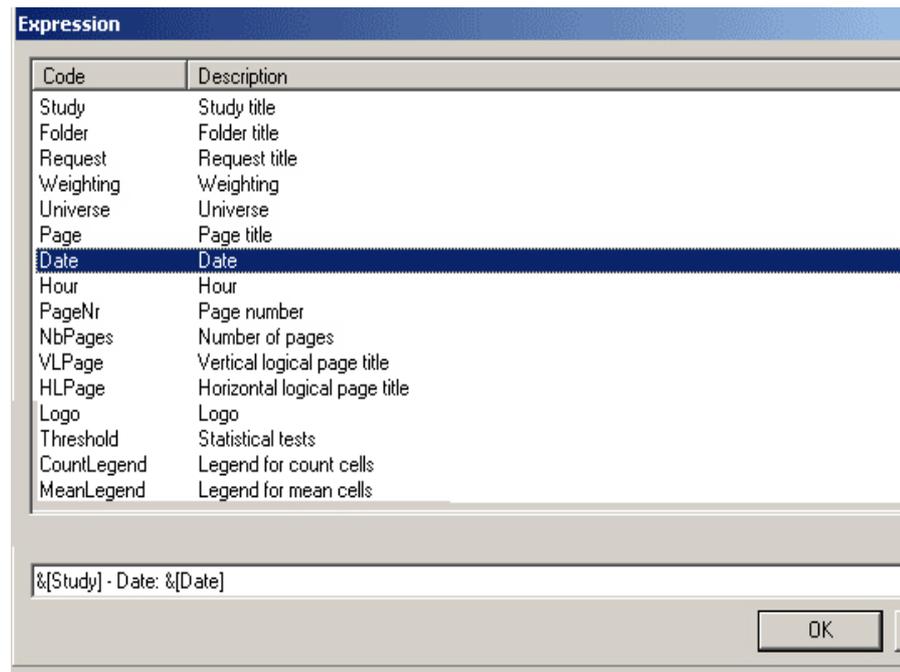


Note: In the **Tools>Table/Page Settings** menu, you can add, delete and edit Page Settings Styles.

Setting	Description
Right then down Down then right	This setting allows you to determine the order in which several pages will be output. It applies when a table is too large to fit on a page, or if Page Breaks have been specified in the Table Settings.
Auto Adjust	When a Table is too large to fit on one page, this setting instructs VoxcoTab to try and fit it on the page by reducing column widths (while maintaining readability of their contents, or course). If it is unable to achieve this, the original column widths will be maintained.
Center Vertically/Horizontally	These settings position the table within the page.
First Page Number	This first page number applies if page numbering is requested (see below).
Margins	These margins are specified in millimetres (1 inch=25 mm). You can set margins for Top, Left, Right and Bottom. The Header and Footer margins apply to the space before them???
Header (applies also to Footer)	The Header consists of one or more lines. You can insert or delete lines with a right-click in the line No. column, or by using the Insert/Delete keys.

A right-click in a line's No. column also enables you to determine the line Height. You can specify a Fixed height in millimetres or select the Auto option. It is strongly recommended that you use the Auto option (unless you have a special situation such as an image etc.). The Fixed option with a value of zero would make a line disappear, even if it contained information.

Each line consist of three parts (Left, Centre, Right). You can determine the content (text, variables, images) and format of each part. Press any key or double-click in the selected part to bring up the Expression window in which you can compose your text (in the field at the bottom of the window) including any of the listed variables (by double-clicking them). Variables are inserted in square brackets prefixed with the & character:



For example: **Study title** displays the title of the study, **Page title** displays the complete title of the variable(s) in the page together with the current categories, for example: Sex: Male.

The **VLPage** (Vertical logical page title) and **HLPPage** (Horizontal logical page title) refer to the titles defined for the Index (see Index).

The **Threshold** option displays the confidence level chosen for Tests.

The **Counts Legend** and **Means Legend** options display legends (grid) showing the positioning of items in the table cells. Example:

Count	%H	Sex of Res
%V		Total
Think about large d		
Total		

A right-click in a part allows you to:

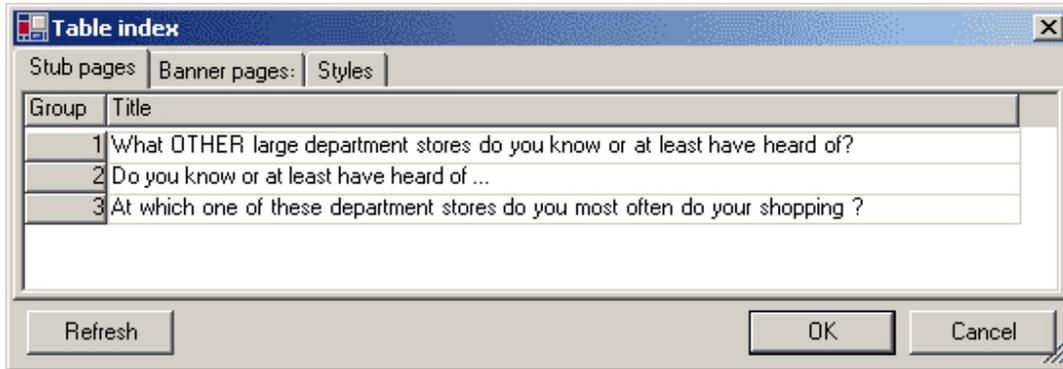
- **Clear** (Reset) the part
- Specify **Font** attributes

	<ul style="list-style-type: none"> • Set Horizontal and Vertical alignment of the contents • Insert an image (a logo for example) <p>Separator: Allows you to insert a line (thick or thin) after the header, and to set its colour.</p> <p>Header Background: Allow you to set a background colour for the header.</p> <p>Space After: Determines (in mm) a space after the Header (or before the Footer)</p>
Footer	(the above description of a Header equally applies to a Footer)
Load/Save buttons	Allow you to Load an existing Stylesheet or to save the current settings as a new Stylesheet under a user-defined name.

Page Index

The Table index is accessed through the Page Index button  in the Toolbar.

When a table is divided into several logical pages by means of the Page Break check-boxes in a Table Request, the title of each page that appears in the Index can be edited.



These titles are initialized with the name of the variables determining the page breaks, you can simply replace them with your own text.

Note: To be visible, these titles must be inserted in a table Header or Footer (see [Logical Page Titles](#) in [Page Settings](#)). This enables each logical page to bear its own title.

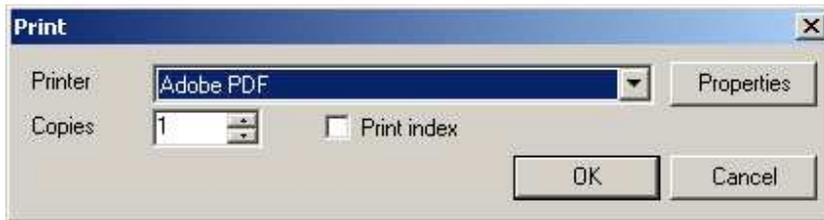
The Styles tab allows you to assign Styles (font, size, colour etc.) to the different titles (Study, folder, request, page, column, row.)

Printing tables

The Print dialog window is accessed through the Print button  in the Toolbar.

You can output the tables to a Printer or to PDF files first having used the Properties button to access the Page Set-up. Any Page Breaks set in the Table Request will generate page breaks *before* the output of their associated variables.

If the **Print Index** option is checked, a summary will be printed at the top of the table, providing on each logical page (obtained with the Break options) the titles included in the Index screen (see [Page Index](#)) together with the corresponding page number.



Note: The Portrait or Landscape orientation chosen in the Page Set-up will be conserved i.e. it does not have to be specified again each time that the table is printed.

Printing several requests

If you click on the Print button when you have several requests selected, the output will contain the tables for all the requests. The same applies when a request folder is selected; all the requests will be included.

Note: When you use the Print Index option, page numbering for the complete set of tables is sequential i.e. any First Page number specified in the Page Settings window will not apply.

Using PDF Creator

Standard PDF printing uses Acrobat whose execution times are very long, excessively so when hundreds of pages are involved. For this reason we recommend installing PDF Creator (free application supplied with VoxcoTab) which delivers very satisfactory execution times. Once installed, PDF Creator appears in the list of available printing devices.

Page Preview

The Page Preview  button in the Toolbar enables you to see the table on screen, formatted the same way as it will be printed.

The Printer dialog window (see above) allows you to select page size and also determine whether the page orientation is "portrait" or "landscape".

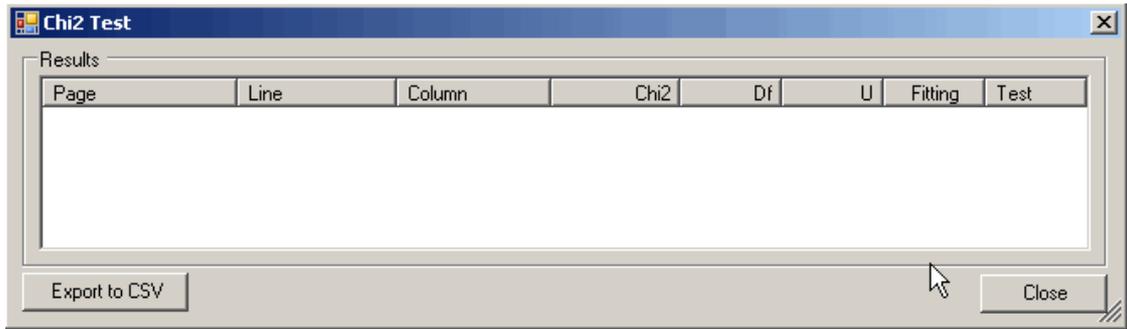
This button can apply to one or more requests or folders.

Gamma (Goodman et Kruskal) and Chi Squared tests.

These tests are available through a right click when previewing the table. The results are provided in a window, with the possibility of exporting them in CSV format.

The Gamma test is used when both the row and column variables are ordinal. The Chi Squared test is computed for the full table, while the already existing Chi2 test applies to each cell of the table.

The results for every table are: Chi2 value, Degrees of freedom, Value of U, Goodness of fit (+, ++ and +++ at 90, 95 and 99% levels), and an indication of the validity of the test.



Notes:

- The table should include a row and a column Total, (obtained with the template option T, or NT, or A), weighted or not as are the cells in the table.
- There should not be more than one D option for one table in a templates whether it be in the rows or columns.
- The format of variables in rows and columns should include individual values, and not just sub-totals: chi2 is calculated for the sub-table made only with these rows and columns, and all the rows and columns corresponding to the sub-totals are excluded from the calculation of chi2.
- In the case of several totals (weighted or not as are the cells in the table), they will be taken into account in the following order : first NT (arithmetic totals), next T, next A (respondents). Thus, in the case of multiple variables, the user has the choice between a Chi2 computed on the arithmetic totals of the table (when using the NT option), or on the logical totals (number of respondents, including or not the 'No Answers' (T and A options).

Exporting tables

The Export tables feature is accessed through the Export button  in the Toolbar.

The format options are XLS for Excel, CSV, or HTML. The option can be made for a request or a folder. With all options the complete table set is obtained, including all the displayed values (counts, percentages, etc.) together with titles and category labels. Table Settings (hiding empty rows and columns, hiding percentages etc.) are implemented.

The XLS (or XLSX when more than 256 columns are required) **option** produces an MS-Excel-readable file, which can contain several sheets, reflecting the page-layout settings (bold text, colours, font sizes etc.):

- If the table has a Page variable, there will be as many Excel sheets as there are pages. Furthermore, a separate sheet is created wherever a Page Break is requested in the Table Request.
- The Index is always supplied in the form of an additional tab titled 'Index'; it provides a list of the other tabs.

The CSV option produces a single sheet. It contains, in addition to Titles and Labels (in the 2nd column), two additional columns:

- The 1st column contains a code 0-3 indicating whether the Row corresponds to a Title (code 0 for 1st Stage Title, code 2 for 2nd Stage Title) or to Results (code 1 for 1st Stage Results, code 3 for 2nd Stage Results).
- The 3rd column contains a code 0-2 indicating which line of sub-cells corresponds to the presented results.

The HTML option can be in the form of a mini-site, or as a single file. In mini-site form, the folder will contain a set of files readable by Internet Explorer; the files separated into pages conforming to page breaks specified for the table. The legend appears to the left taken from the contents of the Table index screen obtained with Index icon.

Table Requests at the Folder Level

The Table Request buttons in the toolbar are also available at the Folder Level.

Button	Description
Execute	All the requests in the folder will be executed automatically
Table Settings	Enables table settings to be defined applying to all requests in the folder
Page Settings	Enables the page settings to be defined applying to all requests in the folder
Print Preview	
Print	All tables in the folder will be printed as a single document
Export	All tables in the folder will be exported into a single file

Row and Column Operations

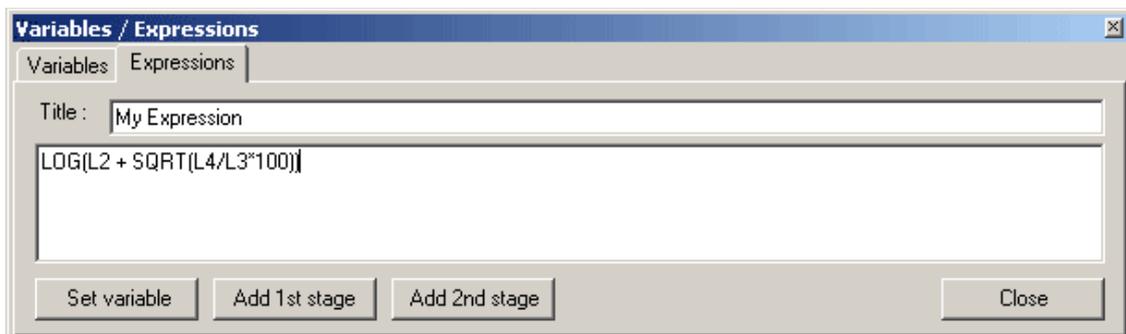
In a Table Request, you can create new Columns or Rows based on arithmetic operations on other Columns or Rows.

Table Requests

In the Table Request window, when you click on the button to the right of a Variable field (whether it be for a Stub or Banner), the Variables/Expression window appears; it consists of two tabs: Variables, Expressions. Instead of selecting a variable in its Variables tab, you can use the Expressions tab to define the "variable" in term of other Rows or Columns, depending on whether you defining a Row or Column respectively.

Using this feature, you can define a Column as the sum of several other Columns, or the relationship of two columns etc. These new computed Rows and/or Columns must be created before executing the Table Request, even through the computation formula can be modified afterwards.

The Expression tab contains a Title field and an Expression field where you can type your formula/expression:



The Title will be displayed in tables at the same level as Variable titles.

The Expression is an arithmetical formula (with optional parentheses) involving:

- Existing line numbers, in the form Lnn (where nn is the number in the first column of the Table Request), whether the Expression is being defined for Rows or Columns in the table.

Notes:

- The referenced lines must contain Variables, Expressions.
 - If the operation is at the 1st Stage, it must not be followed by a 2nd Stage, and the referenced lines must also be at the 1st Stage.
 - If the operation is at the 2nd Stage, the referenced lines must be at the 2nd Stage in the same group.
 - Referenced lines must correspond to variables whose attached Template comprises a single Option. And this Option must be neither D (Distribution) nor LO (List of Occurrences).
 - Care must be taken when deleting, inserting or moving lines. When doing so, it will be necessary to modify the referenced line numbers (this does not apply when specifying Bases for Percentages).
 - It is possible to reference not just a complete Row or Column, but a single element whose position, as it will appear when the table is displayed on the screen, is known. For example: L3[1] references the element in the 1st column of the resulting table (when operating on Rows) or the 1st Line (when operating on Columns). It must be noted that this index does not take into account Rows or Columns which are blank or whose display is suppressed.
- Integer or real number constants
 - Arithmetical operators: + - * /

- Functions:

ABS	Absolute Value
SQRL	Square Root
ROUND	Rounded
TRUNC	Truncate
RANDOM	Random number between 0 and 1
EXP	Exponent
LOG	Natural Logarithm
ISP	Returns 1 if the argument is positive or Null, 0 otherwise.
ISZ	Returns 1 if the argument is equal to zero, 0 otherwise.
ISSP	Returns 1 if the argument is Positive, 0 otherwise.

Note: Before closing the Expression window, the expression must be confirmed with the Set or Add button.

Assigned Template

The Expression must be assigned a Template whose function is to provide: a Title in the table display (in the line corresponding to variable labels), the number of decimals, and the Hide attribute. Any other Options and Attributes in the Template will be ignored. This Template must consist of a single line.

Execution

Execution is performed after the breakdown, and after means and percentages have been computed. This being the case, you can modify the computation expression after you have taken a look at the resulting table.

Operations are based on counts (typical case), sums (C Option in the Template) or Means (M Options in the Template) in the referenced argument element.

When the formula is applied, the result will be Null if:

- A computation is impossible (division by zero etc.)
- One of the operands is empty.

Rows are computed first, and then Columns.

Export

Export

All or a sub-set of a study's variables can be exported to files in different formats:

- VoxcoTab format
- SPSS format (SAV)
- COSI format
- SPAD format (SDA)

VoxcoTab format comprises 3 files:

- A Data file bearing a .ASC extension, in either fixed column or delimited format.
- A Study file bearing a .SDB extension, i.e. a new VoxcoTab study.
- A file bearing a .MAP extension, comprising a list of the variables and the positions of their data in the data file.

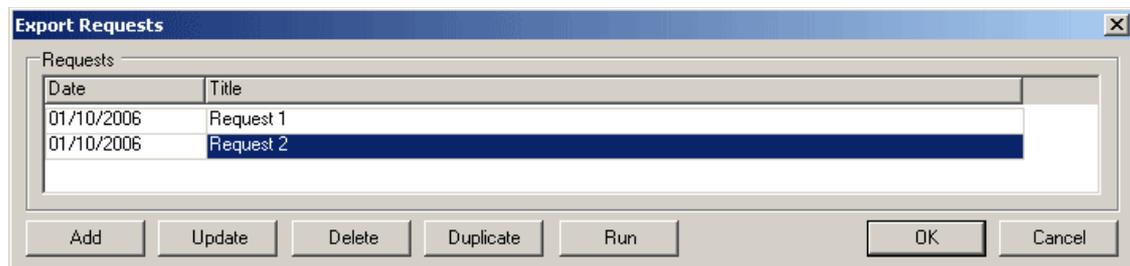
Hence the user is provided with a complete VoxcoTab study, ready for immediate processing with the VoxcoTab application.

COSI format comprises 2 files:

- A Data file bearing a .ASC extensions in fixed column format.
- A file bearing a .SCT extension i.e. the COSI script (variables dictionary).

An Export Request

The Export Requests window is accessed through the **Utilities>Export** menu:



You can Add, Update, Delete, Duplicate and Run a Request.

When you access a Request, the following window appears:



Title: Any title you wish to give to the Request

Universe: This can be used to specify a logical variable to be used as a Filter. Only cases which are TRUE for this variable will be exported.

VoxcoTab, SPSS or COSI tabs: These are used to bring the appropriate specification window into view.

Export to VoxcoTab

You must provide:

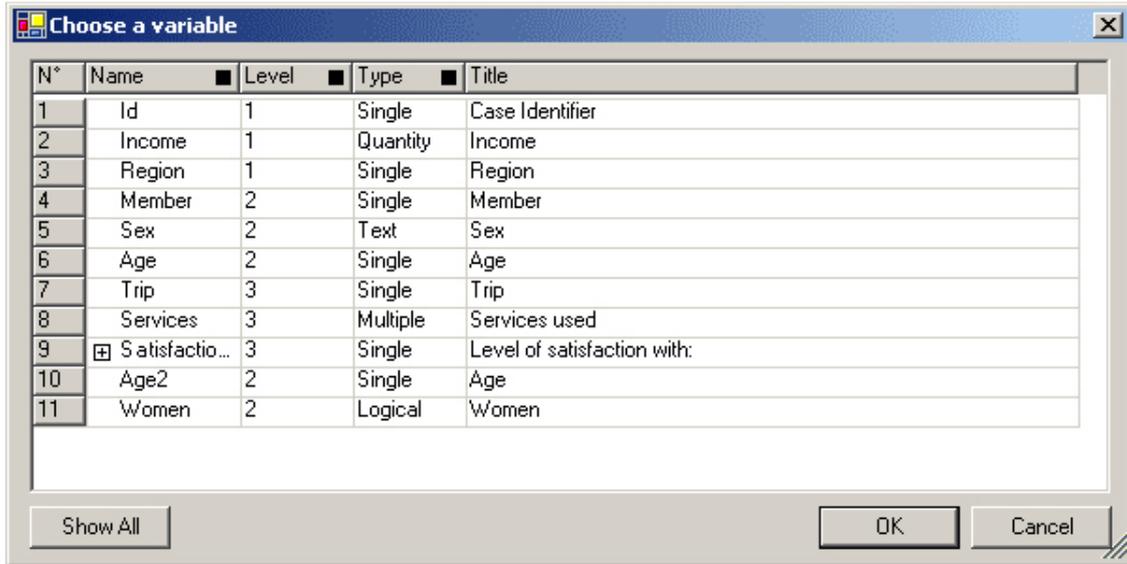
- The path to the folder in which the study will be created (the filename must not be specified)
- The Study Number of the study to be created
- The type of File Format for the data file: ASCII or CSV. For a CSV file, you must specify the Field Separator and Text Delimiter.
- If you select the Code to Label option, variables with Formats will have their category labels rather than category codes exported.

N°	Level	Variable	Lg	Dec	By Values	Nb Values	NA
1	1	Id	2				
2	1	Income	2				
3	1	Region	2				
4	2	Member	2				
5	2	Sex	2				
6	2	Age	2				
7	3	Trip	2				
8	3	Services	2		<input checked="" type="checkbox"/>		
9	3	Satisfaction[3]	2				
10	2	Age2	2				

The files created will bear the following names (where nnn is the specified Study number):

- E_nnn.asc
- E-nnn.sdb
- E-nnn.map

To choose which variables to export, click on the button to the right of the Variable field; the Choose a Variable window will appear, allowing you to select one or more variables:



Once you have selected your variables, for each variable, specify:

- Field length (for ASCII files only; for CSV files it will be ignored)
- Number of Decimals.
- For multiple response variables: The By Values option. If checked the data will consist of a series of codes in successive fields). If unchecked (a series of 0 or 1 values indicating the corresponding codes).
- For multiple response variables: The number of values to be considered.
- The value which is to indicate MISSING (blank by default).
- Use of UTF8 characters or not allowing Text variables containing Chinese or other exotic characters.

In Export Requests (VoxcoTab, SPSS, COSI and SPAD), you can move an already selected variable : select the variable and press the CTRL + Up/Down arrow on your keyboard.

For **Dimensioned variables**, the occurrences are automatically written one after the other.

Autosize option. When you click on the Autosize button, all the properties described above (field length, number of decimals etc.) will be automatically set by the program; it does this by reading the data file to establish the appropriate settings. The only property which you might need to indicate concerns the By Values or By Rank option for multiple response variables. You can change these values if desired.

For **Studies with Levels**, the selected variables can belong to different levels. For this, the MultiLevel option checkbox is provided:

- When the Multi-Level option is **not checked**, only cases belonging to one (and only one) level will be exported, the level determined by the highest Level number present in the selected variables. Selected variables with a lower Level number are also be exported in the same record, and at the same level.
- When the Multi-Level option is **checked**, the new study created will be Multi-Level, and the ASCII file created will comprise multiple records (one per level).

Export to SPSS

You must provide the path to the folder in which the study will be created (the filename must not be specified)

Variables are selected in the same way as for Export to VoxcoTab together with the same attributes.

In Export Requests (VoxcoTab, SPSS, COSI and SPAD), you can move an already selected variable : select the variable and press the CTRL + Up/Down arrow on your keyboard.

Export Request

Title: Request 1

Universe: []

VoxcoTab SPSS

Path: []

Variables

N°	Level	Variable	Lg	Dec	By Values	Nb Values	NA
1	1	Id	2				
2	1	Income	2				
3	1	Region	2				
4	2	Member	2				
5	2	Sex	2				
6	2	Age	2				
7	3	Trip	2				
8	3	Services	2		<input checked="" type="checkbox"/>		
9	3	Satisfaction[3]	2				
10	2	Age2	2				

Run [] [OK] [Cancel]

Notes:

- Multiple variables are split into as many variables as the number of values specified.
- Dimensioned variables create as many variables as there are occurrences.
- The Autosize option works the same as in Export to VoxcoTab.

Export to SPAD

The procedure is the same as for SPSS. Use the SDA extension for the file to be created.

Export to COSI

You must provide the name and path to the script file (.SCT) and the data file (.ASC).

Export Request

Title: First_Request

Universe: Men

VoxcoTab SPSS Cosi

SCT File: C:\VoxcoTab\EXP01.sct

Data File: C:\VoxcoTab\EXP01.asc

Variables

N°	Variable	Lg	Dec	By Values	Nb Values	NA
1	SHOP	1				
2	SEX	1				
3	STATE	2				
4	Q1A	1				
5	Q2A	1				
6	q2[6]	2				
7	Q3A1	1		<input checked="" type="checkbox"/>	99	
8	Q3B1	1				

Run Autosize OK Cancel

Variables are selected in the same way as for Export to VoxcoTab together with the same attributes (field length, number of decimals etc.). The Autosize option works the same as in Export to VoxcoTab.

Weighting

Weighting

Overview

This module computes a weight or adjustment factor for each case in the sample given one or more weighting criteria. The weights are computed so that the structure of the weighted sample conforms to a given theoretical structure.

The computation procedure is iterative.

Weighting adjustment can apply either to numbers of cases or quantities associated with them.

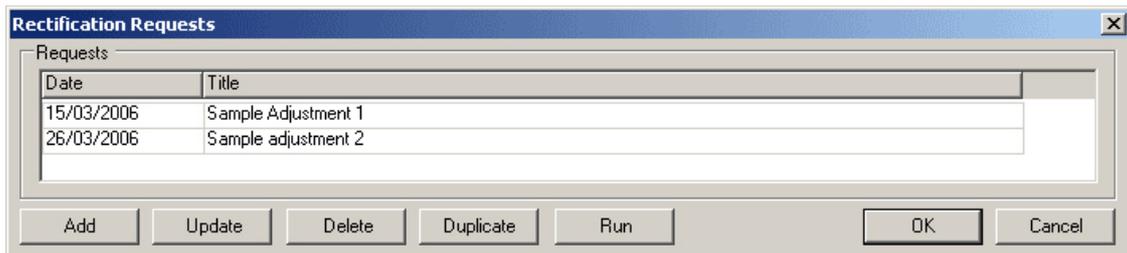
The results are provided in the form of a file containing a weight for each case that has been processed. The file must then be declared as an [External Table](#) for the study, so that the weights can then be accessed.

The necessary elements are the following variables:

- Weighting variable (Quantity type) which will store the adjustment weights.
- Identification variable (Text type) which uniquely identifies each case in the output file.
- Weighting criteria (Single type) which are processed individually or in pairs (crossed criteria).

Weighting request

The Weighting Request window is accessed through the **Utilities>Weighting Factor** menu:



You can Add, Update, Delete, Duplicate and Run weighting requests.

A request is configured in the Weighting Factor Request window shown below. The **Title** field is for any text string to describe the request. The **Options tab** is for providing general options, while the **Criteria tab** is for specifying the adjustment criteria and their objectives:

Options (tab)

Setting	Description
ID	(Obligatory) Text-type variable for identifying cases in the weights output file
Base	(Optional) Logical-type variable describing the Universe (Filter) on which the weighting is to be established
Weight	(Obligatory) Quantity-type variable in which the weight for each case will be stored
Initial weight	(Optional) Quantity-type variable. Values in this variable will be used as initial weights to start the iterative computation process. Note: This feature was introduced to satisfy a client request, however we do not recommend that you use it.
Quantity to adjust	(Optional) Quantity-type variable. If used, the weighting will apply not to numbers of cases (or percentages of them) but to volumes (or percentages of these volumes). For example, in an agricultural study dealing with land usage (acres) for wheat cultivation, we might want to weight on the basis of wheat production (tons) rather than the acres used.
Max iterations	The maximum number of iterations (300 by default). It is recommended that you leave this value unchanged, except in special cases.
Threshold	Difference between targeted values and those obtained during the process below which the iterative process will be terminated. It is recommended that you use the default value (0.1).
Weight is null if less than	Weights less than this value will be replaced by zero.

Null weight is considered	Null weights can be converted to zero or MISSING.
Weight range	Weights will be limited to the Maximum and Minimum callues specified.
Target in	Targets specified for each of the criteria can be specified in either values or percentages.
Extrapolate to	<p>If this option is not selected, the weighting will produce a total count equal to the unweighted sample. If selected, you specify the total number of cases (or total quantity if applicable) to be obtained after weighting. Note: You supply the total number of cases even if Target is specified in percentages (see the previous option)</p> <p>This option is obligatory if:</p> <p>Note: This option is obligatory if there is a quantity to be adjusted. However, it is not necessary to check this option if objectives are provided in values.</p>

Studies with Levels: Weighting applies to a single level, by convention the level of the ID variable. Consequently, the Weighting variable must be at the same level; the same applies to the Initial Weight and Quantity variables if they exist. The criteria must be at the same or higher level (i.e. the same or smaller level number).

Criteria (tab)

The References field enables you to select one or more criteria, whether single or crossed. These criteria must be **Single-type undimensioned variables** or **occurrences or a dimensioned variable**, and they must have a **Format categories**. Targets must be supplied for each and every single-code category i.e. any categories grouping together several codes will be ignored.

Weighting factor request

Title: Sample Adjustment 1

Options Criteria

References: AGER7

...	-
13 - 17 years old	1000000
18 - 24 years old	1500000
25 - 34 years old	2900000
35 - 44 years old	4000000
45 - 54 years old	300000
55 - 64 years old	200000
65 and over	100000
NA	0
Total	10000000

Keep NA values

Run OK Cancel

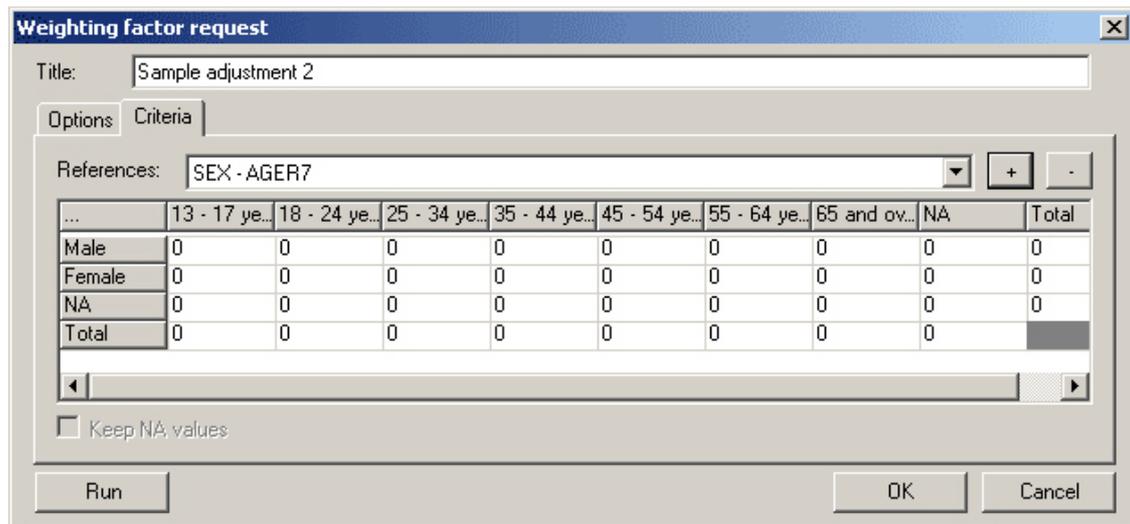
Targets must be supplied for all of the displayed categories, including the value MISSING???. You can use copy and paste functions in the Target column.

Important note: By convention, the category NA here represents all cases which are not classified into the Format's category list. It includes, therefore, the true MISSING for the variable (as it appears in tables), together with all values for the variable which are not encompassed by the Format categories. Hence, the NA considered here is not the same as that which appears in tables under the N (No Response) Option in Templates.

Keep NA values check-box: With this option selected, no target is supplied for NA category (it no longer even appears in the list). The weighting module will compute an adjustment weight which will maintain the same proportion of NA cases as in the unweighted sample.

Crossed criteria

In this case, a target must be supplied in each cell of the table produced by crossing the 2 selected variables:



Note: When using crossed-criteria, the Keep NA values option may no longer be checked.

Execution

The execution produces 2 files (nnn : study number, dddddd : request title):

- Weights file: Wnnn-dddddd.txt
- Report file: Wnnn-dddddd.log

The **Weights file** contains a line for every case belonging to the chosen Universe (Filter) consisting of the case ID together with its weight. This file must be set up as an [External CSV Table](#), with the ID variable used as index key linked to the first field in the table. The weight must read in by means of the ImportsCSV function in the second field.

This file will not contain any case whose ID or initial weight is MISSING.

The Log file reports on the execution, it lists the options and settings, and, for each criterion, the requested value, initial values in the sample, and adjusted values.

The user is also provided a result in the *log* file : **efficiency**.

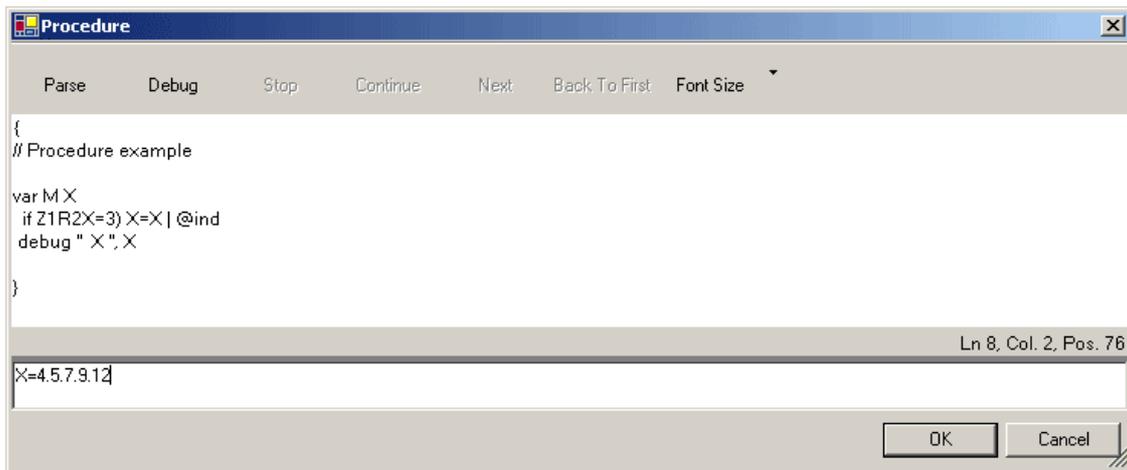
$$\text{Efficiency} = 100 * (\sum \text{weight})^2 / n \sum \text{weight}^2$$

Procedures

Overview (Procedures)

Description

A procedure is a custom [CPL function](#) which you define using the CPL (Cotab Procedural Language) language. This language allows you to perform all sorts of arithmetical, logical, and test operations including iterative loops. You define a procedure in the Variables window by choosing a CPL function in the Function column. You can import a procedure from another study by importing the variable to which it is assigned.



Creating a procedure

The Procedure definition window is divided into two panes: the upper pane is where you write the procedure itself.

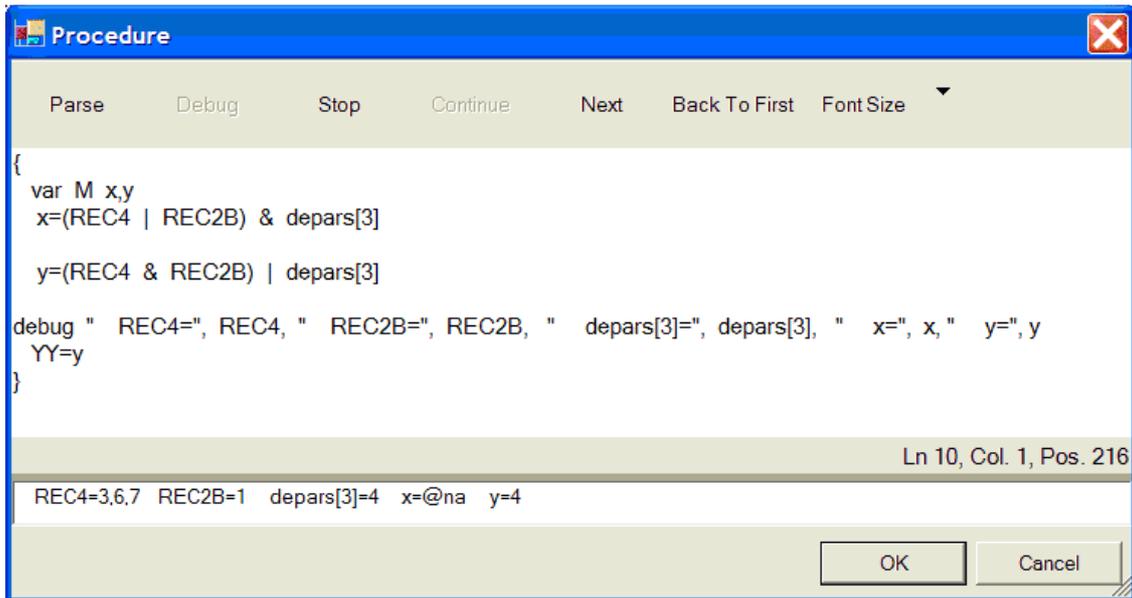
You type the procedure in free format making use of the standard text editing key combinations: Ctrl C (copy), Ctrl V (paste), Ctrl Z (undo), Ctrl A (select all), Insert, Delete, Enter.

The "Parse" button tests the syntax of the procedure.

The "Debug" button switches to Test Mode: when you click on the "Next" button the lower pane displays the result of the "Debug" instructions applied to the first (next) case in the data file.

The "Back to First" button returns to the first case in the data file.

The "Stop" button switches back to Edit mode in which you can continue modifying the procedure.



CPL Language

Symbols and notation

Definitions

CPL language has a relatively simple syntax Available symbols in CPL are:

- System symbols
- Keywords
- Study variables
- Local variables
- Constants

Note: Variable names are case-sensitive (upper and lower-case must be respected). However, system symbols and keywords are not case-sensitive.

Symbol	Description
System Symbols	System symbols are identified by their first character which must be "@". This is followed by one or more characters, for example: @begin, @end, @na, @ind, @write, etc.
Keywords	These reserved keywords consist of one or more characters: IF, FOR, CONTINUE.
Study Variables	These are variables defined in the Variables list (dictionary), and which must precede the procedure in the list.
Local Variables	<p>These variables must be declared before they are referenced (using an instruction such as var S a,b). A local variable can be declared anywhere in the procedure. The first character in the name must be a letter, the other characters are alphanumeric (the "_" (underbar) character is permitted, however the name must not be a CPL reserved keyword.</p> <p>Note that there are several differences between Local and Study variables:</p> <ul style="list-style-type: none"> • At the start of execution, local variables are initialized with the value of 0 for types S(single) or Q(quantity), MISSING for type M(multiple), "" for type T(text), and FALSE for types L(logical). Study variables, on the other hand, are initialized as MISSING for each new case. • The value of a local variable is not modified by VoxcoTab during execution of the study.
Constants	Numerical constants may be of type S (Single) or Q (Quantity); CPL automatically recognizes constant types. <i>For example:</i> 123 is type S, while 123.0 or 123. are type Q. Literal constants must be enclosed within quotes ""; and if the " character forms part of the literal constant, it must be preceded by the character ^. <i>For example:</i> ^"OK^"

System symbols

Symbol	Description
--------	-------------

@begin and @end	Section markers indicating the start of a beginning and ending section respectively.
@na	MISSING constant
@write	Indicates the file (name and directory) to which messages obtained with the write function are to be written. <i>Example:</i> @write filename " c:\\xxx\\test.log" "filename" is the name, internal to the procedure, to which write instructions make reference. You can define several files to which the procedure may write.
@ind	A S(ingle) type system variable which returns the occurrence index in the implicit loop or a dimensioned instruction. For example, to initialize a dimensioned variable STAB, the 2 following instructions are equivalent: STAB = @ind for (i =0 ; i<=dim ; i++) {STAB[i] = i}
Comments	Two types of comment indicators are supported: // makes all characters that follow it to the end of the line into a comment. /* */ delimiters enable several lines in a procedure to be ignored.

Variable types and implicit conversion

Types of variable

Both study variables and local variables can be of the following types:

M	Multiple response
S	Single response (numeric integer)
Q	Quantity (numeric with decimals)
T	Text
L	Logical

Local variables, whose type must always be specified, are declared as follows:

```
var M m1, m2, m3[200], m4[10]
```

Whenever a variable or constant is used in an instruction, CPL automatically performs an implicit conversion if necessary. The conversion does not destroy the variable's type, nor its original value. Note that an instruction requiring automatic conversion is not as fast as an instruction that does not require it.

Rules for conversion between types

Source type	Destination type	Description/Example
Multiple	Text	<i>Example:</i> 2,3,5 becomes "2,3,5"

Quantity	Single	The value is rounded. <i>Example:</i> 2.34 becomes 2, 3.5 becomes 4.
Single	Multiple	If the value is ≥ 0 the value is assigned to M, otherwise M is set to MISSING
Quantity	Multiple	The conversion is executed in 2 steps: Q is converted to S , then S is converted to M (see above)
Multiple	Single or Quantity	If the variable M is MISSING the result is MISSING otherwise the minimum value is taken
Single	Text	2005 becomes "2005 "
Quantity	Text	2.34 => "2.34"
MISSING		

and MISSING => "" or " " ???

Thanks to automatic type conversion, you can write the following instructions in CPL:

```
var M m
var T text
var S i
var Q q
m = "2,4,10" // T => M
q = "2.34" // T => Q
text = m // M => T
i = q // Q => S
i = m // M => S
```

Structure of a CPL procedure

A procedure is a user-defined routine which VoxcoTab calls when a study is being executed.

It can comprise any one or both of 2 section types, each of which is must be enclosed within brackets {}, together with variable declarations which can appear anywhere in the procedure:

1. **Beginning section.** This section is executed only once at the beginning of the study
2. **Body section.** This section is executed for each case in the study

Here is an example of a simple procedure containing both section types; each one is explained by means of comments:

```
var Q cumul_POIDS // initializes the internal variable cumul_POIDS to 0.0
@write outfile "C:\\RESULTS\\RES.TXT" //assigns internal filename referencing
```

```

// an external file
@begin // This marks the start of a begin section which will be
executed once only at the beginning of the study
{
  var S Total // initializes it to 0.0
  // The instructions in this section are executed once at the start of
study
  // execution, and before the data file is read.
  // Called by the main VoxcoTab engine, in the order that they appear
in the
  // dictionary.
  Total = 0
}
{
  // This is the body section. The instructions in this section are
executed for each case
  // at the level of the procedure.
  // Called by the main VoxcoTab engine, in the order that they appear
in the
  // dictionary.
  Total++
  cumul_POIDS += POIDS
}

```

Each of th 2 sections is optional. Each one, if present, must have at least one instruction other than a declaration of variables.

The exit instruction is not allowed within a @begin section.

Study variables are not accessible within a @begin section i.e. they are only accessible in the body section.

Local variables (specific to the procedure) can be declared anywhere, within or outside a section.

Basic CPL instructions

CPL (Cotab procedural language) is a free-form structural language.

Instructions are usually terminated by one or more carriage-returns, but some expressions can be written on several lines; in particular certain characters such as ")", "=", etc. can be followed by a carriage-return in the middle of an instruction since CPL knows that the instruction is not yet complete.

Dimension of instructions (CPL)

The notion of dimensions for instructions is a very important feature of the CPL language. It enables an instruction or block of instructions (with a branching structure, described below) to be carried out occurrence by occurrence on dimensioned variables (possessing the same dimension or no dimension). This is a natural generalization of the execution of Cotab functions when applied to dimensioned variables.

General Rules

The dimension of a basic instruction is determined by that of the variables involved in the instruction. Variables must all be of the same dimension or have no dimension.

The CPL engine executes a dimensioned instruction with an implicit loop whose range is 1 to the dimension of the instruction.

The system variable @ind can be used in a dimensioned instruction; it takes on the value of the occurrence index within the implied loop (@ind is always 0 in the case of an instruction that has no dimension).

For example, take a dimensioned variable CNBHAB and a local un-dimensioned variable cumul. The following 2 instructions achieve the same result:

- cumul += CNBHAB // the instruction has dimension = CNBHAB.dim
- for (i=1; i<=CNBHAB.dim; i++)
{
 cumul += CNBHAB[i] // un-dimensioned instruction
}

Accessing an occurrence of a dimensioned variable

For a dimensioned variable named var_tab, the expression var_tab[arith_expr] enables you to obtain the occurrence of var_tab whose index is returned by result of arith_expr.

Example : var_tab[3], var_tab[i*3+j], var_tab[(i+j)*3+k], etc.

Increasing/decreasing a value

There is often the need to increase or decrease the value of a numeric variable (type S or Q). CPS provides two instructions for the purpose:

variable++ // the variable is increased by 1

variable-- // the variable is decreased by 1

Assignment instructions

variable = expression

expression can be:

- (expression)
- Numeric_Expression
- Logical_Expression
- Text variable or constant

A Numeric_Expression can be:

- Numeric_Expression + Numeric_Expression // addition
- Numeric_Expression - Numeric_Expression // subtraction
- Numeric_Expression * Numeric_Expression // multiplication
- Numeric_Expression / Numeric_Expression // division
- Numeric_Expression % Numeric_Expression // modulo
- Numeric_Expression ** Numeric_Expression // power
- Math_Function (Numeric_Expression)
- Min(argument1[, argument2, ...]) // argument must be un-dimensioned.
- Max(argument1[, argument2, ...]) // argument must be un-dimensioned.
- NbRep(**M**ultiple_Variable)
- Length(**T**ext_Variable)
- Numeric_Argument (variable or constant of type S or Q)

where Math_Function can be one of the following mathematical functions:

- ABS : absolute value
- SQRT : square root
- ROUND : rounding
- TRUNC : truncation
- RANDOM : random number between 0 and 1 (there is no argument with this function)
- EXP : exponent
- LOG : naperian logarithm
- ISP : returns 1 if the argument if positive or null, otherwise 0
- ISZ : returns 1 if the argument equals 0, otherwise 0
- ISSP : returns 1 if the argument is positive, otherwise 0

Logical_Expression can be

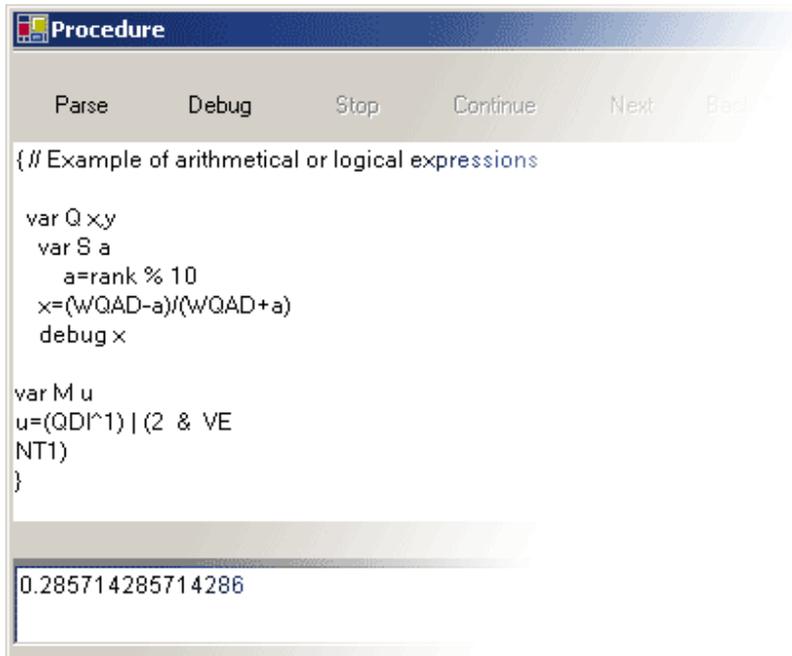
- Logical_Expression & Logical_Expression // AND
- Logical_Expression | Logical_Expression // OR
- Logical_Expression ~ Logical_Expression // EXCEPT
- Logical_Expression ^ Logical_Expression // XOR

Priority between arithmetic operators is:

** then *, /, % then +, -.

There is no priority between logical operators.

Example



The screenshot shows a debugger window titled "Procedure" with a menu bar containing "Parse", "Debug", "Stop", "Continue", "Next", and "Back". The main area displays MATLAB code:

```
{// Example of arithmetical or logical expressions  
  
var Q x y  
var S a  
    a=rank % 10  
    x=(WQAD-a)/(WQAD+a)  
    debug x  
  
var M u  
u=(QDI^1) | (2 & VE  
NT1)  
}
```

At the bottom of the window, the output of the debug statement is shown as the number 0.285714285714286.

Other forms of assignment (shortform)

Single/Quantity_Variable += Numeric_Expression // Single/Quantity_Variable = Single/Quantity_Variable + Numeric_Expression

Single/Quantity_Variable -= Numeric_Expression // Single/Quantity_Variable = Single/Quantity_Variable - Numeric_Expression

Single/Quantity_Variable *= Numeric_Expression // Single/Quantity_Variable = Single/Quantity_Variable * Numeric_Expression

Single/Quantity_Variable /= Numeric_Expression // Single/Quantity_Variable = Single/Quantity_Variable / Numeric_Expression

Single/Quantity_Variable %= Numeric_Expression // Single/Quantity_Variable = Single/Quantity_Variable % Numeric_Expression

Single/Quantity_Variable **= Numeric_Expression // Single/Quantity_Variable = Single/Quantity_Variable ** Numeric_Expression

Multiple_Variable &= expr_log // Multiple_Variable = Multiple_Variable & expr_log

Multiple_Variable |= Logical_Expression // Multiple_Variable = Multiple_Variable | Logical_Expression

Multiple_Variable ~= Logical_Expression // Multiple_Variable = Multiple_Variable ~ Logical_Expression

Multiple_Variable ^= Logical_Expression // Multiple_Variable = Multiple_Variable ^ Logical_Expression

Other functions

Concatenation of character strings:

variable = Chain(CHN1,CHN2,...)

CHN1 is either a text variable or a text constant(« »)

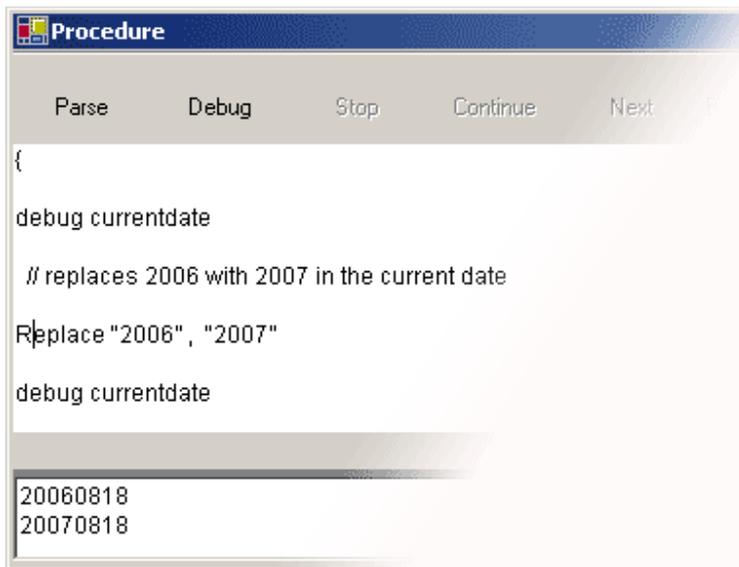
Replacement of the first appearance of a string (1st argument) by another string (2nd argument) in a Text variable (3rd argument).

Replace STRING_FIND , STRING_REPLACE , STRING_RESULT

Replacement of all appearances of a string:

ReplaceAll STRING_FIND , STRING_REPLACE , STRING_RESUL

Example



```
Procedure
Parse Debug Stop Continue Next
{
debug currentdate
// replaces 2006 with 2007 in the current date
Replace "2006", "2007"
debug currentdate
}
20060818
20070818
```

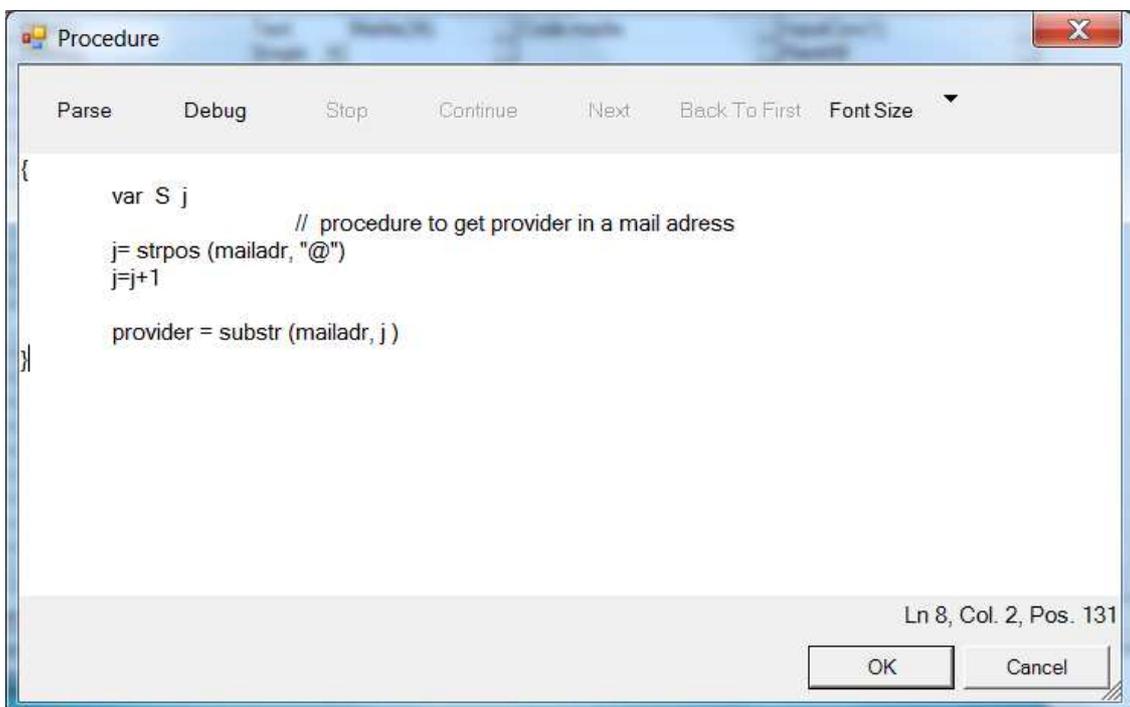
Manipulating sub-strings:

Strlen (varT) : returns the length of the string contained in the Text variable varT.

Strpos (varT, « xyz ») : returns the position of the specified sub-string xyz first encountered within the character string in varT. (the value will be -1 if sub-string xyz doesn't exist).

Substr (varT, j, k) : extracts from the variable varT , a sub-string with k characters, starting at position j or :

Substr (varT, j) : extracts from the variable varT, the sub-string starting at position j.



The screenshot shows a window titled "Procedure" with a menu bar containing "Parse", "Debug", "Stop", "Continue", "Next", "Back To First", and "Font Size". The code in the window is as follows:

```
{  
    var S j  
        // procedure to get provider in a mail adress  
    j= strpos (mailadr, "@")  
    j=j+1  
  
    provider = substr (mailadr, j )  
}
```

At the bottom right of the window, the cursor position is indicated as "Ln 8, Col. 2, Pos. 131". There are "OK" and "Cancel" buttons at the bottom right.

Conditional expressions and control structures (CPL)

Conditional expressions

Control structures use conditional expressions to control the flow of execution of instructions. When a condition is evaluated, the value TRUE or FALSE is returned.

Syntax

Expression	Description
(condition)	The condition may be enclosed in parentheses
(condition condition)	OR
(condition & condition)	AND
(condition ~ condition)	EXCEPT
(condition ^ condition)	XOR
(! condition)	NOT
(argument1>argument2)	Greater than
(argument1>=argument2)	Greater than or equal to
(argument1<argument2)	Less than
(argument1<=argument2)	Less than or equal to
(argument1=argument2)	Equal to
(argument1<>argument2)	Not equal to
(argument1)	argument1 is not MISSING

Note: argument1, argument 2 are variables or constants
 (arg1 = 1, 2, 3) signifies ((arg1 = 1) or (arg1 = 2) or (arg1 = 3))
 (arg1 <> 1, 2, 3) signifies ((arg1 <> 1) and (arg1 <> 2) and (arg1 <> 3))

Special case for Multiple variables

Expression	Description
(MV1 = MV2)	TRUE if the intersection of MV1 and MV2 is not empty, or if both MV1 and MV2 are MISSING (@na)
(MV1<>MV2)	TRUE if the intersection of MV1 and MV2 is empty and both MV1 and MV2 are not MISSING (@na)
(MV1 > MV2)	TRUE if MV1 is not MISSING and MV1's greatest value > MV2's greatest value
(MV1 < MV2)	TRUE if MV1 is not MISSING and MV2's greatest value > MV1's greatest value

Notes:

- **Precision for Quantity (floating point) variables:** Where Q (Quantity) type variables are concerned, the tests = and <> can be faulty due to the precision with which the computer represents floating point numbers. Some floating point numbers cannot be coded exactly by the computer (Cotab uses IEEE double precision standards for internal representation of Q (quantity))

type variables and constants). It is recommended that you use a test such as $VQ > -0.00001$ and $VQ < 0.00001$ in place of $VQ = 0.0$.

- **Dimension of a simple condition** is determined by the variables used in the condition (they must all have the same dimension or be undimensioned).
- **Dimension of a combination of conditions** is determined by the dimension of the simple conditions comprised in the combination (they must all have the same dimension or be undimensioned).

Control Structure

Control structures are simple conditional instructions (if... else...) or loops (for...)

Since they are instructions, they can be nested within other conditional instructions or loops. Note that the brackets '{' and '}' must be used to enclose 2 or more instructions in the following syntax.

The dimension of a control instruction is determined by the dimension of the conditional expression used.

Simple conditional instructions

Conditional instruction	Description/Example
if (condition) instruction	CPL first evaluates the condition. If it is TRUE, CPL will execute the instruction. <i>Example :</i> if ((v = 1) & (VQ <= weight1) (VQ >= weight2)) myvar = VS
if (condition) instruction 1 else instruction 2	CPL first evaluates the condition: If it is TRUE, CPL will execute the instruction1, if it is FALSE , CPL executes instruction2.

The screenshot shows a debugger window titled "Procedure" with a menu bar containing "Parse", "Debug", "Stop", "Continue", "Next", and "Back To F". The code being debugged is as follows:

```

{
var S j,k, z[4]
z=0

if(depars>0){z=depars+1
            debug " depars=",depars," z=",z
            }
else z=999
debug depars, " z=",z
}

```

The execution output at the bottom of the window is:

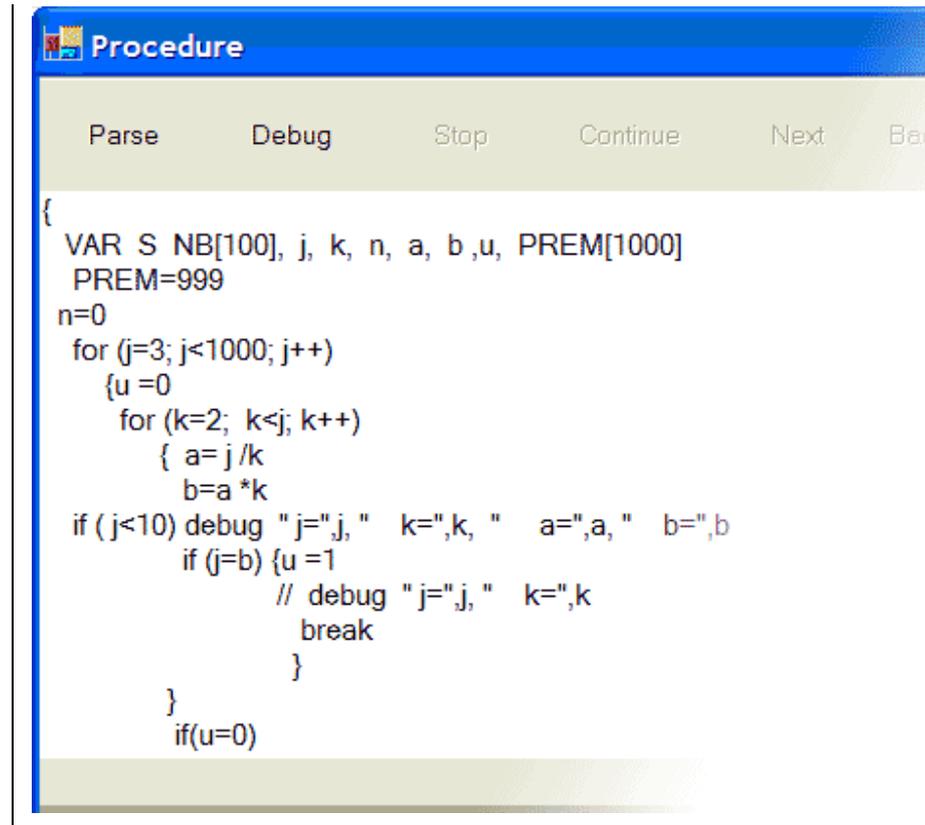
```

depars=55 z=56
depars=9 z=10
55 z=56
@na z=999
9 z=10
0 z=999

```

Loops

Loop instruction	Description/Example
<pre> for ([start_instruction]; [condition]; [loop_instruction]) instruction </pre>	<p>CPL first executes the optional start_instruction.</p> <p>For the loop, CPL evaluates the condition; if it is TRUE, CPL executes the instruction then the loop_instruction and then performs the next loop. Otherwise, CPL exits from the loop.</p> <p><i>Example :</i></p> <pre> for (i = 1; i <= 10; i++) { vs_tab[i] = i } </pre>



```
{
VAR S NB[100], j, k, n, a, b,u, PREM[1000]
PREM=999
n=0
for (j=3; j<1000; j++)
  {u =0
  for (k=2; k<j; k++)
    { a=j /k
    b=a *k
if (j<10) debug "j=",j, " k=",k, " a=",a, " b=",b
    if (j=b) {u =1
              // debug "j=",j, " k=",k
              break
            }
    }
  }
  if(u=0)
```

Warning: CPL does not possess any protection against an infinite loop which will cause the study (and VoxcoTab) to crash.

"Break" and "Continue" instructions

Sometimes there is the requirement to exit a loop when a particular condition is encountered (sometimes known as an exception); the break instruction enables you to exit the loop immediately.

Example :

```
for (i = 1; i <= 10; i++)
{
    // Limit : When i is greater than
    // the dimension of the variable vs_tab, exit the loop
    if (i > vs_tab.dim) break
    vs_tab[i] = i
}
```

Another requirement is to skip the remaining instructions for the current iteration and start the next iteration of the loop. This is achieved using the continue instruction.

Example :

```
for (i = 1; i <= 10; i++)
{
    mytab[i] = i * 2
    if (i < 3) continue // when i < 3 skip the following instructions
                        // in the loop
    vs_tab[i] = i
}
```

Special instructions

exit and stop

To assist in debugging a procedure, and to generate a summary of its execution, CPL provides the following special instructions:

`exit`: immediately quits one of the three sections in the procedure.

`stop`: stops execution of the study.

Given that a procedure is a user routine called by the main application, the `stop` instruction cannot function as a hard stop to the main application. In its place, the CPL engine sets the main application's internal stop flag to TRUE (this simulates a mouse-click on the Stop button or the encounter of an end-of-file condition). The state of this flag is tested by the main application each time that the entry file is read; the application will stop itself before the next read operation.

For example, while correcting a procedure, you may wish to stop execution when 3 records have been read (to avoid having too many debug messages). The following code will do this:

```
{ var S eff
  eff++
  if (eff = 3) stop
}
```

reset

This instruction deletes the file; it is normally used in the Begin section of a procedure.

```
@reset «c:\\xxxx »
```

where (xxxx is the name of the file)

debug and write

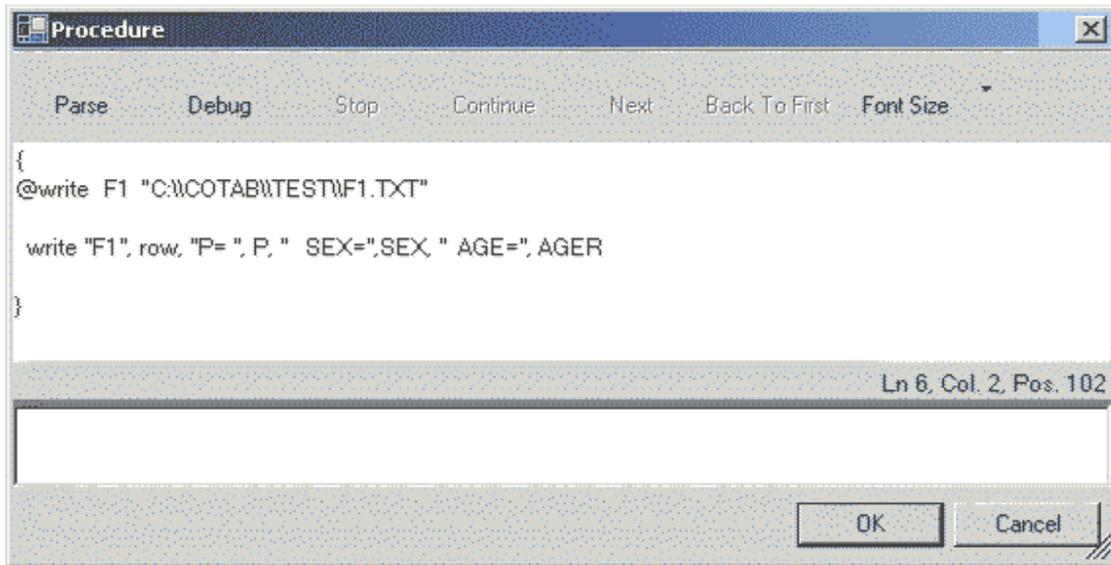
```
debug arg1, arg2, ...
```

```
write « filename », arg1, arg2, ...
```

The dimension of these two instructions is determined by the dimension of `arg1, arg2,...` (they must have the same dimension or be undimensioned).

`debug` displays the message in the lower pane of the procedure window.

`write` will write the specified arguments to the file « filename » specified in the `@write` statement.

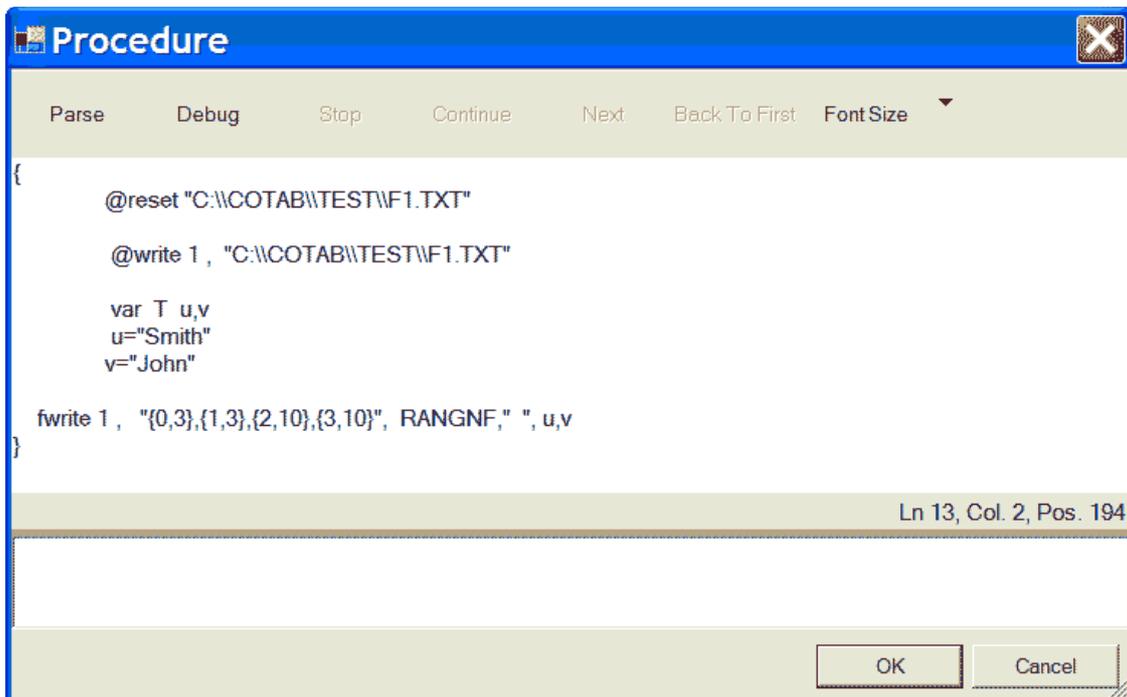


fwrite

This instruction writes to a file according a format which you specify.

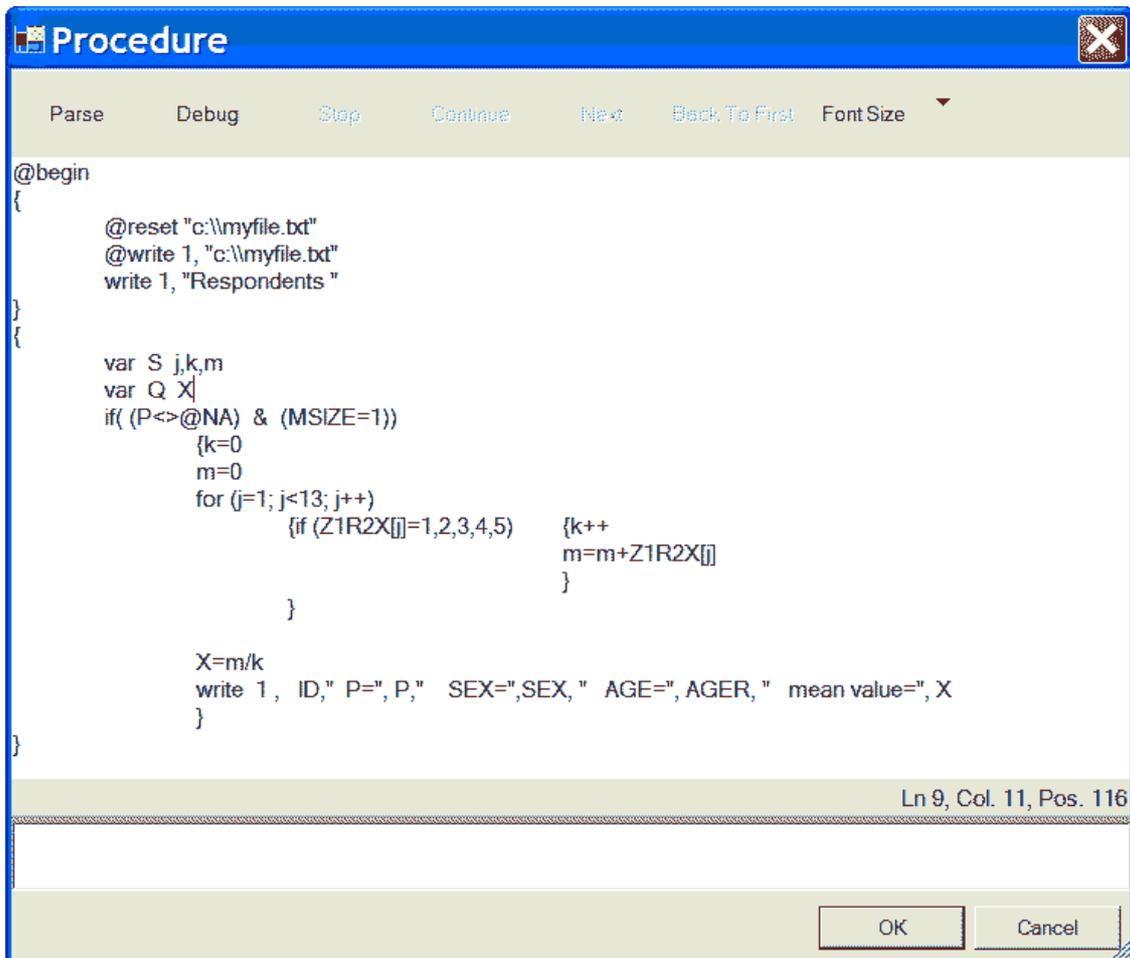
`fwrite "filename" , " {0,n1},{1,n2},{2,n3},... " , arg1, arg2, ...`

Compared to the write instruction, you must additionally specify the number of characters for each argument i thru n which is to be written to the output file.



Important note concerning write and fwrite: the filename specified in the @write instruction is not internal to the procedure, it can be accessed in all other procedures. Hence, you can write to the same file from several procedures.

Example of a procedure



```
@begin
{
  @reset "c:\myfile.txt"
  @write 1, "c:\myfile.txt"
  write 1, "Respondents "
}
{
  var S j,k,m
  var Q X|
  if( P<>@NA) & (MSIZE=1))
    {k=0
     m=0
     for (j=1; j<13; j++)
       {if (Z1R2X[j]=1,2,3,4,5)   {k++
                                   m=m+Z1R2X[j]
                                   }
       }

     X=m/k
     write 1, ID," P=", P," SEX=",SEX," AGE=", AGER," mean value=", X
  }
}
```

Ln 9, Col. 11, Pos. 116

OK Cancel

Predictive Model

Predictive Model

Overview.

A 'data prediction' problem occurs in a number of situations:

- Missing values in a survey.
- Risk Evaluation. Calculating a probability factor to predict abandonment of subscriptions, credit defaults etc.
- Data merging, for the purpose of enriching a client file (for example) based on data collected in a sample-based survey.

The aim is not to create new cases, but to compute values for variables not filled through data collection.

To handle this problem, the basis is a **pilot study** (or a model) which provides values to the **explained** variables, on the one hand, and a set of **explanatory** variables, on the other.

A **data model** is thus constructed, which enables the **explained** variables for each case to be calculated on the basis values of the **explanatory** variables.

The model uses a proprietary method called RESCUT, which operates using single or multiple explanatory variables with Formats i.e. categorized. The explained variables may be of any type (except test).

Implementation.

Data prediction makes use of 2 modules:

1. **Model Builder.** In this module, you declare:
 - **Explanatory variables**, Single, Multiple, or Text, with Format (categorized). Non-responses are permitted.
 - **Explained variable(s)**, Quantity, Single with or without Format, Multiple with Format, or Logical.

Execution creates the predictive model, automatically stored in a file.

2. **Model Request.** You can call this module either in the current study or from another study. In the latter case, the variables dictionary must contain both the explanatory and explained variables using the same attributes (level, type, dimension and format) as the pilot study.

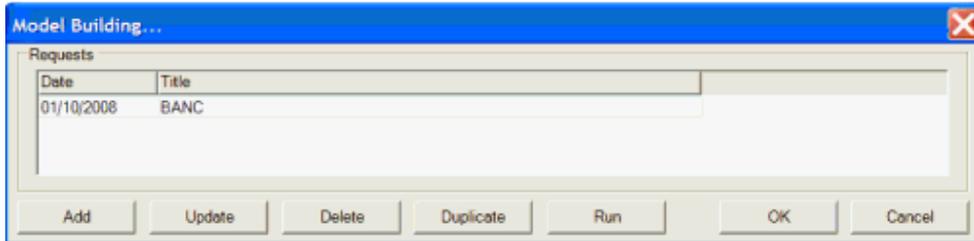
Execution of this module provides a file containing, for each case, the values of the explained variables. If you wish to insert these values into a study, for the purpose of producing crosstabs for example, simply declare this file as an external table, and define ImportCsv functions for all the explained variables

Note: The model will function even if there are non-responses for the explained variables. However, this proportion of non-responses will be reproduced in the 'prediction'; this is legitimate if the non-response to an explained variable has a precise meaning (example 'Not Applicable'). However, it will not be legitimate if the non-response refers to 'Don't Know'. In the latter case, it would be appropriate to include in the model a filter to exclude cases with non-responses to the explained variable.

Building the predictive model

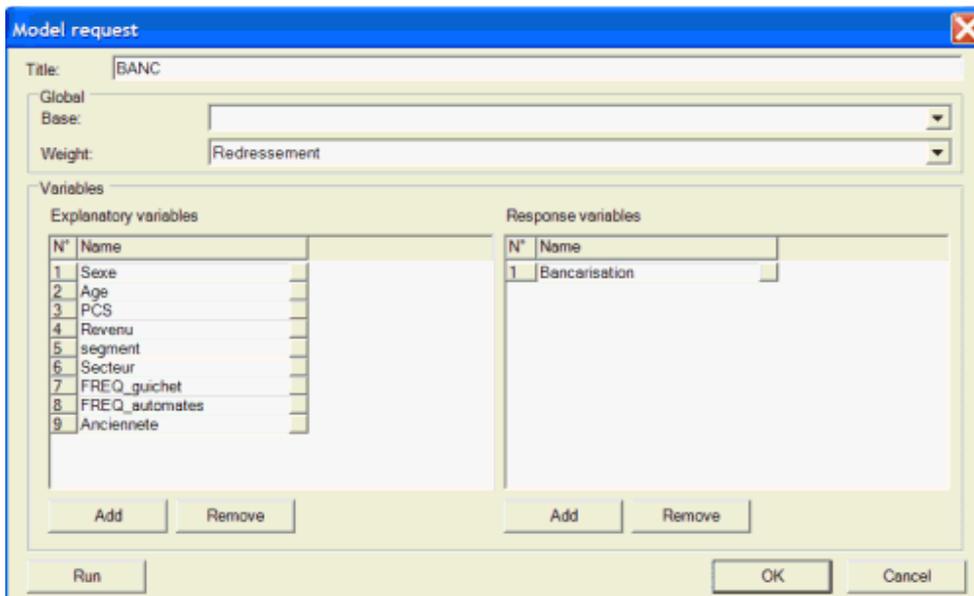
Select Utilities>Model Building in the main menu; this opens the Model Building window.

You can Add, Update, Delete, Duplicate requests, and Run them



A Model Request

An example of a model request is shown in the screen-shot below:



A Model Request comprises:

Setting	Description
Title	Any title which will be used in the names of the files created
Base	Optional: The name of a logical variable determining the universe on which the model should be created
Weight	Optional: A quantity type variable, if the model is to be created taking into account the weight assigned to each case
Explanatory variables	When you click on the button to the right of a name, you can choose variables with a Format (Single, Multiple or Text type) possibly dimensioned
Explained (response) variables	When you click on the button to the right of a name, you can choose one or more variables (any type except Text), with or without Format, possibly dimensioned If there are several Explained variables, a model will be created for each of them

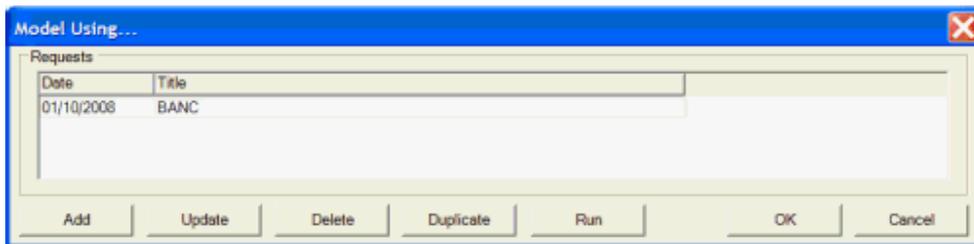
Executing a model on a data file produces a file in the study folder. Its name will be Mnnn_title.txt, and it will contain the model of all the explained variables (nnn is the study number, title is the title of the request)

Study with levels

If the study has levels, the model's level is by convention that of the lowest level variable specified in the request (i.e. the highest level number). The model will be built on the cases at this level. All the explained variables must belong to this level, otherwise the results will be unpredictable.

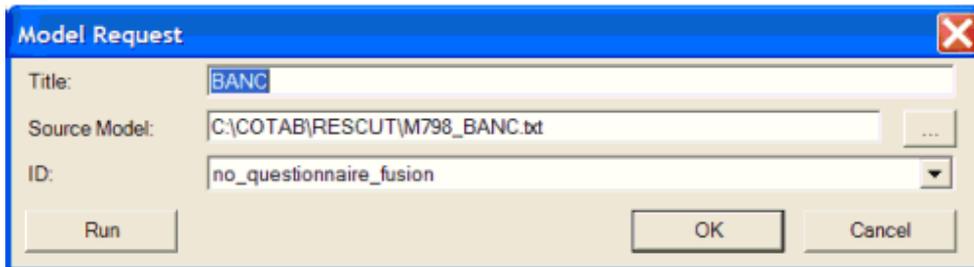
Using the predictive model.

Select Utilities>Model Using in the main menu; this opens the Model Using window



You can Add, Update, Delete, Duplicate requests, and Run them

A request is shown in the screen-shot below:



The request requires that you specify the file containing the source model(s) to be used, and a variable which serves to identify the cases.

The study being used must contain the explained variables (with data of course) and the explanatory variables, configured with the same attributes as those used in the Model Builder.

Execution will produce a CSV file in the study folder bearing the name Rnnn_title.csv (nnn is the study number, title is the request title). Each case in the file will have:

- First field: The value of the ID variable
- Following fields: The values of the explained variables in their order of appearance in the Model request.

Following execution, you will need to declare this file as an external CSV table, and declare an ImportCSV function for each of the explained variables.

Simulation

Overview.

This module, in the "Utilities" menu, responds to the following question: one often asks, for example in satisfaction surveys, « what is the contribution of items in overall satisfaction? ". Intuitively, we can understand the underlying query, however this wording seems vague and misuse.

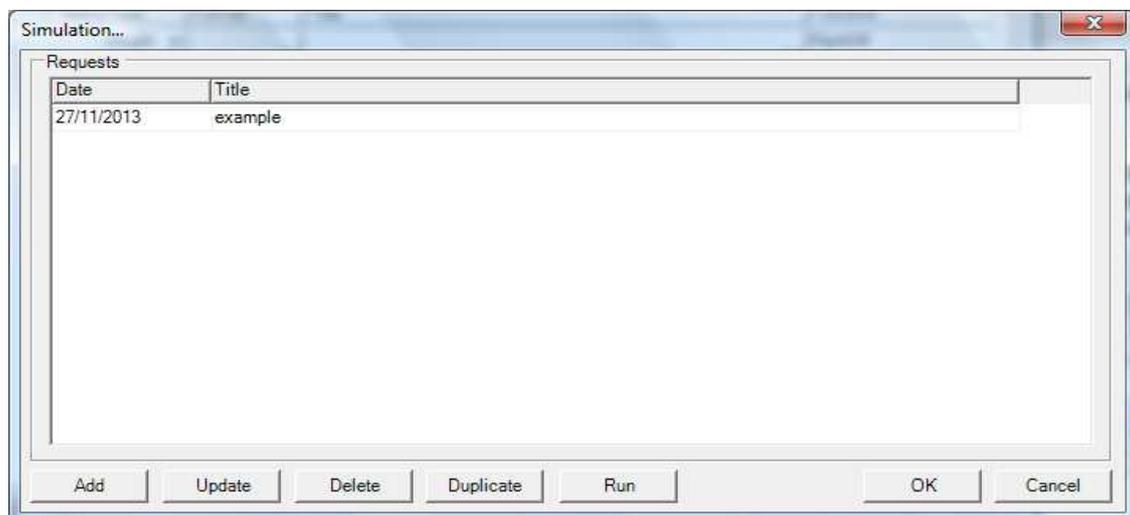
So, we prefer to replace it with this much more accurate and non-disputable statement: what happens (what-if?) if, for every item successively and independently, the answers were all the same. For example, if the satisfaction study includes questions such as overall satisfaction, then items like 'speed', 'friendliness', 'quality of response', etc, we want to know what happens to overall satisfaction if for example all the interviewed were very satisfied with the speed of processing.

This way of doing gives the influence of every item, all other things being equal. To make a simulation, one should have a **variable to simulate** (Quantity type, or Single without format, or Single with format, or Multiple with format, or Logical), and **categorical explanatory variables**, i.e. Single or Multiple, or Text type, in these 3 cases with a format.

Of course, the simulation is based on a **predictive model built previously**.

Simulation request.

You should first build the predictive model (menu 'utilities') corresponding to the variables to be explained and explanatory concerned. Then, by clicking on the line 'simulation' in the menu 'utilities', you get a screen to create simulation applications.

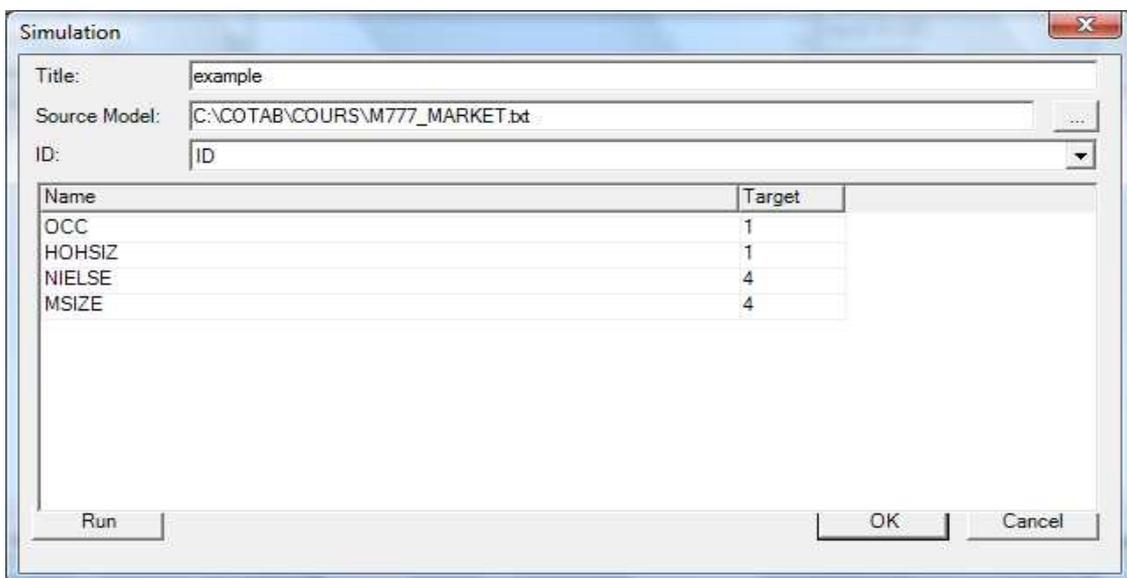


You can create requests, modify them, duplicate them, delete them, run them finally. A request corresponds to the screen below. Must be given:

- the name of file, created by «predictive model building», containing model to use,
- an ID variable to identify individuals
- for each listed explanatory variable, the 'target', i.e the value to give to this variable for all individuals of the file.

Notes:

- 1) If there are **several** variables to explain in the predictive model, the simulation relates only to the first.
- 2) **Multiple** explanatory variables are not considered in the simulation.
- 3) The explanatory variables of type **Text** are taken into account, but should be written with care (e.g. 'very satisfied')



Run will provide, in the directory of the study, a csv file, named **Snnn_title.csv**, (nnn is the number of the study, title being the title of the request), with, for each individual:

- first field: value of the variable ID
- following fields: values of the simulated variable for each simulation: a simulation corresponding to one explanatory variable taking, for all individuals, the value in the field 'target' (target).

You can declare this file as an external csv table, and use an Import-csv function to get the value of the variable for all simulations.

Comparison of all these values will give the respective action of all explanatory items.

Export CoTab Reader

Export CoTab Reader

Overview.

The purpose of this Utilities module is to prepare for a customer a tool called CoTabReader for viewing study tables, and for building all other tables he/she might require.

The tool delivered to the end customer includes:

- A database consisting of study cases, variables, and tables.
- The software, a subset of CoTab, for building new tables.

This software CoTabReader is freeware (no license required), but it only works on the exported study, and it does not access the modules Export Data, Weighting, and Predictive Model.

The variables in the database cannot be changed by the CoTab Reader user, but he/she can build new variables from existing ones with most of CoTab functions.

Creating a Cotab Reader for a given study..

In the main menu click on **Utilities>Export Cotab Reader**. In the screen appears you can create requests, update, duplicate, delete, and finally run them.

The **Variables tab** allows you to select the variables that you wish to export in the database,

The **Requests tab** allows you to select the table folders that you wish to export.

You can export all or part of the cases, some or all of the variables, some or all of the tables.

The request must be consistent, if not a test error occurs at run time, and the execution does not occur.

Running CoTab Reader.

Running CoTab Reader creates, in the requested directory, the software CotabReader paired with the exported database. This entire directory can be copied onto any computer to be run freely.

CoTabReader.

By clicking on CoTabReader.exe, the user enters directly into the associated study.

- The user cannot modify existing variables, but can create new ones, built from existing ones.
- The Preview and Frequency Tables options are available.
- The Templates are initialized with those of the original study. In Tables, the folders that were selected on exportation are included.

These tables can be run, printed or exported. New tables can be created using all the features of CoTab.

CoTab Reader

CoTab Reader - Overview

CoTabReader is a software application for editing statistical tables. It runs under Windows (XP, VISTA, Windows7), with Framework 3.5. installed.

It operates on attached studies created by the Voxcotab application, and enables you to:

- View existing tables
- Build new tables
- Create new variables

An attached study comprises:

- Variables (already created)
- Templates for configuring tables
- A set of tables

On launching the application, the following screen appears:

The screenshot shows the CoTab Reader application window titled 'Test on C [0]'. The 'Variables' tab is active, displaying a table with the following columns: N°, Name, Dim, Type, Format, Title, Function, Base, and Preview (>>). The table lists 31 variables with their respective details.

N°	Name	Dim	Type	Format	Title	Function	Base	Preview (>>)
3	rankt		Text	rankt(500)				
4	P		Quantity		WEIGHT			
5	PTRANCHES		Single	PTRANC...				
6	AG2		Single	AG2(3)	Age of respondent / CHILDREN 1-12 (by age brackets)			
7	AGER3		Single	AGER3(9)	Age of respondent / ADULTS 13 (by age brackets)			
8	SEX		Single	SEX(2)	Sex			
9	MEN		Logical		MEN			
10	AGER		Single	AGER(5)	Age of respondent in clear			
11	SAGE		Multiple	AGEBIS(...)	Sex and Age of respondent			
12	OCC		Single	OCC(8)	Occupation of the Head of household			
13	F50		Logical		WOMEN under 50			
14	AGEHOH		Single	AGEHO...	Age of the Head of household			
15	HOHSIZ		Single	HOHSIZ(...)	Household size			
16	NIELSE		Single	NIELSE(...)	Nielsen regions			
17	MSIZE		Single	MSIZE(4)	Market Size		MEN	
18	PRES		Multiple	PRES(5)	Presence of children aged 17 or below in the household			
19	VENT1		Multiple	VENT1(1...	Age & Gender (by age brackets) / ADULTS 13			
20	VA1		Multiple	VA1(6)	Age & Gender (by age brackets) / ADULTS 13			
21	QCI		Multiple	QCI(7)	QC - Time when started breakfast			
22	QCDUCI		Single	QCDUCI...	QC - Time for breakfast			
23	QDI		Multiple	QDI(1)	QD - Had breakfast or not			
24	V1CL2		Multiple	V1CL2(2...	Liste 2 (products consumed at breakfast) on 1 Week			
25	WQAD		Quantity		Weighting factor - Questionnaire Adults 13			
26	INCOME		Quantity		INCOME			
27	WDIARY		Quantity		Weighting factor - Diary /children adults			
28	Z1R2X	12	Single	Z1R2X(8)	Opinions			
29	V8		Multiple	V8(17)	V8			
30	MULTIPLE		Multiple					
31	NBQ6		Single	NBQ6(28)	Q6- Average number of brands known			

The **Variables** tab displays existing variables, and allows you to create new ones.

The **Templates** tab displays existing templates and allows you to create new ones.

The **Tables** tab displays existing tables, and allows you to create new ones.

Note: These abbreviated instructions deal with Tables only.

Attributes of Variables.

The chief attributes of variables are **Type**, **Dimension**, and **Format**. Studies with levels also possess the **Levels** attribute.

Types of variables: S (single), M (multiple), Q (quantity), T (text), and L (logical) are the types typically used in statistical processing. However, note that a logical variable can only have a value of True or False, and that other variables can only have a value corresponding to their type or the conventional Missing value.

Dimension: Used for a series of variables, for example: a set of 10 items for which a score is requested, a list of 30 radio stations asking for listening frequency etc.

Format : Variables of type S, M or T may have a format, i.e. a list of values to be displayed in the tables. The format can be viewed by clicking on the button located to the right of the Format field.

N°	Code	Label	No Sort	Hide	Value
1	1	Strongly agree	<input type="checkbox"/>	<input type="checkbox"/>	5.000
2	2	Slightly agree	<input type="checkbox"/>	<input type="checkbox"/>	4.000
3	1-2	Agree	<input type="checkbox"/>	<input type="checkbox"/>	
4	3	Neither agree nor disagree	<input type="checkbox"/>	<input type="checkbox"/>	3.000
5	4	Slightly disagree	<input type="checkbox"/>	<input type="checkbox"/>	2.000
6	5	Strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	1.000
7	4-5	Disagree	<input type="checkbox"/>	<input type="checkbox"/>	
8	9	dk	<input type="checkbox"/>	<input type="checkbox"/>	

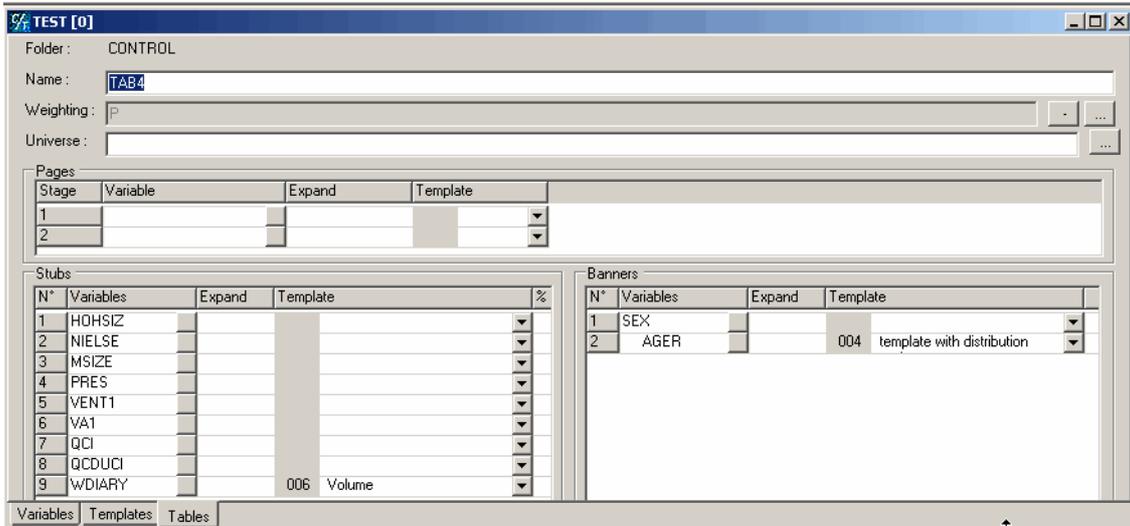
Base: For a variable X, this field can contain a logical variable, in which case the variable X only applies for the value True of this logical variable. In tables, the total for variable X will be the number of True values in this Base variable.

Viewing an Existing Table.

When you click on the Tables tab the following screen appears:

Folders	N°	Request Title
CONTROL	1	TAB1
	2	TAB2
	3	TAB3
	4	TAB4

One or more pre-defined folders are listed on the left. A click on a folder displays its table requests. To open a table request simply double-click on it (TAB4 for example).



A click on the Execute button  in the toolbar executes the table (this step is only necessary if it has not already been done).

Clicking on the Preview button  in the toolbar displays the results.

Folder : CONTROL
Request : TAB4
Page :

TOTAL														
Age of respondent in clear														
Total	15-30		31-40		15-40		41-50		51-99		Mean and SD			
Household size														
TOTAL	500	100%	143	100%	61	100%	204	100%	144	100%	152	100%	42.91	15.79
1	129	26%			18	29%	18	9%	34	24%	77	51%	51.15	10.49
2	162	32%	26	18%	32	52%	58	28%	41	28%	63	42%	48.03	16.42
3	29	6%						18	12%		11	7%	50.40	9.38
4	147	29%	64	54%	11	18%	95	47%	52	36%			33.32	10.13
5 and over	33	7%	33	100%			33	100%					21.47	3.90
Nielsen regions														
TOTAL	500	100%	143	100%	61	100%	204	100%	144	100%	152	100%	42.91	15.79
West	500	100%	143	100%	61	100%	204	100%	144	100%	152	100%	42.91	15.79
Market Size														

The other buttons in the toolbar are :

Button	Description
 Back	Returns to the table requests screen.
 Preview/Request	Toggles between the table's preview and its request.
 Table Settings (see image below)	Opens the page where you specify the table settings controlling the way it is displayed: size, font, colours, title styles, choice of separators, cell contents (counts, percentages, means) and their placement, ordering of rows or columns, statistical tests. Note: The table settings can be saved or loaded from an existing template.
 Page Settings (see image below)	Allows you to set margins, create page headers and footers, edit titles, add logos etc.
 Print	
 Export	Export to Excel or Powerpoint

Table Settings

Table display settings

Stubs

1st stage style : Visible  Show labels 

2nd stage style : Visible  Show labels 

External border : 

Cells separator : 

Line indent : 

Display empty lines

Line ranking :

Banners

1st stage style : Visible  Show labels 

2nd stage style : Visible  Show labels 

External border : 

Cells separator : 

Justify : V : H :

Display empty columns

Column ranking :

Counts

Tests : None Chi2 Proportions Compare to total

CIV CIH Use back colors for Chi2. 

Count	%V	
	%H	

Visible Background color : 

Means

Tests : None Mean test Compare to total

CI

Mean	SD	Count
SE		

Visible Background color : 

Threshold : 90%-95 95%-99 90% 95% 99%

Display percent sign Character for empty cells :

Load
Save As
OK
Cancel

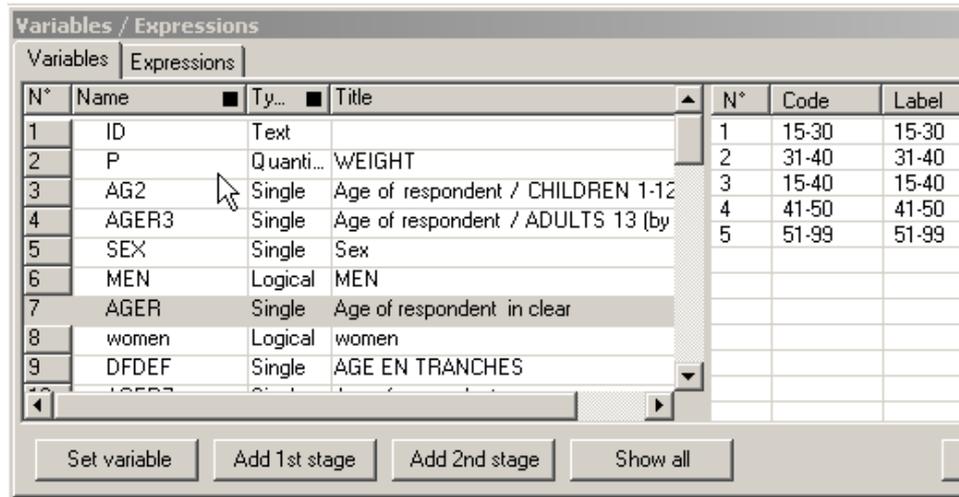
Stubs/Columns

In these areas you insert the variables to be used in the stubs and columns of the table.

To create the first stub or column you may either use a right click or press the Insert key.

Variables

To choose a variable, click on the button to the right of the 'Variables' field to open the list. Click on a variable to select it:



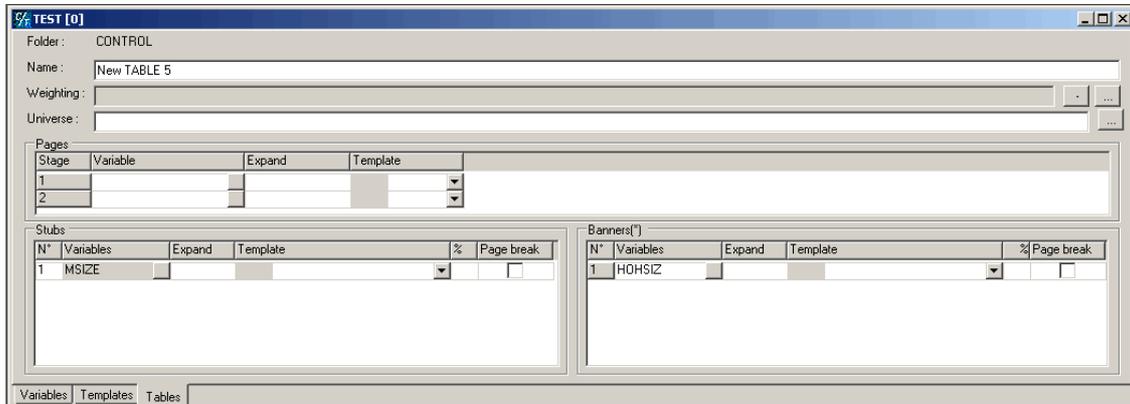
A click on the 'Set Variable' button will add the variable to the currently selected Stub or Column of the table request:

The table is now ready to be executed: . Beforehand, however, it is best to verify that there are no errors: to do so, click on the 'Check' button . Any detected errors will be shown at the bottom of the screen. If there are no errors, you can execute the request and view the results:

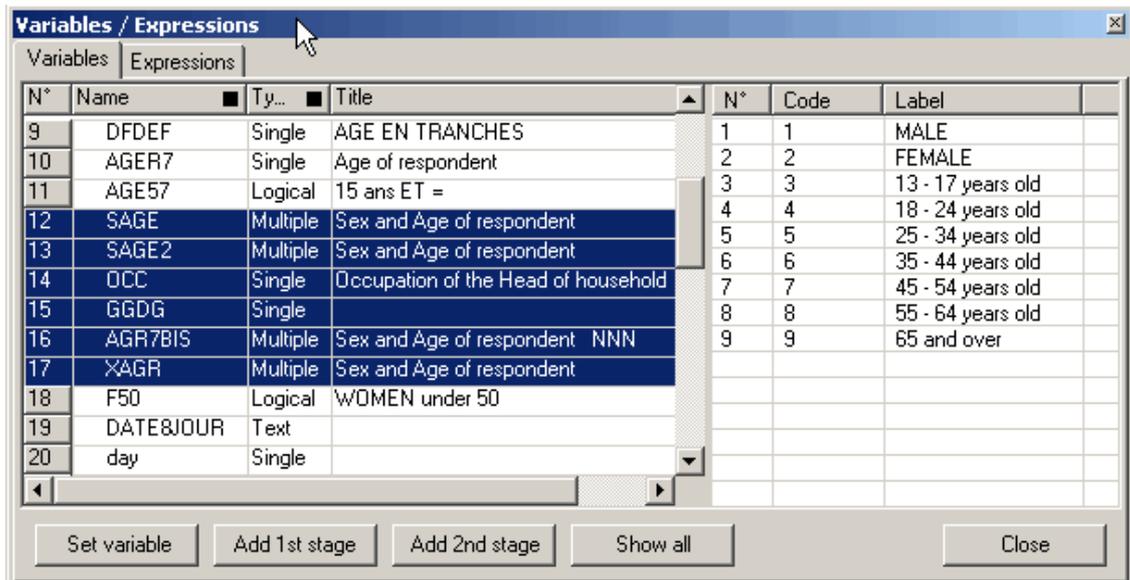
File Edit Tools Utilities Windows ?																
Folder : CONTROL																
Request : New TABLE																
Page : Age of respondent in clear - TOTAL																
Sex and Age of respondent																
TOTAL	MALE		FEMALE		13 - 17 years old		18 - 24 years old		25 - 34 years old		35 - 44 years old		45 - 54 years old			
Household size																
TOTAL	500	100	229	46	271	54	65	13	36	7	67	13	81	16	75	15
1	139	100	90	65	49	35							34	24	27	19
	28		39		18								42		36	
2	149	100	51	34	98	66		6	4		28	19	23	15	10	7
	30		22		36			17			42		28		13	
3	57	100			57	100	28	43	49				13	23		
	11				21								16			
4	122	100	67	55	55	45	25	38	20	21	17	27	40	22	11	9
	24		29		20					58			14		38	31
5 and over	33	100	21	64	12	36	12	36	9	27	12	36				
	7		9		4		18		25		18					
NO ANSWER																
Nielsen regions																
TOTAL	500	100	229	46	271	54	65	13	36	7	67	13	81	16	75	15
Paris and great Paris																

At this point, as before, you can configure the table, print it or export it.

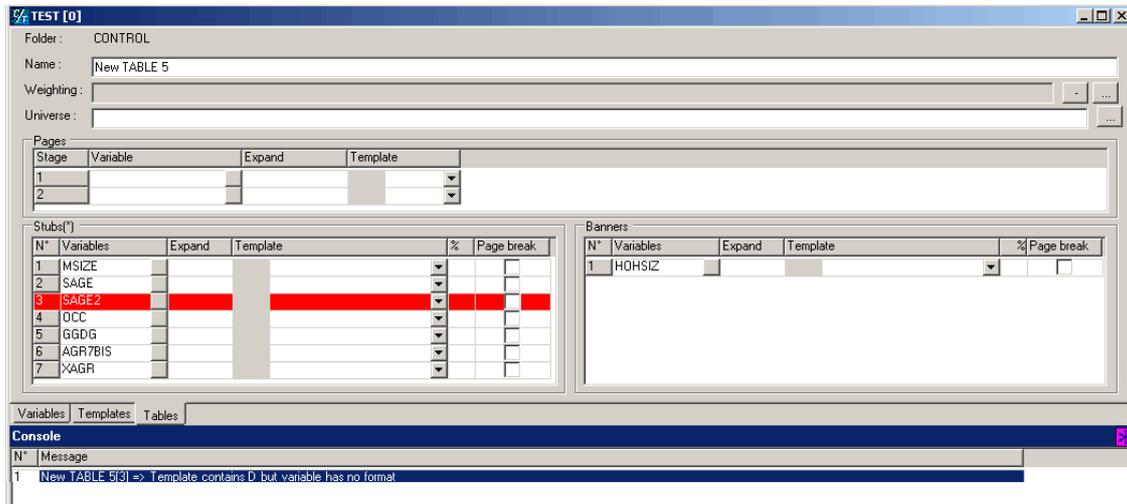
Tables with several variables in rows or columns



Another click on the button to the right of the Variables field reopens the variables selection screen.



After selecting one or more variables you can click on the 'Add 1st stage' button which will place them after the first variable. As a result the 'Column' variable will be crossed with each of the 'Row' variables in succession. You can do the same with 'Column' variables.



The check button tests for the coherence of the table set-up.

Templates

The template associated with a variable in rows, columns or pages of a table specifies what should be computed and displayed for this variable.

The chief options are:

T : Total for the variable. This includes the whole 'Universe' unless the variable is associated with a 'Base' in which case the Total comprises the number of cases which are 'True' in this 'Base'.

N : The number of 'No Answers'.

A : Number of cases (at least one response).

D : Distribution, corresponding to the rows in the variable's 'Format'.

M : Mean.

LO : List of Occurrences (for dimensioned variables).

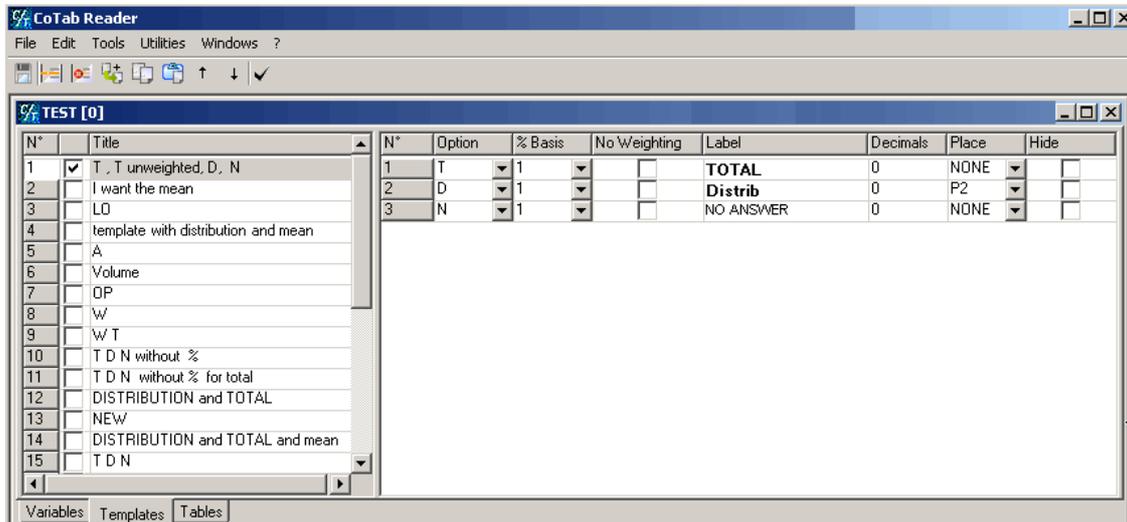
The types of variables possible with each template option are:

Option	Variable Type	Comment
T, A, N	All types	The total for the variable taking into account any base that might be associated with it.
D	Single, Multiple, Text	With Format.
C, M, Mi, Ma	Quantity, Single, [Multiple, Text]	M, T: with Format and Values in the Format
NB,NT	Multiple	The number of responses for a multiple variable.
LO	All Types	Dimensioned variables.
W	Quantity	Rarely used

A template is a set of one or more of the above options. For example, the first template shown below comprises three options:

- Total for the variable
- Distribution
- No Answers

Existing templates can be viewed in the Templates tab:



The 'Default' template is the one selected.

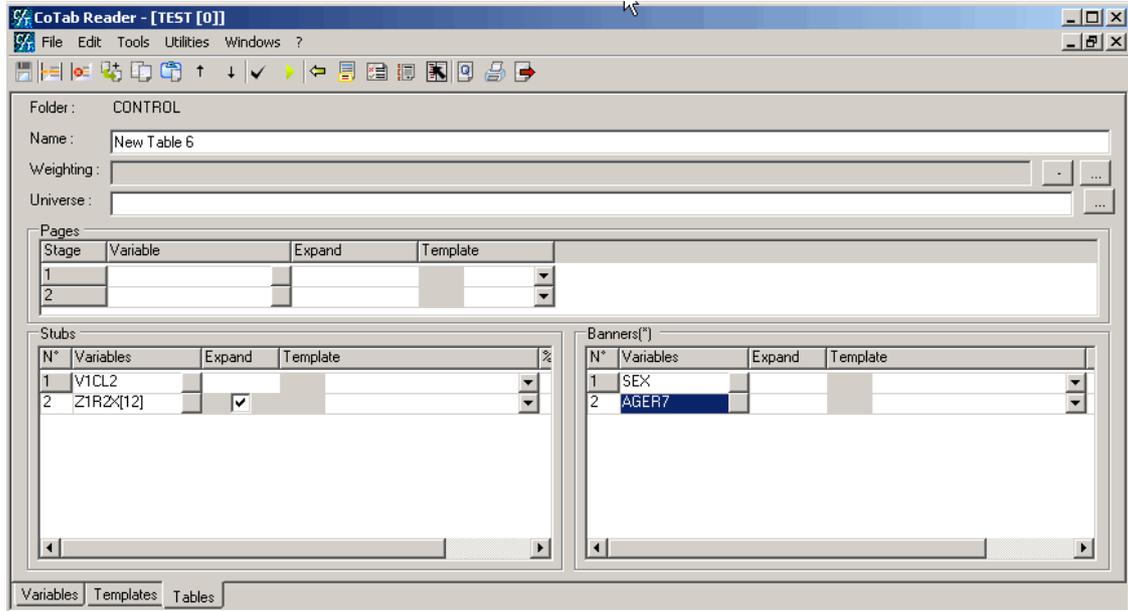
When you check a table using the 'Check' button, error messages might appear at the bottom of the page. If you click on a message in the error list, the corresponding line in the table request will be highlighted in **red**. Usually errors are due to inconsistencies between the variable and the template.

Example: Template contains D but variable has no format.

In the above example of an error in paragraph 4, the variable WQAD's type is Quantity, the template is not specified hence the Default template applies; this consists of the variable's Total, Distribution, and the number of No-Answers. Given that WQAD has no format, the D option is impossible.

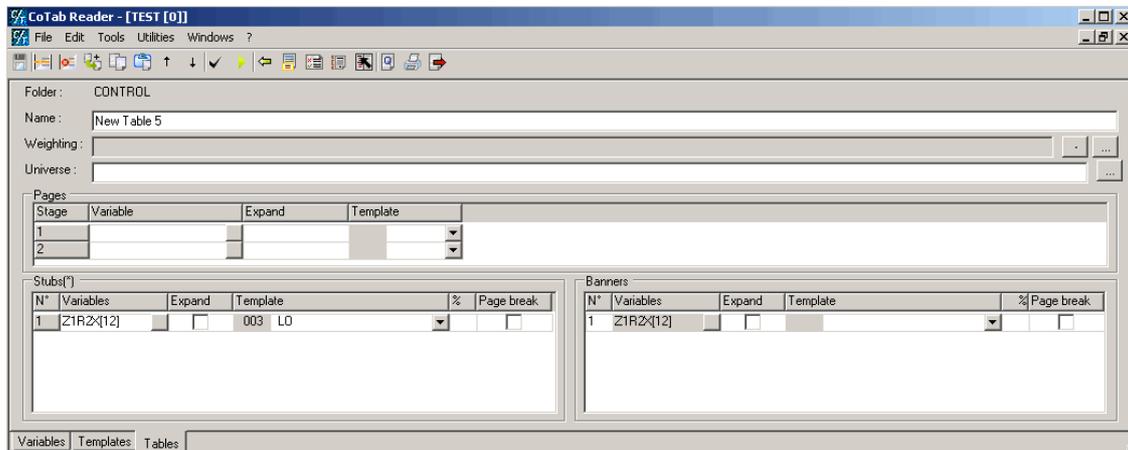
Dimensioned Variables

Dimensioned variables which are selected globally can be expanded by clicking in the corresponding checkbox; the effect is exactly the same as if each occurrence had been selected individually.



You can also use the LO template option, by crossing the variable with itself, using the LO option together with a classic template.

For example, by using the LO option for the rows and the default template for the columns:



After execution, the preview is:

CoTab Reader - [TEST 01]

File Edit Tools Utilities Windows ?

Folder: CONTROL
Request: New Table 5
Page: [dropdown]

Opinions																			
TOTAL	Strongly agree		Slightly agree		Agree		Neither agree nor disagree		Slightly disagree		Strongly disagree		Disagree		DK	NO ANSWER			
Opinions																			
I like trying new food products	500	100	60	12	198	40	258	52	123	25	62	12	30	6	92	18	27	5	
I often eat ready-made cooked	500	100	27	5	104	21	131	26	77	15	127	25	138	28	265	53	27	5	
I try not to snack between meals	500	100	203	41	139	28	342	68	114	23			17	3	17	3	27	5	
Breakfast is my favourite meal of the day	500	100	80	16	56	11	136	27	236	47	87	17	14	3	101	20	27	5	
A full breakfast is essential to get going	500	100	223	45	148	30	371	74	91	18	11	2			11	2	27	5	
A balanced breakfast helps	500	100	183	37	210	42	393	79	69	14			11	2	11	2	27	5	
Breakfast is a meal where each person can shine	500	100	176	35	177	35	353	71	120	24							27	5	
I frequently vary what I eat all day long	500	100			83	17	83	17	78	16	196	39	116	23	312	62	27	5	
During the weekend or on holiday I eat differently	500	100	54	11	46	9	100	20	97	19	122	24	154	31	276	55	27	5	
I really have to force myself to eat breakfast	500	100	13	3	9	2	22	4	40	8	111	22	300	60	411	82	27	5	
Between summer and winter, I eat differently	500	100			78	16	78	16	26	5	98	20	271	54	369	74	27	5	
I like lunch and dinner, breakfast is not my favourite	500	100	197	39	130	26	327	65	66	13	66	13	14	3	80	16	27	5	

Variables | Templates | Tables

Requests from a Folder

The following toolbar buttons are available at the Folder level, not just at the Table Request level:

Button	Description
 Execute	All table requests in the folder will be executed automatically.
 Table Settings	Enables Table Display settings to be defined for all table requests in the folder.
 Page Settings	Enables Page Display settings to be defined for all table requests in the folder.
 Preview	Preview before printing.
 Print	All tables in the folder will be printed in a single document
 Export	All tables in the folder will be exported in a single file.

Reports (Excel)

Reports (Excel)

Overview

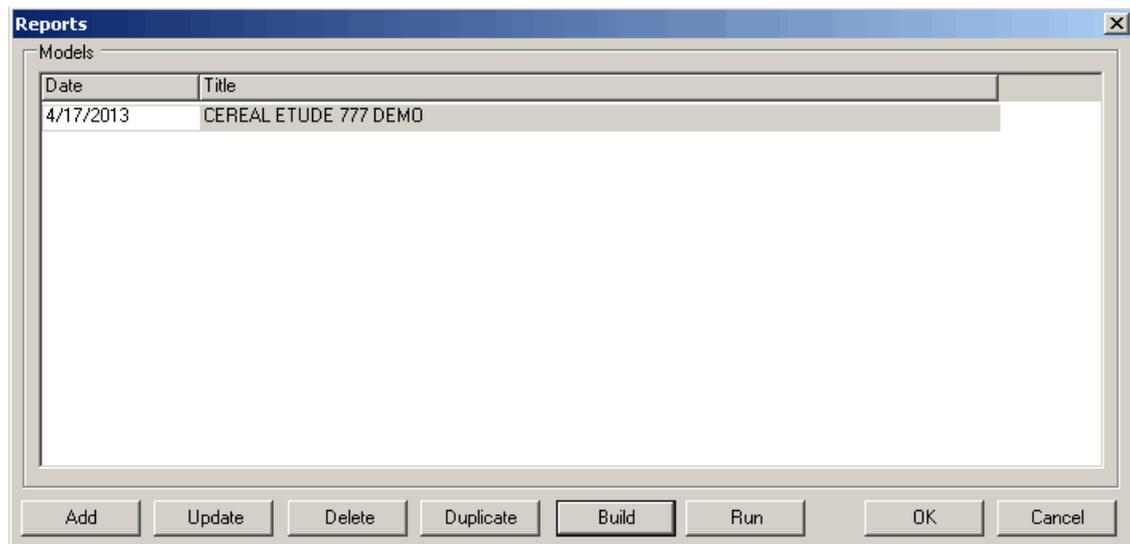
Once tables have been exported in Excel format, one generally customizes a report making use of many of Excel's options and resources. Everything is fine if these tables are satisfactory at first go, and if there is no need to apply this work to other data. However, outputting the same tables using a different data source file would overwrite the report that had been built in the beginning, requiring all the customization work to be done again.

The Reports module gets around this problem; the steps are as follows:

1. Select requests from one or more folders containing the tables constituting the base of the report.
2. Output them to an Excel 'Data Model' which contains the results of all the tables in a single 'data' tab (in contrast to the classic Excel export).
3. Build the report in Excel inside other tabs i.e. not the 'Data Model' tab which contains the exported tables.
4. When there is a need to make further edits on the same tables, for example using a different datafile, outputting to the same Excel 'Data Model' will not overwrite the report. In this way you can separate your work into two phases: 'table editing' and 'report customization' respectively.
5. When everything is correct, you can then generate your final report.
6. An option also enables you to break out this report for all values of a control variable (known as a 'population' variable). This produces as many reports (Excel files) as there are values for this variable.
7. An Excel macro is provided to copy the tables from the 'data' tab to the tabs containing the report. It respects their layout, and embodies the links necessary to automatically update the report to reflect changes in the tables.

Instructions

The 'Reports' module is provided under the 'Utilities' menu.



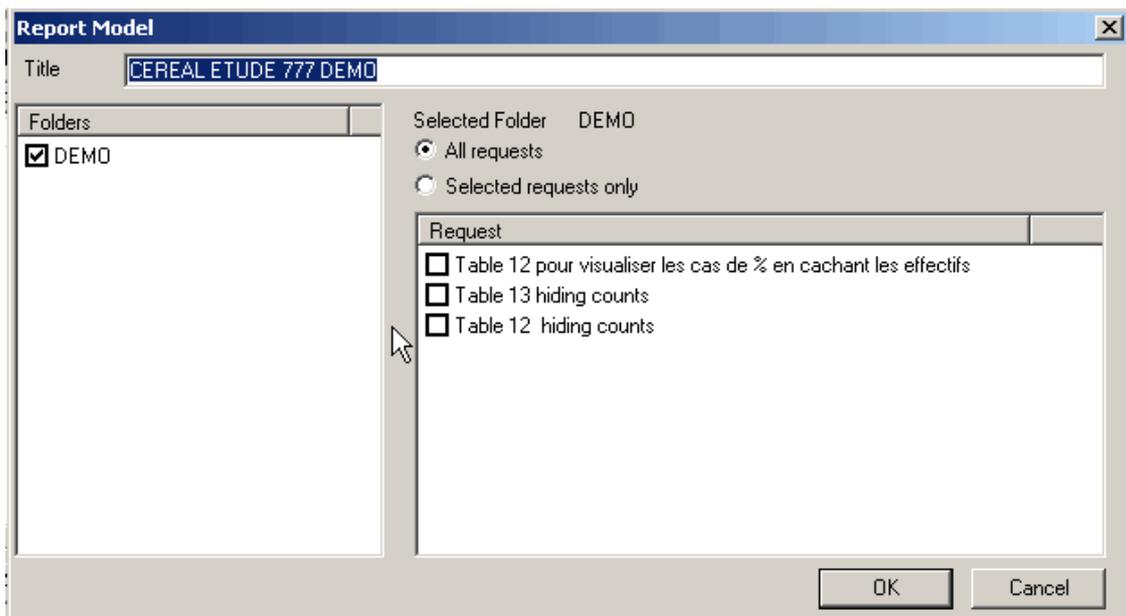
Buttons are provided, as usual, to modify, delete or duplicate a report.

This module comprises 3 phases:

- Creation (or modification)
- Building
- Running

Creation

Select the required folders and table requests



Building

The 'Build' button creates an Excel file with a Data Model tab containing the results of the requested tables (assembled in vertical sequence in a single tab), and a 'Map' tab. This file bears the name XXX.model.xlsx (XXX is the name that you gave it). You can then construct your report in tabs 3,4 etc.

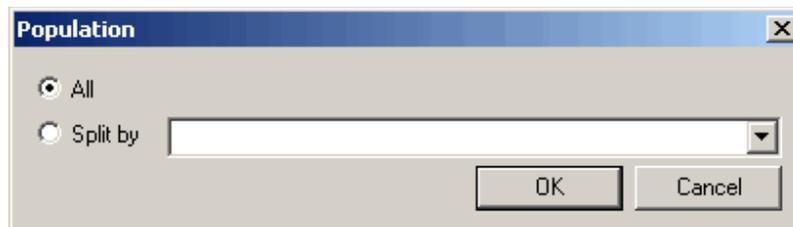
The Excel macro CotabReports.xla which is provided with the application enables you to easily copy tables in the Data tab and paste them into another tab, both respecting the layout and preserving the links.

Notes:

- It is not necessary to 'Run' the tables before launching the 'Build': the 'Build' command will execute the selected tables to edit the Excel file.
- For the 'Build' to operate, the study must be free of any errors, including tables not selected in the 'Report' command. You should verify this beforehand by using the 'Check' button.
- Filenames and table request names within a folder must, of course, be unique.
- Do not modify the contents of the Data and Map tabs.
- Once you have built your report, if you launch 'Build' again (for example using a new datafile) the XXX.model.xlsx will be replaced, and the new Data and Map tabs will be placed after the tabs which you created for your report. You must make sure that the links in your report tabs are consistent with the zones in the Data tab, otherwise you are likely to have meaningless results. From one 'Build' to the next, the tables must be the same (with the exception of additions), have the same structure (no modifications to variable formats changing the number of values) etc.

Running

The 'Run' button creates the definitive report in another XXX.xlsx file (the Data and Map tabs do not appear).



The 'All' option creates a single file corresponding to the complete data file.

The 'Split by' option enables the creation of multiple reports, one for each value category of a selected variable (subtotals included). The variable (Simple, Multiple or Text type) must have a format. The name of each file will be identified by the label of the corresponding value category.

Note: When using the 'Split by' option it is important to be able to include in each report the name of the population in question. For this reason the Data tab includes the name of the population variable in cell A1, and the label of the value category in cell B1.

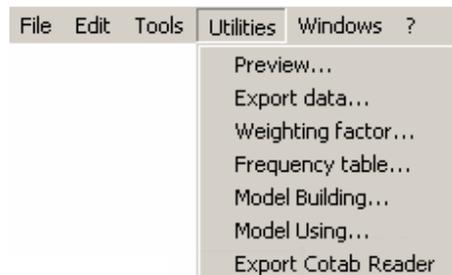
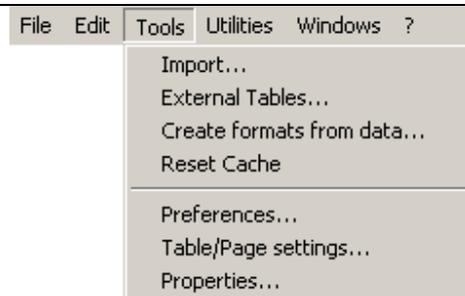
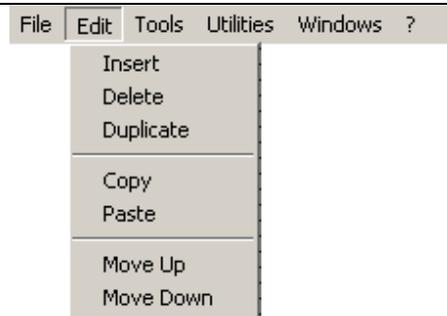
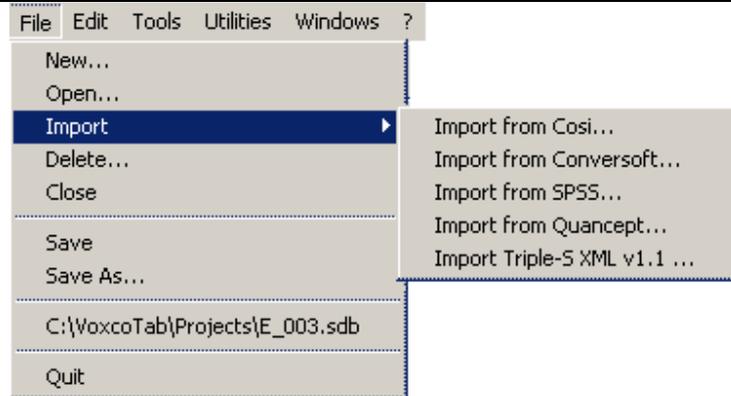
General Notes

Several precautions must be taken when using the Reports module, some of which are discussed above:

- There must be no errors in the study, even within tables not selected in the request.
- Both in 'Build' and 'Run' phases, existing Excel files involved must be closed.
- The version of Excel must not be earlier than 2007.
- Do not make modifications in either the Data or Map tab.
- From one 'Build' to another for the same report requests, and also in the 'Run' phase, the tables must not be modified: no new tables (except appended ones), no changes in the number of columns or rows in a table. To summarize, the structure of the Data and Map tabs must remain the same, otherwise the links enabling the copy of cells between the Data tab and reports will be faulty. It is up to you, the user, to make sure that this consistency is preserved.
- For the same reason, we strongly advise you not to use 'remove empty columns' or 'remove empty rows' in the tables, since from one data file to another the number rows or columns could vary.

VoxcoTab Main Menu

Menu



Note: Click on any item in the Tools menu for detailed information.