Redatam Webserver

Web Environment Reference Manual

REDATAM© is a software developed by CELADE (Latin American and Caribbean Demographic Center), Population Division of the Economic Commission for Latin America and the Caribbean (ECLAC) of the United Nations
www.cepal.org/en/topics/redatam

ECLAC/CELADE
Redatam Development Center

Table of Contents

I. Introduction

The purpose of this document is to serve as a reference manual for the development of a Redatam Webserver application through which information of a Redatam database can be requested and processed online.

The Redatam Webserver gives controlled access to the data contained in Redatam databases allowing the information to be processed in the Intranet/Internet. The default outputs permit the generation of basic indicators and tables without requiring a knowledge of Redatam command language, such as: frequencies (one or more variables), crosstabs (up to five dimensions), averages, counts, area lists, and indicators such as sex ratio, sex and age distributions, dependency ratios, and other user-defined indicators.

The outputs can be displayed from preset geographical selections and specific output areas and filters, and shown in chart, graph, or map format.

This module provides the option to access the definition of the database structure, i.e., the Redatam database dictionary (including all the variables, geographical entity variables, variables for specific entities, and even the variable categories that are characteristic of a certain entity), in order to provide the end users with all the existing documentation on the variables that are available in the database and their categories.

Additionally, it provides access to the metadata, i.e., all the information explaining and documenting the data sources, through links in HTML format. It also gives the option to access methodological manuals, questionnaires, help documents, links to corporate Web pages, and so on.

This document doesn't include the Web Server installation and startup stage, described in the Manual "*Red7 Webserver_Install ENG*". It is advisable to read that document because, in addition to the installation of the Redatam Webserver, it also contains definitions on the setup of the local server protocol (Apache or IIS).

The first chapters of this Manual describe the operation of the Redatam Webserver and its files. Then the various parts specific to each screen and the existing indicator types are covered.

II. Brief Operative Description

The main program (RpWebEngine.EXE) basically works just as shown in Figure 1. This program, guided by the controlling file, displays a list of the available databases, and the user chooses a database to work with. At that point, the *Guest* file pertaining to the selected database is triggered, taking control of the program and displaying its own listing of the processes and indicators that are available to the user.

Since this program is run by an Internet browser (Explorer, FireFox, Opera, etc.), the majority of its connections are HyperText Markup Language (HTML) pages, with the exception of the support files and the program controller (WEBSERVERMAIN.INL), which is an INL type file, mainly used in the Redatam environment. This is the only file type that should be modified by the user. No other files (especially the HTML files) should be changed, except by those users familiarized with the HTML language. This file is unique also because is the first file called by the Redatam Webserver to access our own database table of contents, therefore **Its name shall NOT be changed**. All the other files linked to the program can be renamed: to this end it's only required to modify the appropriate parameters in the Main table of contents.

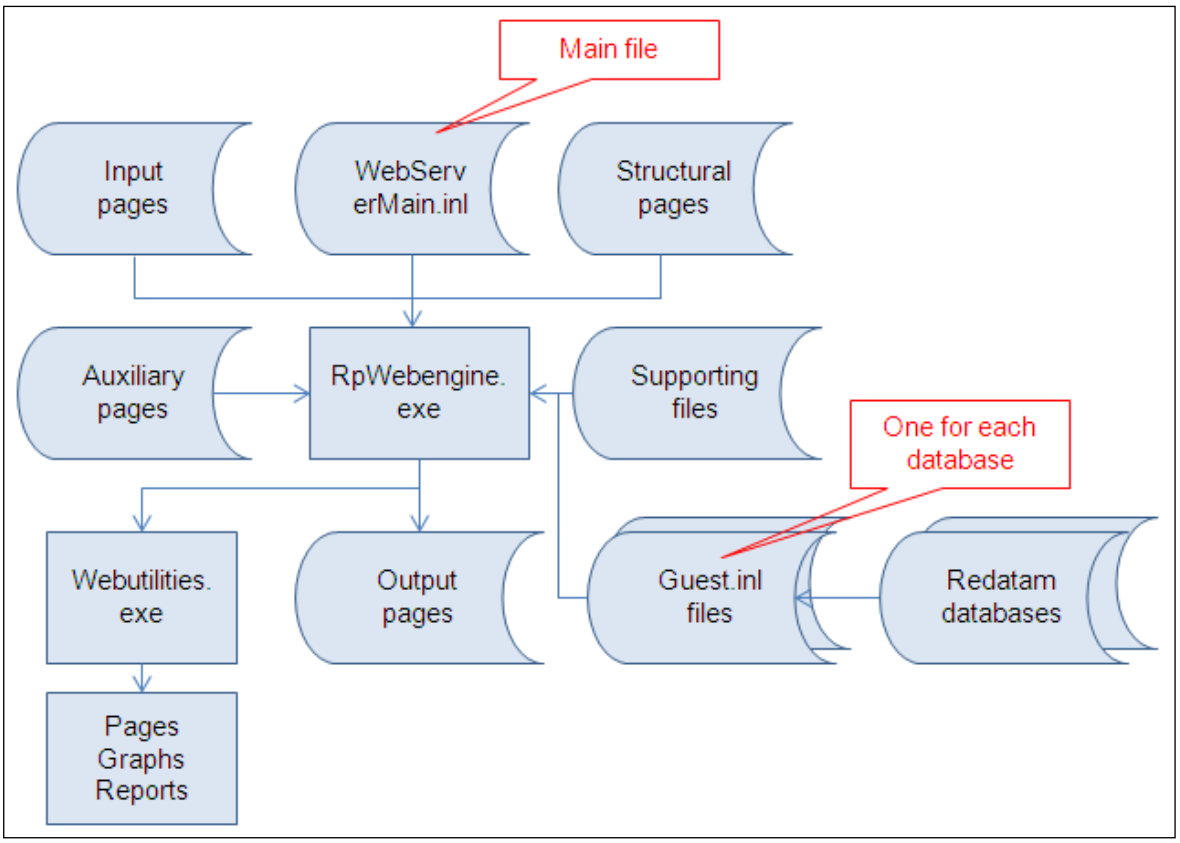


Figure 1. REDATAM Webserver Program

Important: All the HTM files used as templates to display Redatam tables and parameters should be stored in the RPSITE directory, under the private CGIBIN directory

(accessed by the alias *redatam*) where the program resides. If for any reason the HTM files must be changed, care must be employed regarding those parameters beginning with the character "#", because they are the keywords used by the WebEngine program to communicate with the pages. These parameters should not be modified or deleted from.

II.1 Structural Pages

File	Description
RpSiteMain.htm	Main Page
RpHeader.htm	*Header* Panel
RpIndex.htm	*Index* Panel
RpOutput.htm	*Center* Panel

These are four HTM type files that structure the program's screens. The first file sets a main screen divided into three panels: *Header*, *Index,* and *Output,* as shown in Figure 2 below. If this layout must be altered, or if the height of the *Header*, or the width of the *Index* must be modified, the values for "rows" and "cols" are adjusted in the first file. Each of the other three files controls the contents of the corresponding panel on screen.

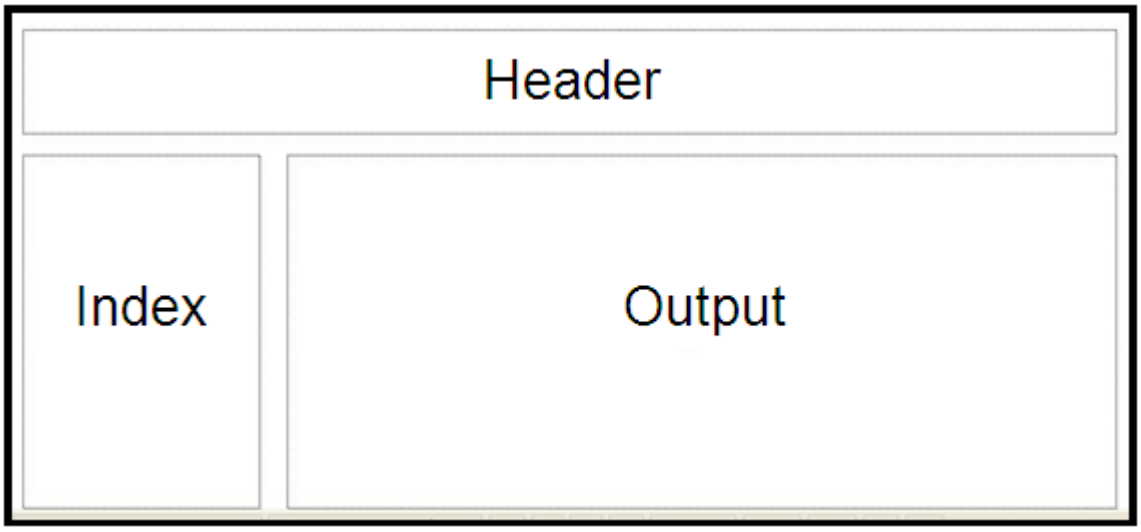


Figure 2. RpSiteMain.HTM

The Header panel is not really significant for the execution of the program; it just serves to display a header where titles, images, graphics, and other information can be shown to identify the database.

The Index panel to the left is used to display a list of the available processes to be executed by the program. The central Output panel is used as a framework, both to receive the parameters selected by the user (input pages) and to display the results from queries to the Redatam databases (output pages).

II.2 Input Pages

These pages are used to receive the selections and execute commands. Each of them is associated with a specific type of result. They can be as simple as the Dictionary screen (Figure 3) or as complex as the Fraction Indicator screen (Figure 4). Chapter VI provides a detailed description for each of those pages.

File[1]	Process (NODETYPE)
RpDepRatioInputForm.htm	Dependency Ratio
RpDicQueryInputForm.htm	Dictionary
RpEasyCrossInputForm.htm	Cruz, Average y Median
RpEasyFreqInputForm.htm	Frequency
RpEasyListInputForm.htm	Arealist
RpFractionInputForm.htm	Fraction
RpIndicatorInputForm.htm	
RpInputSPCForm.htm	Parameterized indicators
RpMultiFilterInputForm.htm	Multi Filter
RpQtsInputForm.htm	QTS
RpSexRatioInputForm.htm	Sex Ratio
RpTriRecodeInputForm.htm	Tri Recode

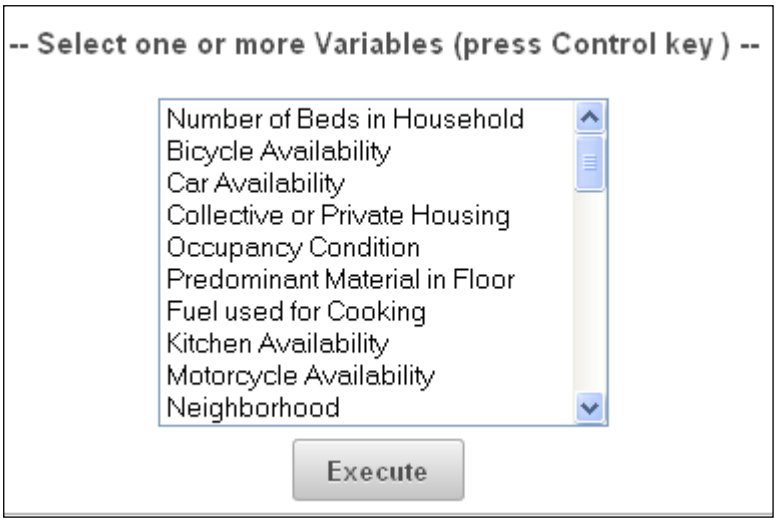


Figure 3. Dictionary Input Page

[1] Note: The name of an Input page will always contain the word "input"

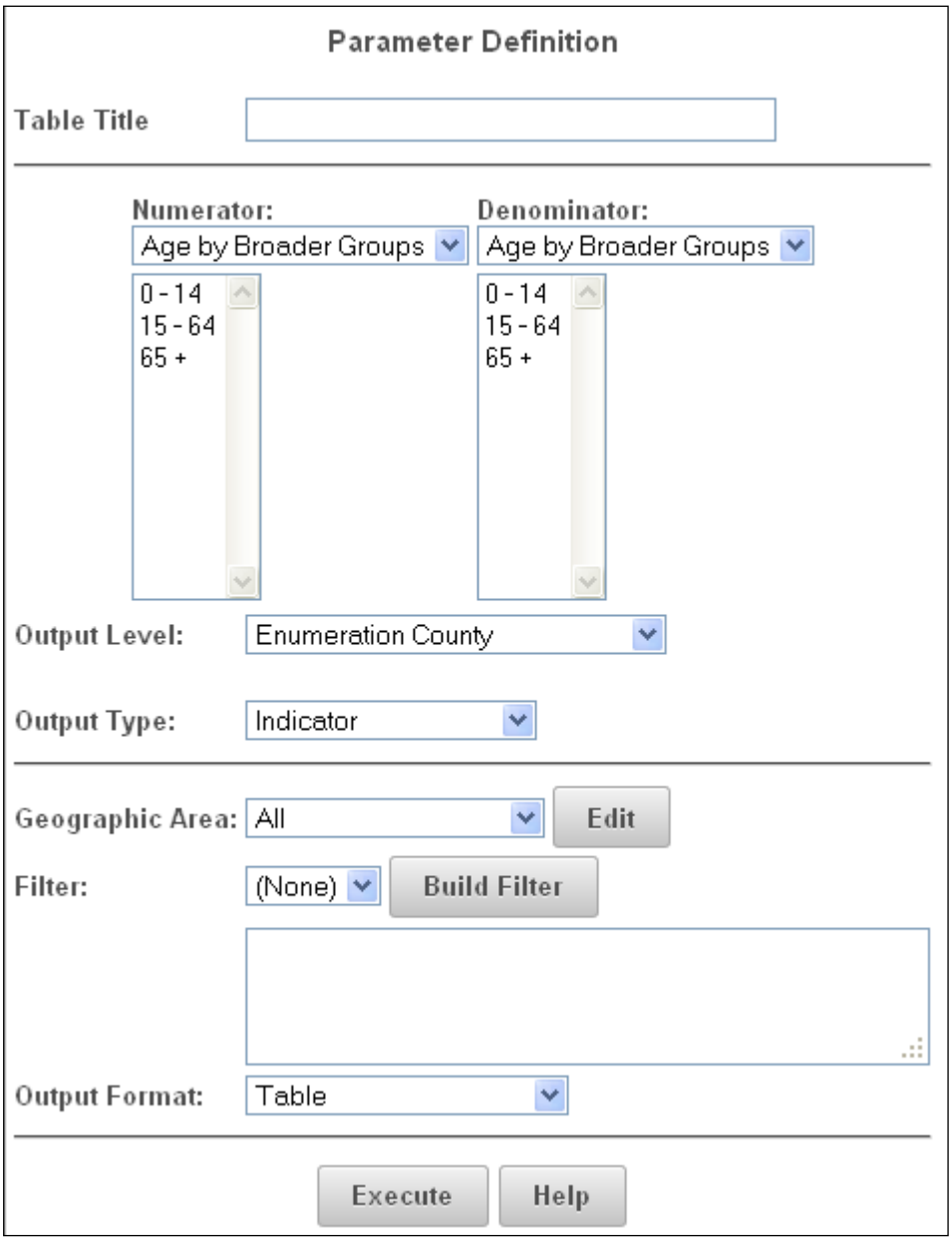


Figure 4. Input Page for Fraction

II.3 Output Pages

This program has only two output pages. The first output page is the most important, since it's used to display the results from queries. The second output page is used only to display the contents of the database dictionary. The figures below show examples of the two output pages described above, using the contents of the New Miranda database.

File[2]	Description
RpOutputForm.htm	Process Outputs
RpDictionaryOutputForm.htm	Dictionary Process Output

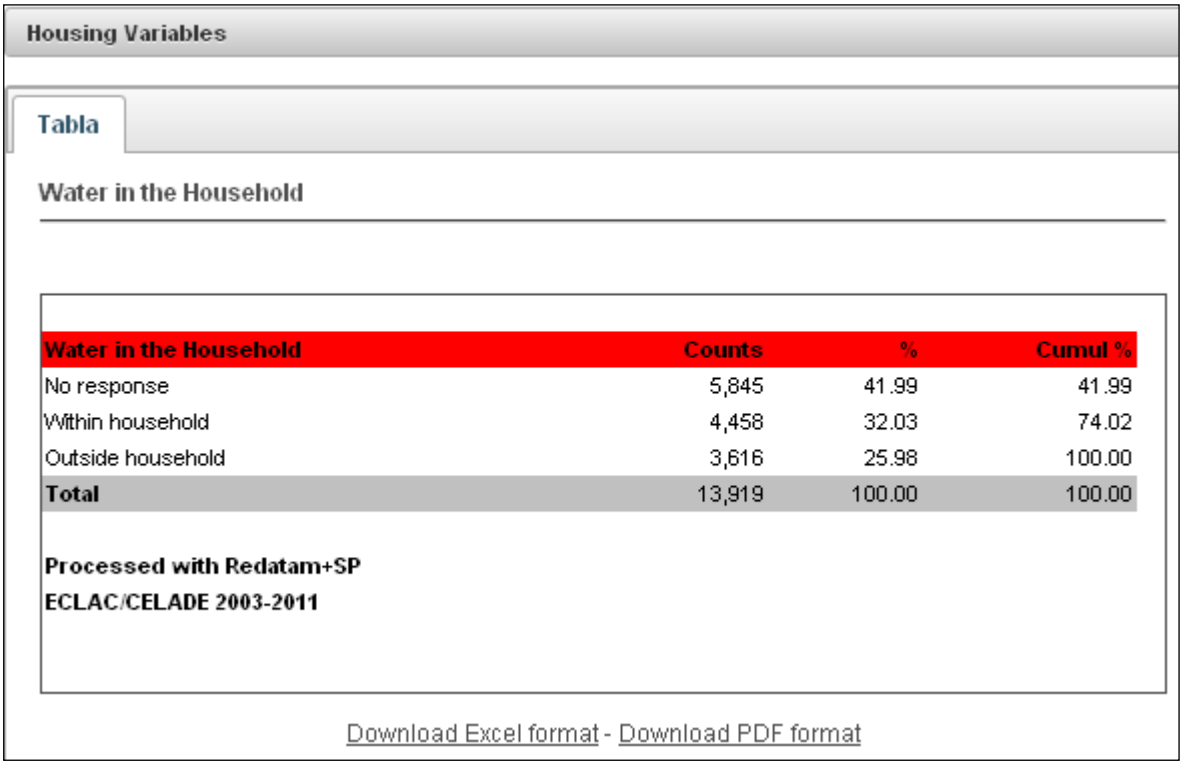
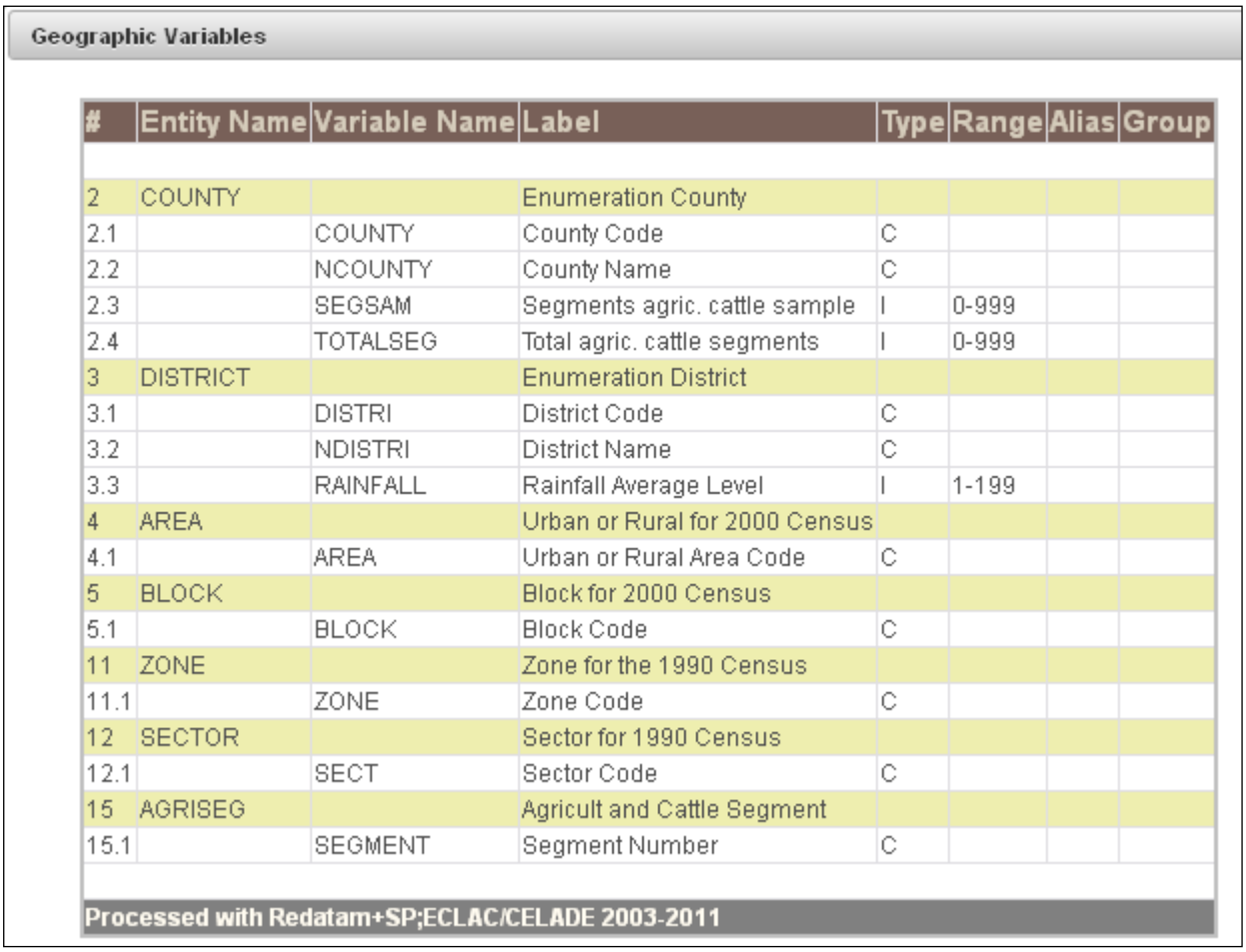


Figure 5. Frequency Result Output

[2] Note: Just like with the input pages, page names always contain the word "output".



Geographic Variables

#	Entity Name	Variable Name	Label	Type	Range	Alias	Group
2	COUNTY		Enumeration County				
2.1		COUNTY	County Code	C			
2.2		NCOUNTY	County Name	C			
2.3		SEGSAM	Segments agric. cattle sample	I	0-999		
2.4		TOTALSEG	Total agric. cattle segments	I	0-999		
3	DISTRICT		Enumeration District				
3.1		DISTRI	District Code	C			
3.2		NDISTRI	District Name	C			
3.3		RAINFALL	Rainfall Average Level	I	1-199		
4	AREA		Urban or Rural for 2000 Census				
4.1		AREA	Urban or Rural Area Code	C			
5	BLOCK		Block for 2000 Census				
5.1		BLOCK	Block Code	C			
11	ZONE		Zone for the 1990 Census				
11.1		ZONE	Zone Code	C			
12	SECTOR		Sector for 1990 Census				
12.1		SECT	Sector Code	C			
15	AGRISEG		Agricult and Cattle Segment				
15.1		SEGMENT	Segment Number	C			

Processed with Redatam+SP;ECLAC/CELADE 2003-2011

Figure 6. Dictionary Query Output

II.4 Auxiliary Pages

These pages are called auxiliary since they are used for specific purposes and are optional.

File	Description
RpExpBlder.htm	Filter Expression Builder
RpCmdSetInputForm.htm	Process REDATAM Program
RpLogin.htm	Process Protection Keyword

II.4.a Expression Building

When the user presses the button “Build Filter” (Figure 4 above) in order to define a customized process filter, the Expression Builder page (Figure 7 below) appears. This button is shown on screen whenever a FILTER type control is used (see item VIII.2) in processes.

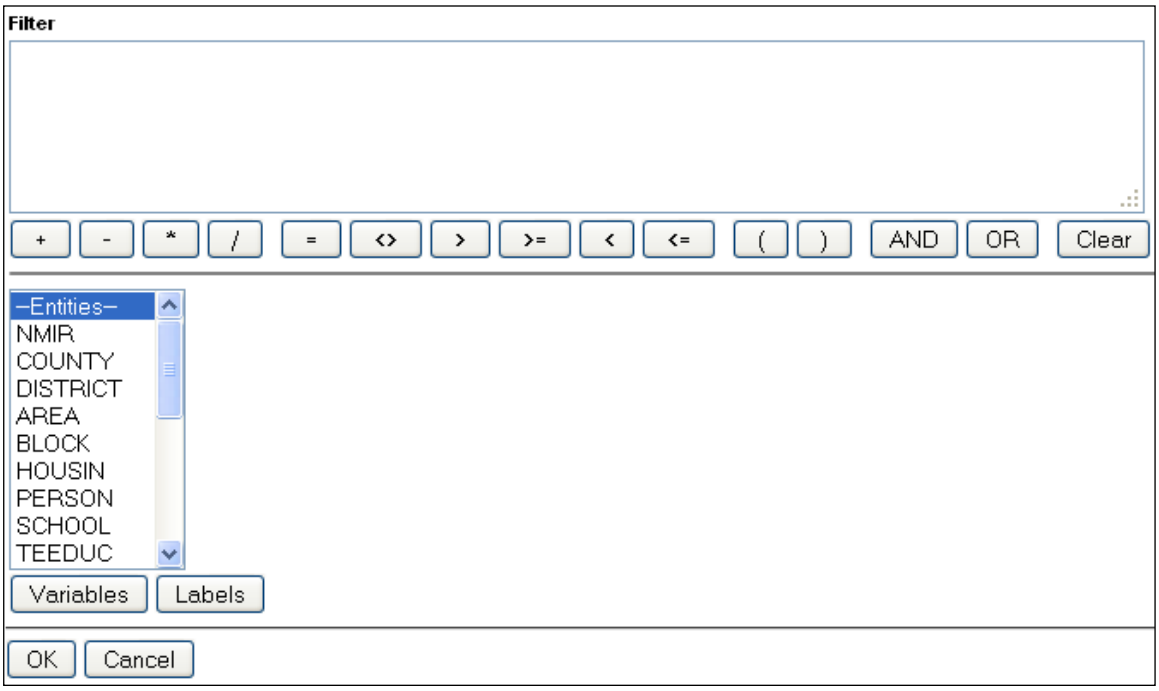


Figure 7. Filter Expression Builder

II.4.b Redatam Programming

The Redatam Programming page (Figure 8) is used to write a program directly in the Redatam language, without using preset indicators. Item VII.3, NODETYPE=CMDSET, provides more details on this process.

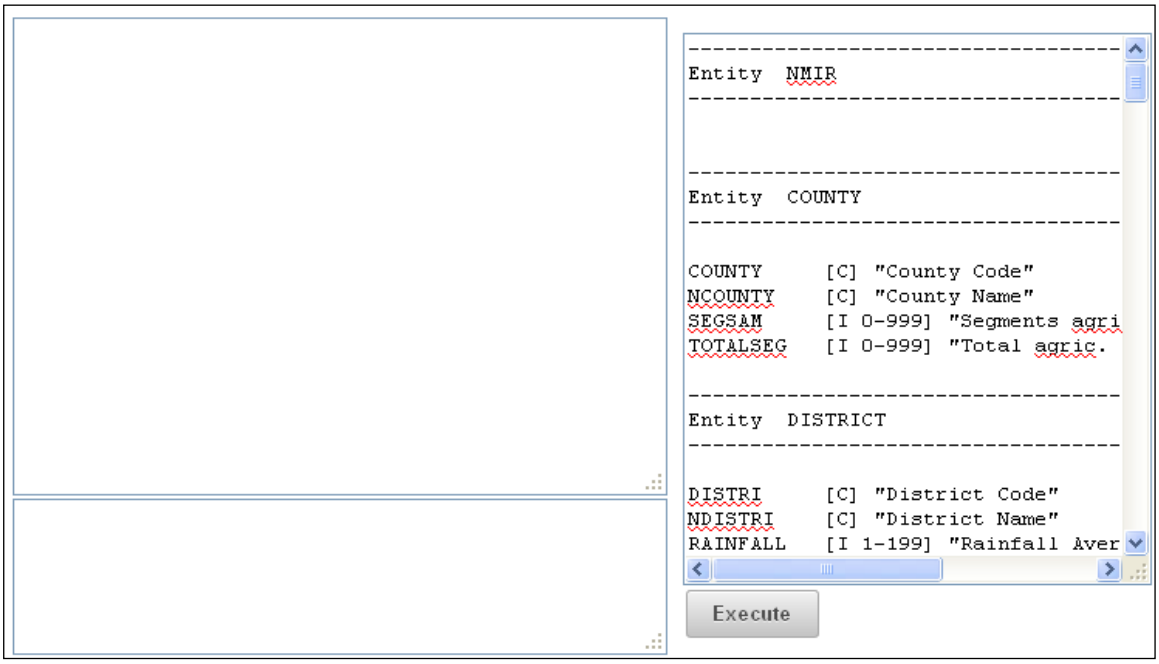


Figure 8. Redatam Programming

II.4.c Login

The Login page (Figure 9) is used to request the user id and keyword, if the database is protected. Annex VI provides more details on this topic.

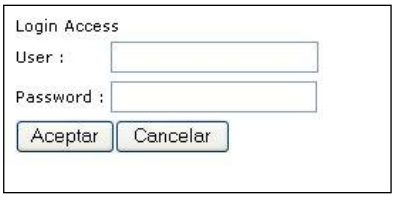


Figure 9. Login Page

II.5 Supporting Files

File	Description
RpWebEngineEng.WXD	English dictionary
RpWebEngineEsp.WXD	Spanish Dictionary

RpWebEngineEsp.WXD

These supporting files should NOT be modified. They are used only by the program. This file contains all the messages and internal texts required for the operation of the program[3]. While containing texts and messages, this file has a format proprietary to the system; it's NOT a TXT type file.

Both files must be stored in the private access mldb program directory.

II.6 Controlling File (WebServerMain.INL)

This is the most important file in the system. It must to be stored in the program directory. It is of INL type format with sections, clauses, and parameters (see a description of the INL type files in Annex I). It's called a Controlling File because, once its parameters have been set by the user, it controls the entire execution of the program. In this controlling file there are references for Guest files which, when triggered, will take control of the program and execute the tasks requested by the user. A detailed description of the contents of the controlling file can be found in Chapter III.

Figure 10 below shows an example of the application of this program. The Panel Header is located at the top, while the Panel Index to the left provides a listing of the available databases. When clicking on a line, the control (Guest) file corresponding to the database is invoked.

[3] There is a specific file for each of the languages supported by Redatam Webserver, i.e. English, Spanish, Portuguese, and French.

Figure 10. WebServerMain.INL local Table of Contents

II.7 Guest Files and Databases

There is a Guest file (Main INL file) for each available application in the server. These files, in INL format, are accountable for the interface between the user and the databases. A description of the form of these files can be found in Chapter V. Figure 11 below shows the screen created using the Guest file for New Miranda, including a Header Panel ("Nueva Miranda:…."), the Index Panel to the left, listing the prescheduled processes, and the Output Panel in the center, used for receiving requests from the users and displaying the results from processes.

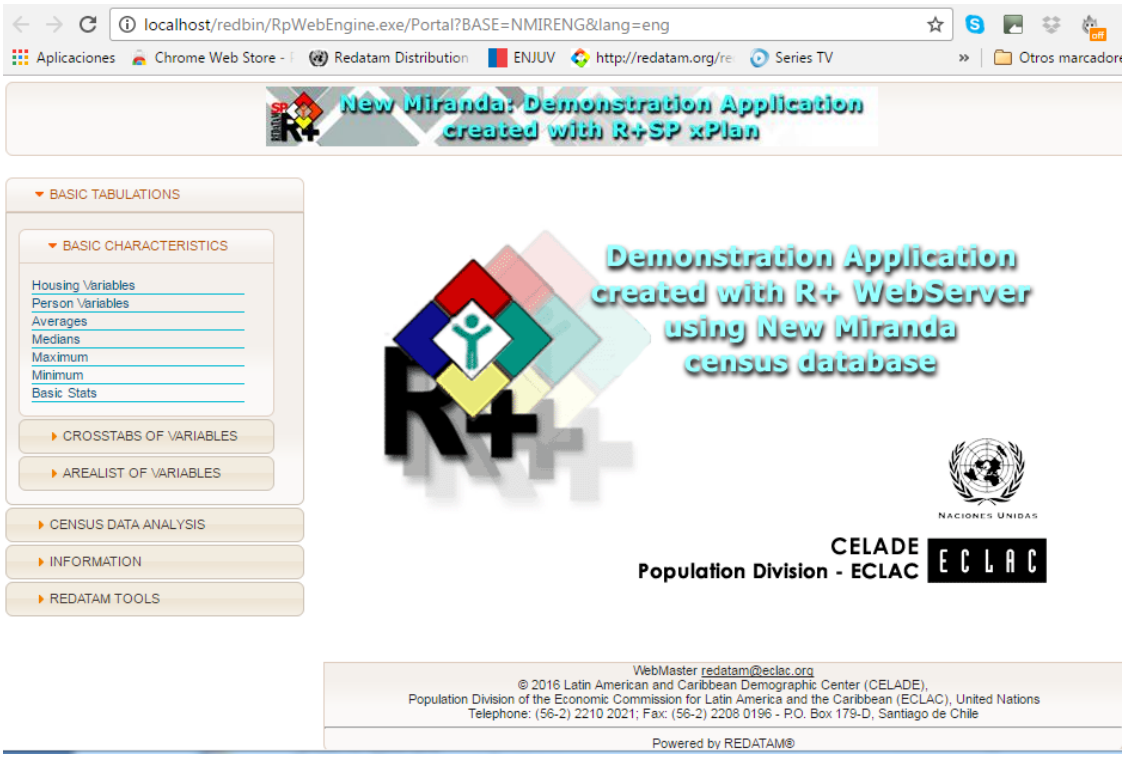


Figure 11. Execution of the Nueva Miranda Guest File

II.8 How to Call a Program

Manual "*Red7 Webserver_Install_ENG*", describes the structure of the program directories and their peripheral files. The main directory, called "Servers/Redatam", contains the INDEX.HTML and DEFAULT.HTML files, having exactly the same contents each. The use of either of these files will depend on the installed Server (either Apache or IIS; an Apache will look for the Index.html file, while an IIS will recognize the Default.html file as the default page). These pages serve as a link to invoke the Redatam Webserver program from the browser, when the following command is executed:

http://localhost/redatam/

In fact, if we look at the contents of these files, we can see that the only significant item in them is the line below, to call the Redatam Webserver program:

<FORM METHOD=POST ACTION="/cgibin/RpWebEngine.exe/PortalAction?">

This command executes the RpWebEngine.exe program, and the call, using the contents of the WebServerMain.INL file, results in the display of the screen in Figure 10 above, i.e., a list of the databases available in the site. However, if you want to directly call a certain database without entering the database list, you may just execute a command such as in the example below. In that command the Nueva Miranda database is executed in Spanish, as referenced by the "NMIRESP" parameter. That parameter is the name of the section in the WebServerMain.inl file by which the Nueva Miranda database is known.

http://localhost/cgibin/RpWebEngine.exe/PortalAction?&MODE=MAIN&BASE=NMIRESP&MAIN=WebServerMain.inl

This type of command may be placed in a shortcut of your Internet Browser (IE Explorer, or other browser such as Firefox) as a direct connection to the database through its Guest file, without having to use the RpWebEngine.exe input screen.

III. WebServerMain.INL Programming

Just like any INL file (see Annex I), the WebServerMain.INL file is made up of Sections and Clauses. It has a single mandatory section ([STRUCTURE]), which is generally located at the beginning[4]. This section has two basic features: a) to define the contents of the Index panel; and b) to define the names of the files containing the structural pages, input pages, auxiliary pages, output pages, and supporting files mentioned above in the program description (Chapter II).

If appropriate, the file could be separated into several #includes, e.g., the WebServerMain.inl with the [STRUCTURE] section and calls using #includes to the node definition part.

Clause	File
Structural Pages	
HTMLSITEMAIN	RpSiteMain.htm
HTMLHEADER	RpHeader.htm
HTMLINDEX	RpIndex.htm
HTMLOUTPUTEMPTY	RpOutput.htm
Output Pages	
HTMLOUTPUT	RpOutputForm.htm
HTMLDICFILE	RpDictionaryOutputForm.htm
Input Pages	
HTMLDICQUERY	RpDicQueryInputForm.htm
HTMLEASYCROSS	RpEasyCrossInputForm.htm
HTMLEASYLIST	RpEasyListInputForm.htm
HTMLEASYFREQ	RpEasyFreqInputForm.htm
HTMLSEXRATIO	RpSexRatioInputForm.htm
HTMLDEPRATIO	RpDepRatioInputForm.htm
HTMLFRACTION	RpFractionInputForm.htm
HTMLQTS	RpQtsInputForm.htm
HTMLTRIRECODE	RpTriRecodeInputForm.htm
HTMLMULTIFILTER	RpMultiFilterInputForm.htm
HTMLINPUTSPC	RpInputSPCForm.htm

[4] You may obtain more benefits from this Chapter by editing the file in your preferred text editor.

Auxiliary Pages	
HTMLEXPBLDER	RpExpBlder.htm
HTMLCMDSET	RpCmdSetInputForm.htm
HTMLLOGIN	RpLogin.htm

The clauses above don't have default values; hence ALL of them must be reported.

III.1 Defining the Index

The Index (left panel on screen) is used to display a list of the available actions and databases in the server. These actions are represented in the index panel by a list of nodes.

Clause	Type	Comment
NODES	Integer	Maximum number of nodes.
NODE$_i$	Section Name	Name of a section or asterisk (“*”). *i* Ranges from 1 to NODES.
TITLE	Text	Process title in the OUTPUT page. If there is no such clause, then the title will be the contents of CAPTION. In order to delete a title from the output, this clause must be set to null (TITLE=).

First, the maximum number of nodes must be reported, using the NODES clause. Example:

```
NODES=100
```

This means that the system will accept up to 100 nodes, and will look for data in the NODE1, NODE2,…, NODE100 clauses.

It is not required that all clauses in the interval 1:100 exist. The maximum number of nodes is arbitrary, but it should not be very high so as not to affect the efficiency when looking for nodes (e.g., if NODES=99999 is used, then the system will look for all possible nodes between 1 and 99999).

Then, the clauses of the nodes must be reported. These can be of two types: a) a text string, that would be the name of a section in the file; or b) an asterisk (“*”), used only to display a blank line. Example:

```
NODE1=TITLE
NODE2=*
NODE10=NMIRGROUP
NODE11=*
```

TITLE & NMIRGROUP should be the names of sections to be reported later on
Nodes 2 & 11 display blank lines in the Index

Those sections reported in the nodes in the WebServerMain.INL file can ONLY be of the following types (NODETYPE):

NODETYPE	Purpose
GROUP	To define a node subgroup, to be displayed in the Output portion (to the right of the Index)
DisplayHTML	To display a HTML page in the output
BASE	To define a database that is available in the Server
*	To display a caption with no features

III.2 Additional Information

In addition to reporting on the structure of the index of available services and defining the names of HTML page files, the WebServerMain.INL file has other clauses such as:

Clause	Type	Comment
USERCONTROL	RWord YES/NO	Proposed for future versions.
GROUPALIGN	RWord LEFT CENTER RIGHT	Allows displaying the nodes of a GROUP aligned according to the value that is defined here. CENTER is the default value
SERVERTIMEOUT	Integer	Allows defining a maximum value for a Redatam process in order to limit the runtime of a process, since servers (especially IIS servers) have a defined timeout for running a cgi application. Then the application is terminated and RpWebEngine.exe continues the processing.
HTMLPATH	Directory Name	Defines the path for the directory where the definitions of the application's HTML pages reside
PORTALLEFTIMAGE	Filename	Name of the file containing the figure that will be displayed to the left of the HEADER panel.
PORTALTITLE	Text	Use a text as title in the Portal[5]
PORTALSUBTITLE	Text	Use a text as subtitle in the Portal
PORTALBACKGROUNDHEADERIMAGE	Filename	Background image for the Header panel. They may be .gif, .jpg, or .bmp files, and they must be in the directory pointed to by HTMLPATH
PORTALBACKGROUNDINDEXIMAGE	Filename	Background image for the Index panel

[5] Texts for Title and Subtitle may include the
 character set, meaning a line break

PORTALBACKGROUNDOUTPUTIMAGE	Filename	Background image for the Output panel

Example:

Contents

```
USERCONTROL=YES
GROUPALIGN=LEFT
SERVERTIMEOUT=999
HTMLPATH=RpSite\
PORTALLEFTIMAGE=RpSP80.JPG
PORTALTITLE=CEPAL / CELADE
PORTALSUBTITLE=Test Site<br>List of Available Databases
PORTALBACKGROUNDHEADERIMAGE=
PORTALBACKGROUNDINDEXIMAGE=BKLADRILLO.JPG
PORTALBACKGROUNDOUTPUTIMAGE=
```

Lastly, the WebServerMain.INL file may contain a glossary section (GLOSSARY), which will determine some default texts for the program controls (see item IX.3). In summary, the contents of the file could be laid out as follows:

[STRUCTURE]
Definition clauses for pages (HTM…)
Definition clauses for Index (NODES, NODE1, …)
Additional Clauses (GroupAlign, etc.)

[SECTION_1]
…
[SECTION_N]

[GLOSSARY]

IV. Structure and Definitions of the Guest files

The Guest files are also INL type files (see Annex I), and they are used to program the access to the databases. “Guest” is a generic name. Each of these files is indexed in the WebServerMain.INL by the INL clause NODETYPE=DATABASE. For example, the Nueva Miranda database is triggered by the NMIRANDACENSO_ESP.INL file (for Spanish). A single Guest file can trigger a database, but the same database can be triggered by more than one Guest file. I.e., the Guest files work as “views” or inputs into a database.

IV.1 Processes

A Guest file contains various processes, which are parameterized (or programmed) to go into the database. Thus it can be said that a Guest file is programmed, or rather that the Guest files are programs in INL language (see Chapter V). There are several types of processes, each with a different objective and therefore with different parameters. For example, the “Dictionary” process is used to list the database variables, while its parameters are used to choose the options (only geographical variables, or variables for an entity, or categories for some variables, etc.) that will be executed. In turn, the “Frequency” process is used to run a simple frequency on the variables in the database. Its parameters would be, e.g., the entity or a list of variables.

IV.2 Controls

The parameters in a process are called controls. Each process has specific controls (called proprietary controls, as mentioned in the paragraph above), and common controls, that can be used in more than one process. For example, the processes may use a filter to select the cases, or a geographical selection, or even the definition from the weight factor to be used. All of those controls are common to virtually all the processes. Proprietary controls for each process shall be defined along with the process itself (see Chapter VI), while common controls will be defined in Chapter VIII.

Some controls are used to account for lists, whether they are entities, variables, or variable categories. These lists may be displayed on screen in one of two ways: a) using boxes (“combobox”) where only one item is shown at a time, and only this item (whether an entity, a variable or a category) may be selected; or b) using a list of items, in which case more than one item may be selected.

Let’s take a look at the CRUZ type process. One of the controls in this process is a list of column variables, i.e., a list of the variables among which the user will choose a variable to be shown in the column when the process is executed. A clause for the number of items must ALWAYS be defined for such boxes and lists. Then a clause for the contents of each item must be defined. As a rule, this is done using as a prefix the name of the CONTROL, and then adding the letter ‘N' for the item clause, and a sequential number for the contents clauses. For example, for the row variables (‘ROW '), the clauses ROWN are used for the number of items, and ROW1, ROW2, ROW3, etc. are used for the individual items. Example of a list of column variables:

Contents	Comment
[CRUCVIV]	CRUZ type process.
CAPTION=Housing	
NODETYPE=CRUZ	
COLN=5	Number of entries in the list of variables.
COL1=(none)	The first entry is text in brackets. By default, any texts in brackets mean a NULL entry, i.e., no variable. This works for all the lists of items in the Redatam Webserver.
COL2=PERSON.SEXO	Other entries in the list, with the names of the variables in the database.
COL3=PERSON.ECIVIL	
COL4=PERSON.PARENT	
COL5=PERSON.CURSO	

However, if the list will appear in a number of processes, then it would be preferable to create the list in a separate section, and then have each process point to that section using the COL clause, which will have the name of the other section as if it were a subroutine or function. Thus, a single list of variables can be used in several controls, and changed if required at a single point (the section where it's written). The example above would look like this[6].

Contents	Comment
[CRUCVIV]	CRUZ type process.
CAPTION=Housing	
NODETYPE=CRUZ	
COL=COL_PERS1	
…	Elsewhere in the INLs...
[COL_PERS1]	
COLN=5	Number of entries in the list of variables.
COL1=(none)	The first entry is not text in brackets. By default, any texts in brackets mean a NULL entry, i.e., no variable. This works for all the lists of items in the Redatam Webserver.
COL2=PERSON.SEXO	Other entries in the list, with the names of the variables in the database.
COL3=PERSON.ECIVIL	
COL4=PERSON.PARENT	
COL5=PERSON.CURSO	

On the other hand, if the list contains only one item and you do not want to display that item on screen, then the generic clause 'FIXED ' is used. This clause is made up of the name of the Control and the word FIXED. This clause takes precedence over the others. Example:

COLFIXED=PERSON.SEXO	The columnar variable is always PERSON.SEXO. In that case the columnar control is not displayed on screen. Even when reported as COLN=5, or COLN=COL_PERS1, the system would not take this list into account.

IV.3 Properties

[6] Thus all list controls will be represented in that document, by referring to another section.

Each of the controls, either proprietary or common, is a set of clauses that are grouped together as a “control” in order to facilitate their description. In general, a control is an image in the INPUT screen, e.g. an execution button, or a box containing a list of variables. Each control has properties that shape the control. For example, its caption[7]. But these properties are not the most significant. They are called Peripheral or Cosmetic Properties because they don't directly affect the control function itself. In the case of a box of variables, such as the “Row” control in the “Frequency” process, what really matters is the list of variables that will populate that box when the program is run. These are called Main Properties and are defined within each control. Peripheral properties are defined in Chapter IX. Generally, they are reported in the nodes of the NODESTYLE type (see item VII.10) in order to somehow ‘clean' the programming of processes, leaving in the process itself only the most important control clauses, or main properties.

When referred to controls these clauses are also named following a rule, using the name of the control as a prefix. For example, ROWCAPTION.

IV.4 Structure

If we look at a Guest file from the point of view of an INL file, we see that the Guest file has the same contents in terms of sections and clauses. Structurally however, we’ll see that a Guest file contains processes that are in turn made up of controls. These controls may be specific controls or common controls, as in the figure below.

```
      INL View
[Section1]
      Clause11
      Clause12
      Clause13
[Section2]
      Clause21
      Clause22
…
```

```
      Structural View
Process1
      Proprietary Control
      Common Control1
      Common Control2
      Common Control3
Process2
      Proprietary Control
      Common Control4
      Common Control5
…
```

Guest files are triggered when users click on their names in the list in the WebServerMain Index. Then the Guest file takes control of the program and “overrides” the WebServerMain structure with its own structure, using the same spaces, which means that a Guest file will also manage an Index, a Header and an Output.

Guest files can also be directly run by the Internet browser (IE Explorer), without going through the Redatam Webserver screen. Actually, what is run is the RpWebEngine.exe program, using as a parameter the name of the section in the WebServerMain.INL file that points to the Guest file. For example, the file contains a section for the Nueva Miranda database.

[NMIRESP]
CAPTION=Español

[7] In Spanish we use the term “cabezal” for CAPTION in order to differentiate it from the “title” property that exists in some Processes and Controls

INL=RpBases\NMIR\NMIRANDACENSO_ESP.inl
PAGETYPE=BASE

To directly call that Guest file, the following command is used:

http://localhost/cgibin/RpWebEngine.exe/PortalAction?&MODE=MAIN&BASE=NMIRENG
&MAIN=WebServerMain.inl

V. Programming of Guest Files

In organizational terms, Guest files are very similar to WebServerMain. They also have a main mandatory section called [STRUCTURE], and then a section for each process. Just like WebServerMain, a Guest file may also contain some special sections.

Instead of starting from an empty file without any clauses or sections to program a Guest file in an application, it's preferable to take an existing Guest file, such as the Nueva Miranda (NMIR) example included with the program and perform the required modifications.

V.1 Section [STRUCTURE]

This section has several functions, such as setting the process Index, connecting to the database, accepting of the #includes, and other special parameters.

V.1.a Index

This section has the primary purpose of providing the contents for the index panel on screen. For this purpose, this section uses the same clauses as the WebServerMain Index (see item III.2). For example:

Contents	Comment
`NODES=8`	Number of nodes in the Index
`NODE1=POBYVIVIENDA`	Node 1 points to a section in the file (see below)
`NODE2=ESTRUCTURA`	Node 2 points to another section
`NODE3=ADULTOMAYOR`	Additional nodes
`NODE4=FECUNDIDAD`	
`NODE5=EDUCACION`	
`NODE6=DICGROUP`	
`NODE7=PROGGROUP`	
`NODE8=HELPGROUP`	
`[POBYVIVIENDA]`	Section pointed to by Node 1
`NODETYPE=STRUCTURE`	Type of node, a process set
`CAPTION=POPULATION & HOUSING`	
`NODES=3`	Number of nodes in the set
`NODE1=CRUCVIV`	Node 1 points to another section (below)
`NODE2=CRUZ1`	
`NODE3=AREALIST1`	
`[CRUCVIV]`	Section in Node 1 POBVIVIENDA. This is a
`NODETYPE=CRUZ`	Cruz Process (variable crossing)
`CAPTION=Housing`	
`NODESTYLE=CRUZ1.DEFAULT`	
`...`	

Nodes refer to other sections in the file, which may be of one of two types: a) a process; or b) a collection of processes, with a list of nodes. This node type is called a STRUCTURE. This is something of a directory, and may contain files or other directories

(that in turn may contain subdirectories, etc.) In the example above, Node 1 (POBVIVIENDA) is a STRUCTURE node with an additional list of nodes, while Node 1 in this new list is a process (NODETYPE=CRUZ).

V.1.b Connection to the Database

A second function of the STRUCTURE section is to connect the Guest file to the database. This is done through the DATASETS=1 clause, which refers the control to the [DATASET1] section that, in turn, calls the database. The following example shows how that connection works.

Contents	Comment
`DATASETS=1`	A list of databases

`[DATASET1]`
`NODES=1`
`NODE1=DATABASE1`
	Intermediate section with a list of databases containing only one item

`[DATABASE1]`
`NODETYPE=DATABASE`
`NAME=%INLPATH\BaseR\NmirEsp.dic`
	Section defining the database
	Node type for the database
	Name of the database dictionary, using the directory marker %INLPATH

While at the time the Redatam Webserver application accepts only one database in a Guest file, the connection structure is designed to accept multiple databases. For this reason the [DATASET1] section contains a list of nodes with only one item, which is the name of a different section (DATABASE1]). In this section there is the NODETYPE=DATABASE that “finally” points to the database as such (see item VII.5). You may use directory markers in the names of the files (see Annex II).

V.1.c Additional Functions

The STRUCTURE section defines a range of other parameters required by the program, such as the name of the geographical selection set that is available to the user (SELSET clause), the number of DEFINEs to be read (DEFINES clause), the number of MAPs to be read (MAPS clause) and a list of its items (MAP1, MAP2 clauses, and so on), the background images for the pages (just as in the WebServerMain), and more. A complete list can be found in item V.1.e.

V.1.d INCLUDEs

In addition to setting a process index, the STRUCTURE section has other functions. #INCLUDEs are reported in this section. #INCLUDEs are used to call other files that will be used during execution. #INCLUDEs are not mandatory, but they may facilitate the organization of processes in the Guest files. If there are any #INCLUDEs, it’s crucial that these commands are the last clauses in the section. Otherwise, subsequent clauses will not be taken into account; since what the #INCLUDEs do is to take the contents of the file mentioned in the #INCLUDE command and place them in the Guest master file. If there are any section names in the #INCLUDE files, then an additional section is begun, and it

is as if the clauses that were in the STRUCTURE section after the #INCLUDEs were transferred to that section.

Example: Each of the listed files will have its contents associated to the Guest file.

Contents	Comment

`#include ESP\Preferences_ESP.inl`
`#include ESP\DataSels_ESP.inl`
`#include ESP\Panels_ESP.inl`
`#include ESP\Styles_ESP.inl`
`//********  end of the STRUCTURE section`
	INL command files of to be added at the end of the Guest file. You can have as many as required.

V.1.e STRUCTURE Clauses

Clause	Type	Comment
DEFINES	Integer	Number of commands for the definition of derived variables that must be read by the program. Example: **DEFINES=18**
MAPS	Integer	Number of map entries that must be in the file. These entries correspond to map sections and, in order to be defined, there must be MAP_i clauses (see below). Example: **MAPS=5**
MAP_i	Section Name	Where *i* ranges from 1 to the value of the MAPS clause. It is the name of a NODETYPE = section. Example: **MAP1=MAPCOMU**
SELSET	Section Name	Name of a section with a list of available geographical selections. Example: **SELSET=SELSET1** Selections may also be defined and organized in more than one set of available selections, by using the clauses SELSETS, SELSET1, SELSET2, etc. E.g.: **SELSETS=2** **SELSET1=SELREGION** **SELSET2=SELCOMUNA** In that case, within each Process it must be reported which of the selection sets will be used and therefore displayed in that node.
DEFAULTSELSET	Integer	Number of the selection section in the selection set that will be used by default if the SELSET clause is not reported in the process. Example: **DEFAULTSELSET=1**

LASTSELECTION	Integer	Number of the entry in the list of available selections that will be shown by default in the program screens. Example: **LASTSELECTION=5** That means that selection Number 5 in the section pointed to by SELSET will always be the selection that is shown in the screens when there is a geographical selection box.
FOOTNOTES	Integer	Number of footnote sections that should be in the program.
DEFAULTFOOTNOTE	Integer	Number of footnote sections that will be shown in the program outputs.
PORTALLEFTIMAGE **PORTALTITLE** **PORTALSUBTITLE** **PORTALBACKGROUNDHEADERIMAGE** **PORTALBACKGROUNDINDEXIMAGE** **PORTALBACKGROUNDOUTPUTIMAGE**		These clauses are exactly those in the WebServerMain. If not reported, then they will assume the values defined in the WebServerMain.
HTMLOUTPUTEMPTY		Example: **HTMLOUTPUTEMPTY=ESP\RpOutput_ESP.htm**
HTMLSITEMAIN		Example: **HTMLSITEMAIN=RpSiteMain.htm**
MAXLABELSIZE		Box size for displaying variables (“combobox”) in the Input panel. Example: **MAXLABELSIZE=60**

V.2 Priority Order for Clauses

The data transferred to programs may be sourced from several sections of a Guest file, such as the Glossary section, the Preferences section, the process sections themselves, the style sections, or the sections of controls invoked by processes. Sometimes it’s appropriate to have the data provided for general use (such as box sizes) in a style section for all the processes that may use that form (e.g., a CRUZ process), and change that value when required for selected processes. Or this could be used even for some specified text (such as the CAPTIONs or BUTTONSUBMITs) that could be reported in the Glossary but that sometimes require specific modifications.

In these cases, if the data from an existing clause is repeated in more than one section, then the priority order for obtaining of the clause value will be as follows: first, data is looked for in the main process section, i.e., the section containing one of the NODETYPE that have been defined as primary. Then, data is looked for in the section specific to that clause (e.g., a ROWCAPTION will be looked for in the section defined by the ROW clause, if this is what was used for referencing the sections). Then the style section follows, if this section was reported in the main process. Then the program will look for data in the Preferences section, and lastly in the Glossary section, where it will primarily look for the texts of the common clauses.

VI. Main Processes

These are the execution processes as such, which will produce some type of output from or processing of the database. The other processes are supporting or less significant tasks, and are called NODES. Each of the main processes is defined by a specific NODETYPE. Processes are in alphabetical order.

Process	Output	Objective & Description
AREALIST	AREALIST	A list of areas at a specified level, with the distribution of frequencies of selected variables.
AVERAGE	AVERAGE	A chart listing averages for a selected variable, and controlled by row and column variables.
CRUZ	CROSSTABS	A variable crossing up to 5 dimensions.
DEPENDENCY RATIO	AREALIST	Dependence ratio between the population of 'supported' age divided by the 'supporting' population.
DICTIONARY	DICTIONARY LIST	Contents of the data dictionary of the database.
FRACTION	AREALIST	Ratio indicator for dividing a variable category by another variable category.
FREQUENCY	FREQUENCY	Distribution of frequencies for one or more variables. This is similar to the Cruz process, but in a single dimension.
INPUTSPC	ANY	Parameterized indicators on a REDATAM base program (template).
MEDIAN	MEDIAN	A chart medians averages for a selected variable, and controlled by row and column variables. Also used to calculate the Minimum and Maximum values for a variable.
MULTIFILTER	CROSSTABS & AREALIST	Specific for databases with data added generally with indicators already calculated. Charts and arealists for mapping indicators with various segregations selected by users such as urban/rural, sex, age, poverty status.
QTS	AREALIST	Ratio indicator between categories of a user-selected variable, and total of cases.
SEXRATIO	AREALIST	Population's Gender Ratio (men divided by women).
TRIRECODE	CROSSTABS & AREALIST	Specific for databases with added data.

Basic Processes:

AREALIST, AVERAGE, FREQUENCY, CRUZ & MEDIAN are processes that reproduce the primary functions of REDATAM.

In theory, the FREQUENCY process should not be required, since it's very similar to CRUZ. Both distribute case frequencies, but FREQUENCY is limited to a single dimension. The advantage of using FREQUENCY is that this process may generate frequencies from more than one variable at a time and also, it can pass

as a parameter an entity name without having to pass a (sometimes extensive) list of variables.

This group of basic processes could also include the DICTIONARY process, since in reality it only displays the contents of the database dictionary.

Indicator Processes

DEPENDENCY RATIO, FRACTION, QTS, and SEXRATIO are ratio processes that generate a list of areas with the results from the division of a counter by another counter.

DEPENDENCY RATIO uses the age variable to create such counters. SEX RATIO uses the sex variable to define a numerator and denominator.

QTS takes up any variable and uses as numerator a combination of categories from that variable, and as denominator the total number of cases that have responded to that variable. For example, QTS is used for creating age pyramids, schooling rates, etc.

FRACTION is the most generic among these indicators, since it takes up 2 variables, one as numerator and the other as denominator, and takes from each a combination of its categories.

Aggregate Processes

MULTIFILTER and TRIRECODE are called 'added processes' because both work with databases which already have values from calculated totals, i.e., their data is already at the geographical levels.

MULTIFILTER can select the values to be displayed and also create filters out of cases, while TRIRECODE only selects cases, without filters.

VI.1 Common Clauses

Some clauses are common to all processes, such as CAPTION and TITLE. Therefore, these will be defined only once, but are valid for all processes.

Clause	Type	Comment
CAPTION	Text	Text to be displayed in the Index. For these execution processes, this is also the text that appears in the OUTPUT page when the process is selected in the INDEX list.
TITLE	Text	Process title in the OUTPUT page. If there is no such clause, then the title will be the contents of CAPTION. In order to delete a title from the output, this clause must be set to null (TITLE=).
NODESTYLE	Section Name	Name of a section which will contain the common clauses to be used by the Processes invoking that

		clause. Example: **NODESTYLE=FREQUENCY.DEFAULT**
WEBMASTER	Text	Text that will be shown in the process footers. In general the text defined in the Glossary is used, but if a specific text is required for each process, then a clause can be defined for each. Or, to override the clause, use null text (WEBMASTER=)
COPYRIGHT	Text	Same as for WEBMASTER
UNIVERSE	Text	To select and the define the cases that are relevant to the process. For example, to filter only 15 year-old and above women for a fecundity process, or 65 year-old and above persons for a senior indicator. For details, see item VIII.6 UNIVERSE & ALTFILTER.
ALTFILTER	Text	Same as UNIVERSE, for those cases where the variable involved in the expression is a derived variable.
HTMLHELP	Filename	If this clause exists, then a button is shown with the contents of the BUTTONHELP clause (which, in general, can be found in the GLOSSARY). If the user presses this button during runtime, then the program shows the contents of that clause's file, which generally is a HTM file. Example: **HTMLHELP=/redatam/RpHelp/procesamientodeindicadores.htm**
PROCESSTITLE	Text	

VI.2 Common Controls

Some controls are common to all processes, such as TABLETITLE and AREABREAK. Therefore, these will be defined only once, but are valid for all processes.

Clause	Type	Comment
TABLETITLE	Text	Title to be shown in the table results.
ABK	Section Name	Section name that will contain the AREABREAK clauses. Example: **ABK=ABK_1** In this case [ABK_1] is a declared section. Alternatively you may use a fixed areabreak with the name of an entity. Example: **AREABREAKFIXED=COMUNA**
FOOTNOTE	Text	Footnote.
TALLY	Section Name	Section name that will contain the Tally Control clauses to be used as increments for calculating the table. Example: **TALLY=SECTALLY** [SECTALLY] is a declared section. Alternatively you may use a variable as fixed increment. Example: **TALLYFXED=PERSONA.HIJOS**
WEIGHT	Section Name	Section name that will contain the Weight Control clauses, in case that the database may have several

		different weights. Example: **WEIGHT=SECPESO** In this case [SECPESO] is a declared section. Alternatively you may use a variable as fixed weighting factor. Example: **WEIGHTFIXED=VIVIENDA.PESO** In this case the WEIGHT control is not shown on screen.
FILTER	Section Name	Section name that will contain a list of filters so that the user may choose (or not) one of them. **FILTER=FILTER_MULTI** Alternatively you may use an expression as fixed filter. Example: **FILTERFIXED=VIVIENDA.ESPECIE=1** In this case the FILTER control is not shown on screen.
FORMAT	Section Name	Section name that will contain a list of output formats (either table, graph, map, or Redatam program), so that the user may choose one of them. Example: **FORMAT=FORMAT_1**
SELSET	Section Name	Section name that will contain a list of geographical selections so that the user may choose (or not) one of them. Example: **SELSET=SELSET_1**
MAP	Section Name	Section name that will contain a list of maps so that the user may choose (or not) one of them. Example: **SELSET=SELSET_1**
GRAPHCROSS	RWord	The type of graph to be used in the output. These include: PIE BAR LINE MULTIBARSERIE AGEPYRAMID HORIZBARSERIE LINESERIE BARSERIE RADAR DEFAULT (by default, it takes on the preset value for the type of result to be displayed)
PCT	Section Name	Section name that will contain a list of potential percentages (row, column, total, or absolute value), so that the user may choose one of them. Example: **PCT=PERCENT_1**

Note 1

Many or almost all controls, whether they are proprietary or common, work with variable lists that are shown in boxes. Generally, those lists appear in controls as ITEMS=x and then ITEM1=variable1, ITEM2=variable2, etc[8]. In this case, x is the number of variables that must be shown in the list, and variable1, variable2, etc. are the full names of the variables (entity.variable). In this case, what is shown in the list are the variable labels (if a variable doesn't have a label then what is shown is the name of the variable itself).

[8] In this context, ITEM is a generic name that may symbolize a clause, such as ROW, COL, CTL, etc.

Example:

```
TAL1=PERSON.CHILDY
TAL2=PERSON.MARSTA
```

Note 2

In order to provide more flexibility to controls, several of these clauses must have section names. For example, in the case of formats you could have one section (FORMAT_1) only with the table and graph options, and another section (FORMAT_2) with the table, graph, and map options. The same applies to PCT, SELSET, FILTER, etc.

As mentioned in Chapter IV: Structure and Definitions of the Guest files, the processes have both proprietary and common controls, and each control has both main and peripheral properties. The main processes are listed below.

VI.3 Process: Arealist

Nodetype	**AREALIST**		
Purpose	This process is used to display a list of areas at a certain geographical level. The distribution of variable frequencies is selected by the user.		
Operation	The user selects a geographical output level (a selectable entity), and then selects the variables to be displayed. The program executes a TABLE AREALIST using the selected variables. Those variables from the same or a higher level will appear with a single value in a column, while those variables from lower entities will have their frequency calculated and shown in the list, using one column for each category.		
Particular Controls & Clauses	OUT VAR OPTNAME OPTNAMESEL TOTAL	Output Entity Variables in arealist Totals Row	
Common Controls (grayed out controls are not available)	Areabreak Format Percent Tally	Filter Graph SelSet Weight	Footnote Map Tabletitle
Common Clauses (grayed out clauses are not available)	Altfilter Htmlhelp Title	Caption Nodestyle Universe	Copyright Processtitle Webmaster
Restrictions and Comments	1. Variables at the same level may be any at all. 2. Variables at a lower level should have fewer categories (less than 256). 3. Output entities must be selectable.		

Description of Controls and Clauses

Control/Clause	Type	Comment
OUTN	Section Name	Section that will define a box with a list of entities from where to choose the output entity (see below)[9]
VARN	Section Name	Section that will define a list of variables from where to choose the distribution variables (see below).
OPTNAME	Text	
OPTNAMESEL	RWord	
TOTAL	RWord	Displays a row of totals at the end of the list. Example: **TOTAL=YES** The default value is NO (a row of totals won't be displayed).

Control: variables in arealist (OUT)

Control/Clause	Type	Comment
OUTCAPTION	Text	Text that goes with the entity box.
OUTN	Integer	Number of items in the box. Must include the clauses OUT1, OUT2, etc., up to OUTn.
OUT_i	Entity Name	The name of a database entity. May come with a descriptive text that will replace the label of the entity in the box.

Control: Variables in arealist (VAR)

Control/Clause	Type	Comment
VARCAPTION	Text	Text that comes with the list of variables.
VARN	Integer	Number of items in the list. Must include the clauses VAR1, VAR2, etc., up to VARn
VAR_i	Variable Name	The name of a variable in the database. May come with a descriptive text that will replace the label of the variable in the list.
$VARSEL_i$	RWord	YES to preselect the variable, as if the user had selected it. Default value is NO. Variable 1 is always selected.

[9] Despite being defined as a proprietary control, this section may be reused by several processes, since it's merely a list of entities

Example

Contents	Comment
`[AREAVIV]`	Section Name
`NODETYPE=AREALIST`	Process Type
`CAPTION=Housing`	
`NODESTYLE=AREALIST.DEFAULT`	
`OPTNAME=Include Area Names`	
`OPTNAMESEL=YES`	
`OUTN=OUT_1`	OUT_1 section is below
`VARN=VAR_1`	VAR_1 section is below
`GRAPHCROS=MULTIBARSERIE`	Graph type to be used
`TOTAL=YES`	Displays the row of Totals
`FILTER=FILTER_1`	See example in Filters
`FORMAT=FORMAT_2`	See example in Output Formats
`HTMLHELP=/redatam/RpHelp/procesamientodeindicadores.htm`	
`[OUT_1]`	
`OUTCAPTION=Output Level:`	
`OUTN=3`	3 entities in the list
`OUT1=COMUNA`	
`OUT2=DISTRITO`	
`OUT3=AREA`	
`[VAR_1]`	
`VARCAPTION=Variables to include:`	
`VARN=13`	13 variables in the list
`VAR1=VIVIEN.AGUAD`	
`VAR2=VIVIEN.AGUAO`	
`VAR3=VIVIEN.AGUASN`	
`VAR4=VIVIEN.AUTO`	
`VAR5=VIVIEN.BICICL`	
`VAR6=VIVIEN.COMBUS`	
`VAR7=VIVIEN.COCINA`	
`VAR8=VIVIEN.TENENC`	
`VAR9=VIVIEN.CONDOC`	
`VAR10=VIVIEN.PISO`	
`VAR11=VIVIEN.TECHO`	
`VAR12=VIVIEN.PARED`	Variable 5 in list will be selected for processing
`VAR13=VIVIEN.TIPOVI`	
`VARSEL5=YES`	

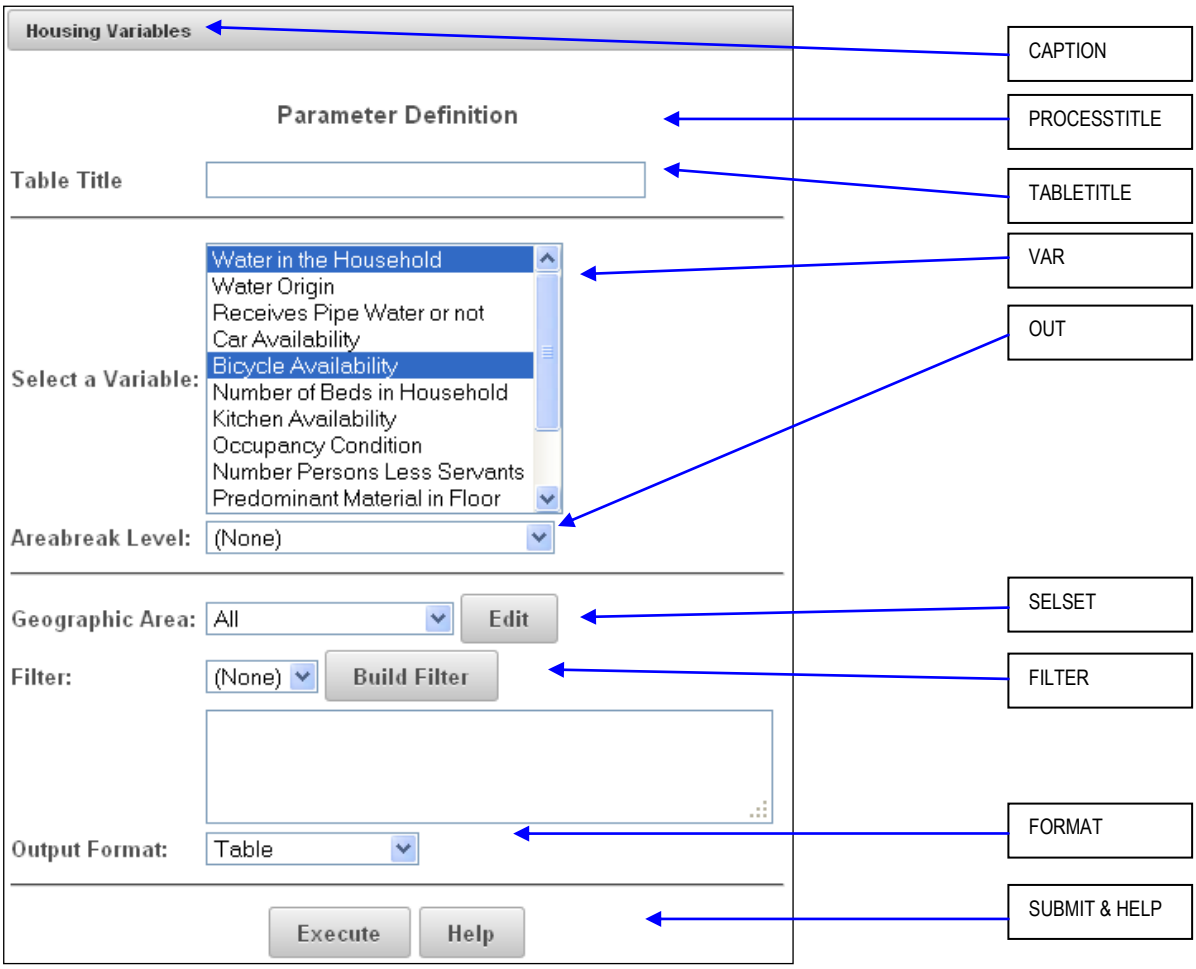


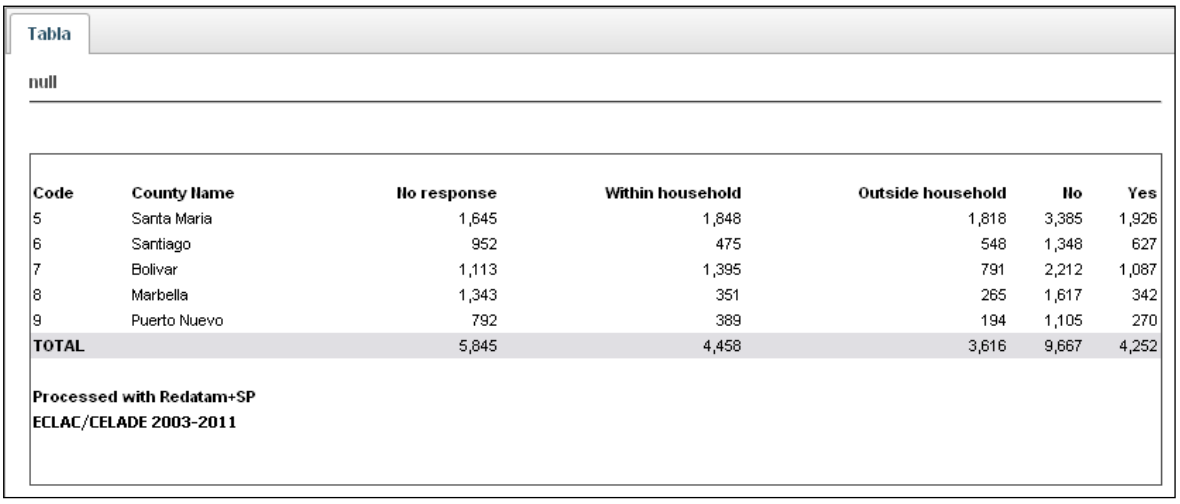
Figure 12. Arealist Process

The Figure begins with a CAPTION (in the example), and then come PROCESSTITLE and TABLETITLE (in the [Glossary]). Then there are an entity box and a list of variables, where the first and fifth variables are preselected. Then the geographical selection, Filter and Output Format, and the Execute and Help buttons (in the [Glossary]).

Note

> The order of the controls on screen may not be changed. However they can be deleted so that they would not display, either by not setting any clauses for the control or by fixing the control using the clause xxxFIXED (FILTERFIXED=PERSON.EDAD>15, for example). This is valid for all processes.

Once two variables have been selected and the Execute button has been pressed, a result as in Figure 13 is displayed



Tabla

null

Code	County Name	No response	Within household	Outside household	No	Yes
5	Santa Maria	1,645	1,848	1,818	3,385	1,926
6	Santiago	952	475	548	1,348	627
7	Bolivar	1,113	1,395	791	2,212	1,087
8	Marbella	1,343	351	265	1,617	342
9	Puerto Nuevo	792	389	194	1,105	270
TOTAL		5,845	4,458	3,616	9,667	4,252

Processed with Redatam+SP
ECLAC/CELADE 2003-2011

Figure 13. Arealist Output

VI.4 Processes: Average & Median

Nodetype	**AVERAGE**		
Nodetype	**MEDIAN**[10]		
Purpose	This process is used to display a box with the average (median) of the selected variable, controlled (or not) by variables in rows, columns, panels, and grid, i.e., up to 4 dimensions (not including the averaged variable). The MEDIAN nodetype is also used to calculate a variable's Minimum and Maximum values.		
Operation	The user selects the variable for which an average (or median) is to be calculated, and also selects, if required, the row and column control variables. The program executes a TABLE AVERAGE (or TABLE MEDIAN) using the selected variables.		
Particular Controls & Clauses	TAL	Variables for which an average (or median) is to be calculated	
	ROW	Row control variables	
	COL	Column control variables	
	CTL	3rd dimension control variable	
	PAN	Panel (4th dimension) control variable	
	MAXIMUM	Calculate the maximum value for the variable	
	MINIMUM	Calculate the minimum value for the variable	
Common Controls	Areabreak	Filter	Format
	Percent	Selection	Graph
	Weight	TableTitle	
Common Clauses	Universe	AltFilter	Caption

[10] These two processes are completely the same, the only thing that changes is the NODETYPE

ProcessTitle

Restrictions and Comments Variables for which an average (or median) is to be calculated must be quantitative variables such as age, income, or number of children.

Description of Controls and Clauses

Control/Clause	Type	Comment
TALN	Section Name	Section that will define the box with the list of variables for which an average (or median) is to be calculated)[11]
ROWN, COLN, CTLN & PANN	Section Names	Sections that will define the list of variables to select the row, column, control, and panel variables.
MAXIMUM	RWord	YES with a MEDIAN nodetype to calculate a maximum value for the variable
MINIMUM	RWord	YES with a MEDIAN nodetype to calculate a minimum value for the variable

Examples

Displays a page with the box of variables for which an average (or median) is to be calculated, and the boxes of row and column variables. Figure 14 shows a sample page and comments.

Contents	**Comment**
`[AVERAGE1]`	Section Name
`NODETYPE=AVERAGE`	Process Type
`NODESTYLE=AVERAGE.DEFAULT`	
`CAPTION=Averages`	
`TALN=TAL_1`	See section below
`TALCAPTION=Average from:`	This caption overrides that of the section, if there is any
`ROWN=ROW_PER2`	See section below
`COLN=COL_PER1`	See section below
`ABKN=ABK_2`	See example in Areabreak
`FILTER=FILTER_1`	See example in Filters
`FORMAT=FORMAT_1`	See example in Output Formats
`[TAL_1]`	Average Variables
`TALN=3`	
`TAL1=PERSON.EDAD`	
`TAL2=PERSON.NCHILDOK`	
`TAL3=PERSON.CHILDAOK`	
`[ROW_PER2]`	Variables de fila
`ROWCAPTION=By (row):`	
`ROWN=18`	
`ROW1=(none)`	
`ROW2=PERSON.CATEOC`	

[11] For all the variable controls, see VARN Control in Average

```
ROW3=PERSON.ASISTE
[COL_PER1]                                        Column Variables
COLCAPTION=By (column):
COLN=5
COL1=(none)
COL2=PERSON.SEXO
HTMLHELP=/redatam/RpHelp/obteniendotabulacionessencilla.htm
```

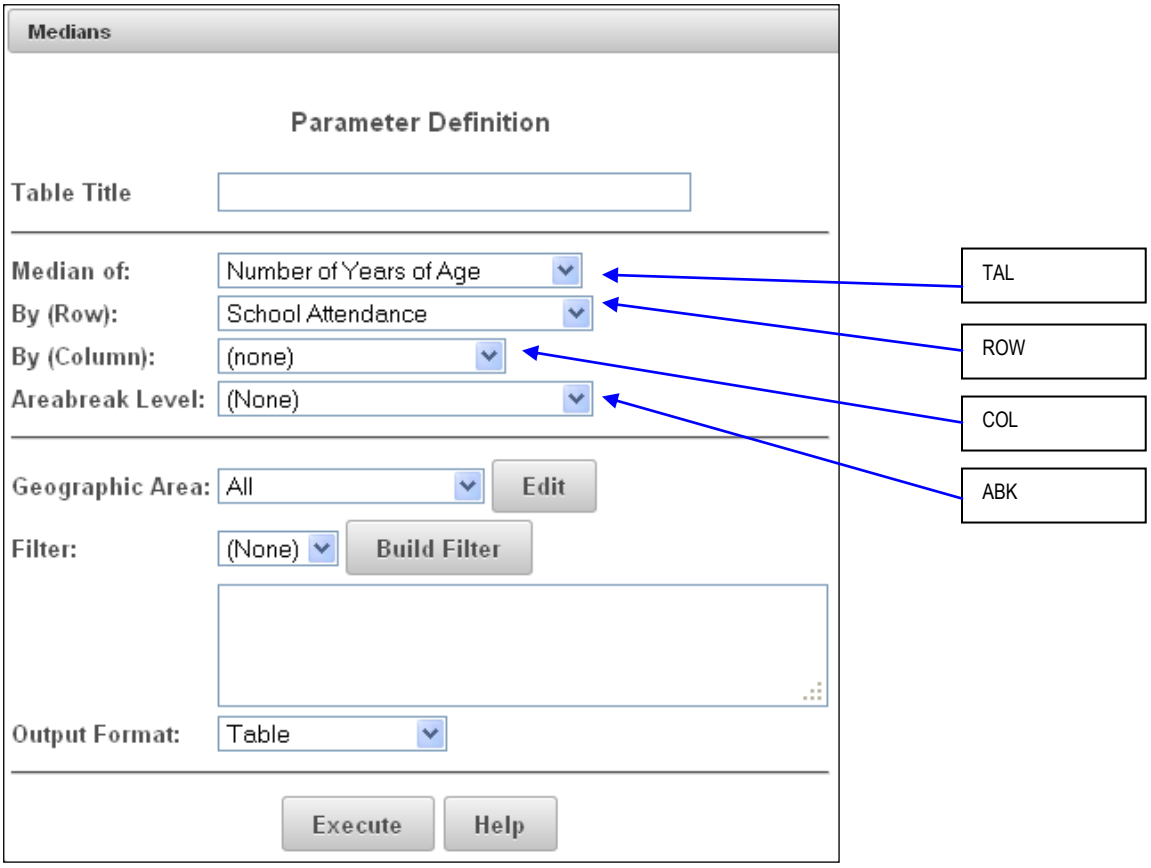
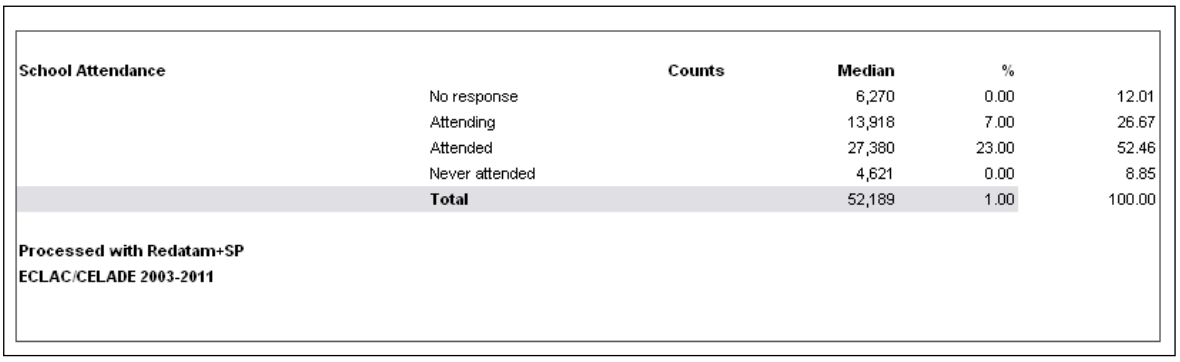


Figure 14. Average Process

The figure shows the variable boxes to be averaged, row and column variables, and area break. Then the geographical selection, filter and output format, and the Execute button. If this variable “Number of Years of Age” has been selected to be averaged and “School Attendance” has been selected as row variable, when pressing the Execute button the result will display as in Figure 15.



School Attendance		Counts	Median	%	
	No response		6,270	0.00	12.01
	Attending		13,918	7.00	26.67
	Attended		27,380	23.00	52.46
	Never attended		4,621	0.00	8.85
	Total		52,189	1.00	100.00

Processed with Redatam+SP
ECLAC/CELADE 2003-2011

Figure 15. Average Output

VI.5 Process: Crosstabs

Nodetype **CRUZ**

Purpose This process is used to display a box with a crossing of variables, up to 5 dimensions.

Operation The user selects the row, column, ctrl, panel, and grid variables. The program executes a TABLE CROSSTABS using the selected variables.

Particular Controls & Clauses

ROW	Row Variables
COL	Column Variables
CTL	3rd Dimension Variables
PAN	Panel (4th Dimension) Variables
GRD	Grid (5th Dimension) Variables
PERCENT	Options for calculation of percentages

Common Controls

Areabreak	Filter	Format
Percent	Selection	Tally
Weight	TableTitle	

Common Clauses

Universe	AltFilter	Caption
ProcessTitle		

Restrictions and Comments The COL variable may have up to 256 categories.

Control: Row, Column, and Other Variables (ROW, COL, CTL, PAN, GRD)

Control/Clause	Type	Comment
ROWN, COLN, CTLN, PANN & GRDN	Section Names	Sections that will define the list of variables to select the row, column, control, panel, and grid variables[12]
PERCENT	Section Name	Section to define the percentage option (see below)

[12] For all the variable controls, see VARN Control in Average

Control: Options for Percentage Calculation (Percent)

Control/Clause	Type	Comment
PERCENTCAPTION	Text	Caption that comes with the box of items
PERCENTN	Integer	Number of items in the list. Must include the clauses PERCENT1, PERCENT2, etc., up to PERCENTn
PERCENT$_i$	RWord & Text	Reserved word symbolizing the percentage type to be calculated, and a descriptive text which will be displayed in the box on screen. Options include:
	OFF	No percentage, only absolute values
	ROW	Row percentage
	COLUMN	Column percentage
	TOTAL	Total percentage
PERCENTLAST	Integer	Number of the item to be initially shown in the box. By default this is 1.

Example

Contents	**Comment**
`[PERCENT_1]`	Section Name
`PERCENTCAPTION=Percentage Options:`	Control Caption
`PERCENTN=4`	Number of items
`PERCENT1=OFF Absolute Value`	Values for each item, with its reserved word and the text to be shown in the control box on screen
`PERCENT2=ROW % Row`	
`PERCENT3=COLUMN % Column`	
`PERCENT4=TOTAL % Total`	

The result is displayed on screen as shown in Figure 16.

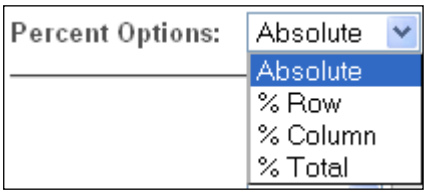


Figure 16. Percentage Options

Example

Displays a page with boxes of row, column, and control variables. Figure 17 shows a sample page.

Contents	**Comment**
`[CRUCVIV]`	Section ID
`NODETYPE=CRUZ`	Node Type
`NODESTYLE=CRUZ1.DEFAULT`	See examples in NODESTYLE
`CAPTION=Housing`	

`ROWN=ROW_VIV`	Section with a list of row variables
`ROWCAPTION=Crossing of (row):`	This caption overrides that of the section, if there is any
`COLN=COL_PER2`	Section with a list of column variables
`COLCAPTION=By (column):`	This caption overrides that of the section, if there is any
`COLLAST=10`	By default, the variable displayed in the box
`CTLN=CTL_1`	3rd dimension section name (in this case, variables to simulate an area break). See below.
`FILTER=FILTER_1`	See example in Filters
`FORMAT=FORMAT_3`	See example in Output Formats
`PERCENT=PERCENT_1`	See example of PERCENT_1 above
`[CTL_1]`	Section with 3 items. The first item is for No breaks, and the other 2 items are to simulate a Column break and a District break.
`CTLCAPTION=Output Level:`	
`CTLN=3`	
`CTL1=(The%20entire%20database)`	
`CTL2=COMUNA.COMUNA`	
`CTL3=DISTRITO.DISTRIT`	
`CTLLAST=1`	Default item to be displayed in the box. NOT required, since by default it will always be the first item

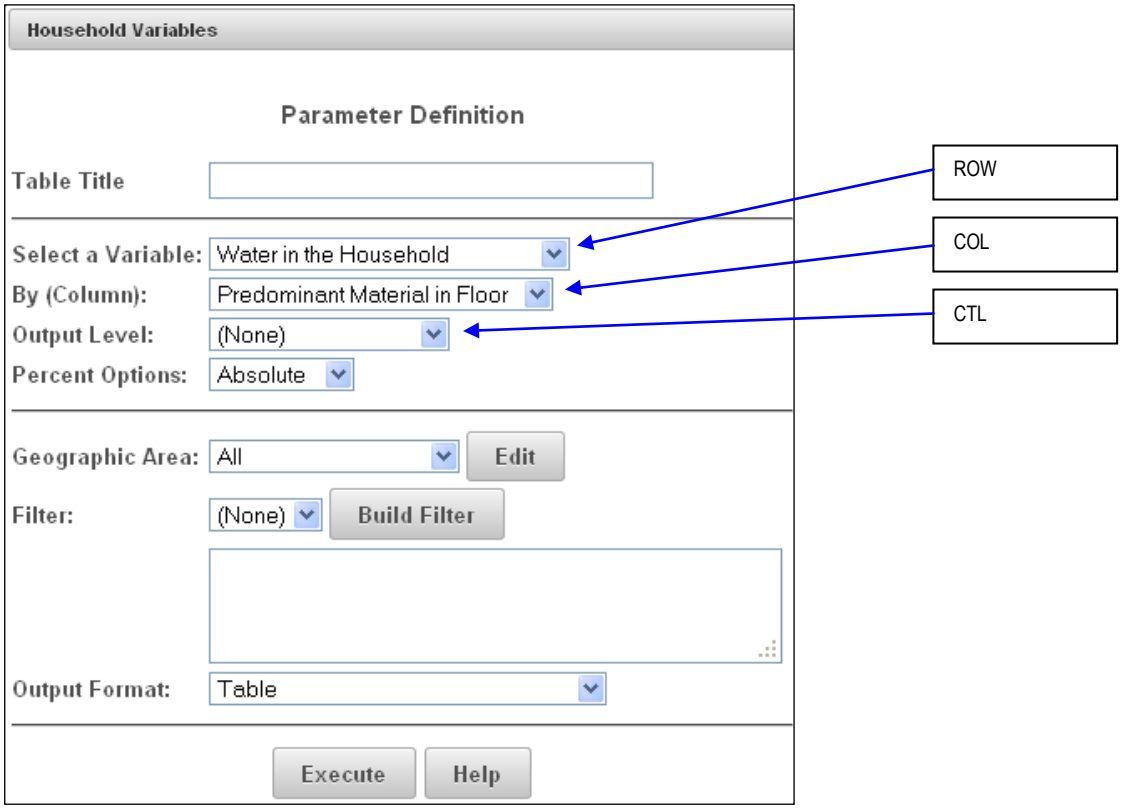
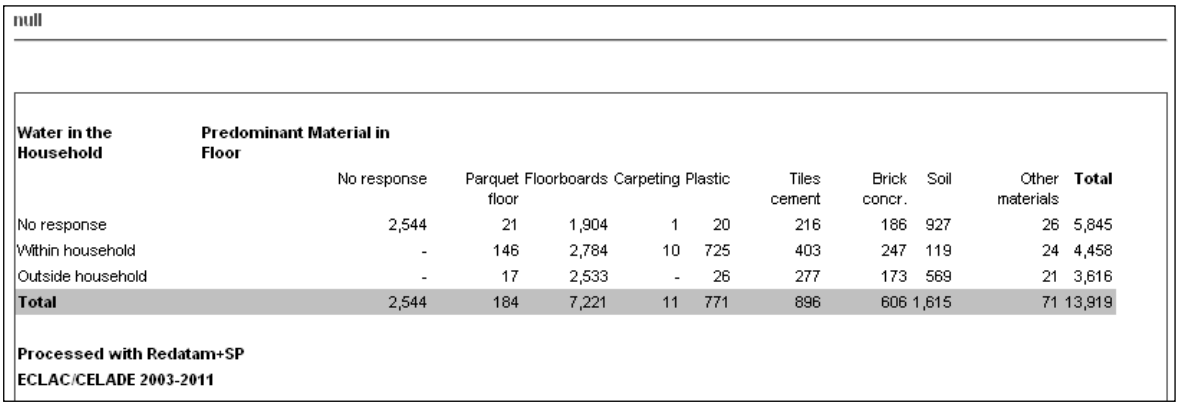


Figure 17. Cruz Process Screen

This figure shows the row, column, and control boxes. In that example, the third dimension in the table is used to represent the area break (being CTL2 the Enumeration

County and CTL3 the DISTRICT). Then the geographical selection, filter and output format, and the Execute button. Using this selection, the “Water” variable will be crossed by the “Floor” variable, and when pressing the Execute button the results will display as in Figure 18.



null

Water in the Household	Predominant Material in Floor									
	No response	Parquet floor	Floorboards	Carpeting	Plastic	Tiles cement	Brick concr.	Soil	Other materials	**Total**
No response	2,544	21	1,904	1	20	216	186	927	26	5,845
Within household	-	146	2,784	10	725	403	247	119	24	4,458
Outside household	-	17	2,533	-	26	277	173	569	21	3,616
Total	2,544	184	7,221	11	771	896	606	1,615	71	13,919

Processed with Redatam+SP
ECLAC/CELADE 2003-2011

Figure 18. Cruz Output

VI.6 Process: Dependency Ratio

Nodetype	**DEPRATIO**		
Purpose	This process is used to calculate the dependence ratio between the population of ‘supported’ age divided by the ‘supporting’ population. By ‘supported’ age are defined those people who cannot enter yet the labor market, and the population that is no longer in the labor market. And the ‘supporting’ age population are those people of working age (active age population). These 2 sets are mutually excluding.		
Operation	In order to define both counts, the user must select the minimum and maximum ages (ages are mutually exclusive, i.e., no 2 ages are considered as ‘supported’). For each output level item, the program calculates the ‘supported’ and ‘supporting’ populations, divides one by another, and then executes a TABLE AREALIST using the dependence ratio.		
Particular Controls & Clauses	AGEVAR ROW OUTPUT	Definition of minimum and maximum ages Output entities Display options	
Common Controls	 Weight	Filter Selection TableTitle	Format
Common Clauses	Universe ProcessTitle	AltFilter	Caption
Restrictions and Comments	-		

Description of Controls and Clauses

Controls below may also be displayed separately as sections. In this case, use as a clause the name of the control followed by the letter N and a section name, as in most controls. For example:

```
AGEVARN=AGE1_SECTION
ROWN=ENTITY1_SECTION
OUTPUTN=OUTPUT1_SECTION
```

Control: Definition of minimum and maximum ages (AGEVAR)

Control/Clause	Type	Comment
AGEVAR	Variable Name	Variable in the database (or calculated in DEFINEs) with data on persons' ages.
AGEMINN	Integer	Number of items in the minimum age box. Must include the clauses AGEMIN1, AGEMIN2, etc., up to AGEMINn
AGEMIN$_i$	Value & Text	Age in years and descriptive text. This shall be the text to be displayed in the minimum age box on screen.
AGEMINCAPTION	Text	Text that goes with the minimum age box
AGEMAXN	Integer	Number of items in the maximum age box. Must include the clauses AGEMAX1, AGEMAX2, etc., up to AGEMAXn
AGEMAX$_i$	Value & Text	Age in years and descriptive text. This shall be the text to be displayed in the maximum age box on screen.
AGEMAXCAPTION	Text	Text that goes with the maximum age box.

Control: Output Entities (ROW)

Control/Clause	Type	Comment
ROWCAPTION	Text	Text that goes with the entity box.
ROWN	Integer	Number of items in the entity box. Must include the clauses ROW1, ROW2, etc., up to ROWn
ROW$_i$	Entity name and text (optional)	Name of the output level entity. The entity label appears in the entity box and may be replaced with the (optional) text following the entity name.

Control: Display Options (OUTPUT)

Control/Clause	Type	Comment
OUTPUTCAPTION	Text	Text that goes with the display option box.
OUTPUTN	Integer	Number of items in the display box. Must include the clauses OUTPUT1, OUTPUT2, etc., up to OUTPUTn

OUTPUT$_i$	Rword & Text	Reserved word symbolizing the output type to be displayed, and a descriptive text which will be displayed in the box on screen. Options include:
	OFF	Only the dependence ratio
	COUNTS	Also includes absolute values

Example

Displays a page with the minimum and maximum age boxes, and the output entity box. Figure 19 shows a sample page. Below the figure are some comments on the example.

Contents	**Comment**
`[INDICDEPRATIO]`	Section Name
`NODETYPE=DEPRATIO`	Node Type
`AGEVAR=PERSONA.EDAD`	Age variable
`AGEMINN=8`	Number of minimum ages
`AGEMINCAPTION=Minimum Age`	Minimum age box caption
`AGEMIN2=13 Age 12 and below`	Minimum age values and descriptive text
`AGEMIN3=14 Age 13 and below`	
`AGEMIN4=15 Age 14 and below`	
`AGEMIN5=16 Age 15 and below`	
`AGEMIN6=17 Age 16 and below`	
`AGEMAXN=5`	The same applies to maximum ages
`AGEMAXCAPTION=Maximum Age`	
`AGEMAX1=59 Age 60 and above`	
`AGEMAX2=64 Age 65 and above`	
`ROWN=3`	Number of entities
`ROWCAPTION=Output Level`	Entity box caption
`ROW1=DISTRITO District`	Output entities
`ROW2=CTV Town/City`	
`ROW3=ED Enumeration Sector`	
`OUTPUTN=4`	Number of output options
`OUTPUTCAPTION=Output Options`	Option box caption
`OUTPUT1=OFF Ratio Only`	Displays ratio only
`OUTPUT2=COUNTS Include absolute values`	Displays the numerator, denominator, and ratio values

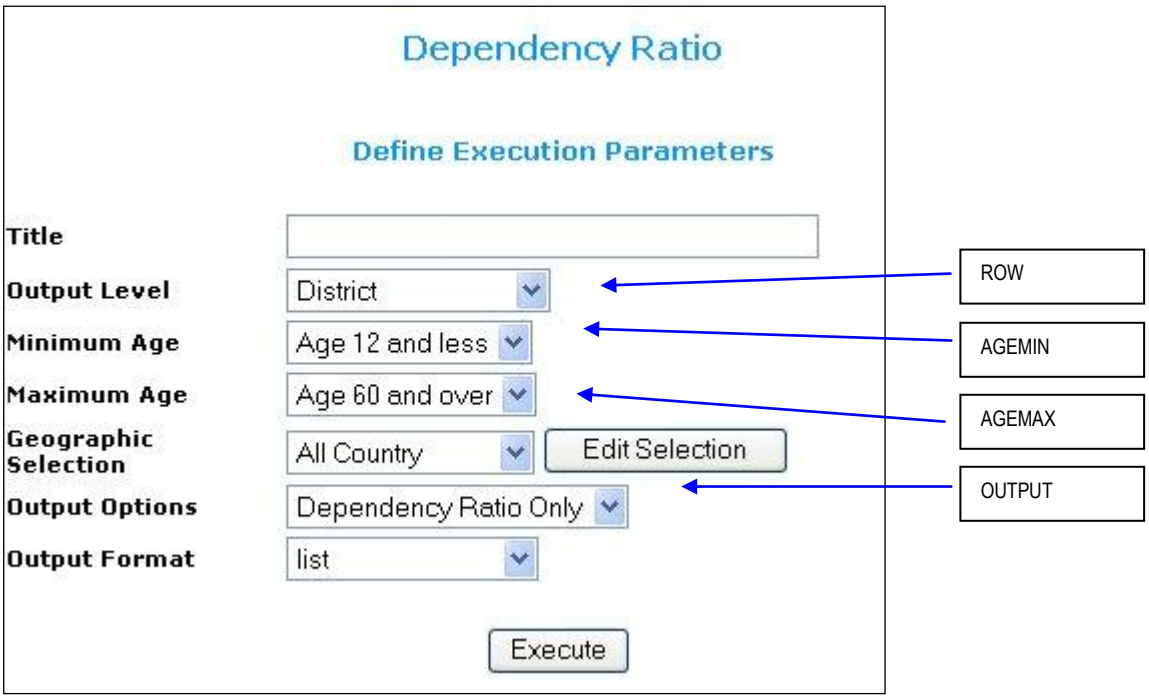


Figure 19. Dependency Ratio Process

The figure above shows the output entity, minimum age and maximum age boxes. Then the geographical selection, the output options and output format, and the Execute button. In this example there are no user-selectable filters. When the minimum age is selected as 13, the maximum age is selected as 59, the output entity is selected as District, and Execute button is pressed, the results display as in Figure 20.

Code	District Name	Dependency Ratio
1	Corozal	0.70
2	Orange Walk	0.70
3	Belize	0.63
4	Cayo	0.77
5	Stann Creek	0.74
6	Toledo	0.92

Figure 20. Dependency Ratio Output

VI.7 Process: Dictionary

Nodetype	**DICTIONARY**
Purpose	This process is used to display the contents of the REDATAM

database's data dictionary.

Operation

When MODE=SHORT, only the entity variable list is displayed. When MODE=VARIABLE, the program displays a box with the entity variables, allowing for multiple selections, so that the user may select one or more variables to show the categories.

Particular Controls & Clauses

MODE	Display mode
ENTITY	Dictionary entity
BUTTONFREQUENCY	Execute button

Restrictions and Comments

This process would NOT accept any of the common controls.

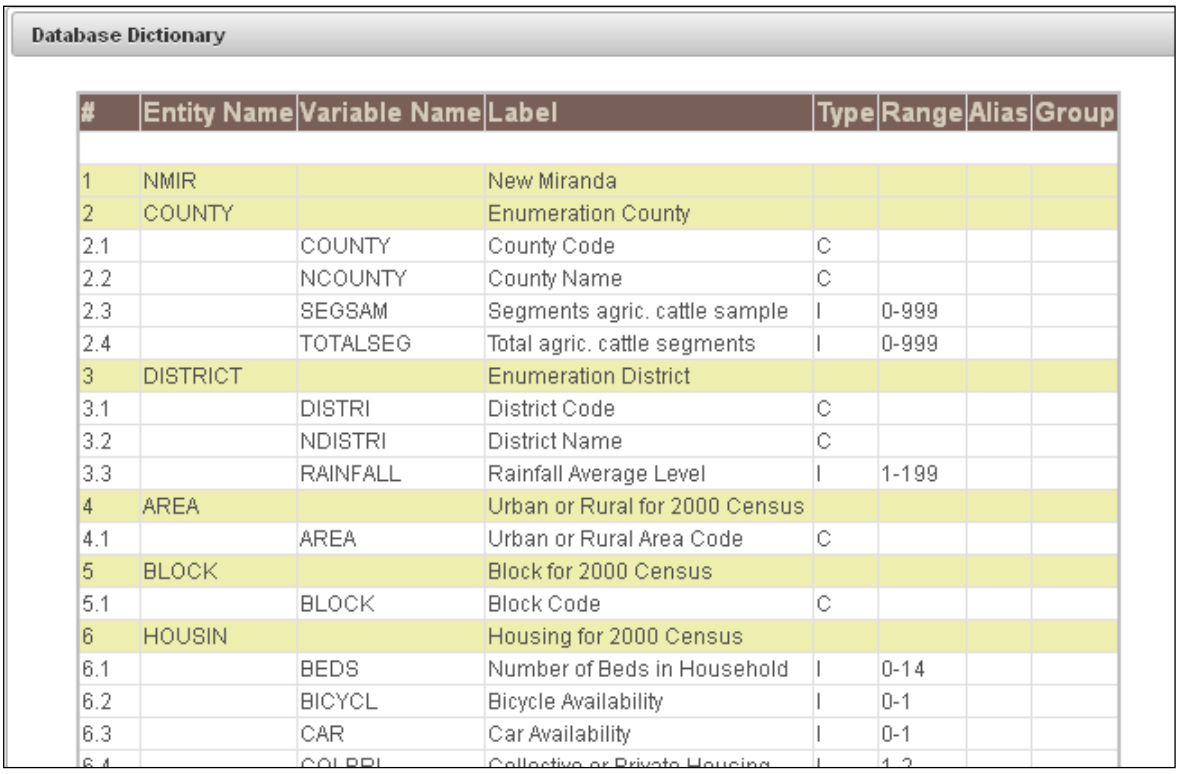
Description of Controls and Clauses

Control/Clause	Type	Comment
MODE	RWord	Type of dictionary display. It may be SHORT, only for entities and variables with their names and labels, or VARIABLE, to display the variables' categories. Example: **MODE=SHORT** **MODE=VARIABLE**
ENTITY	Entity Name(s)	Name of a database entity, or list of database entities, separated by blank spaces (only if clause MODE=SHORT). If this clause is not reported, then the system will assume all entities
BUTTONFREQUENCY	Character	
PROCESSTITLE	Text	Text to be displayed on the variable list. Example: **PROCESSTITLE=-- Select one or more variables (Press Ctrl key) --**

Examples

1. Displays a list of all the variables in the dictionary (ENTITY not reported). Figure 21 below shows the output.

Contents	**Comment**
`[DICALL]`	Section ID
`NODETYPE=Dictionary`	Node Type
`CAPTION=All Variables`	
`TITLE=Database Dictionary`	
`MODE=SHORT`	Only the names and labels of variables



Database Dictionary

#	Entity Name	Variable Name	Label	Type	Range	Alias	Group
1	NMIR		New Miranda				
2	COUNTY		Enumeration County				
2.1		COUNTY	County Code	C			
2.2		NCOUNTY	County Name	C			
2.3		SEGSAM	Segments agric. cattle sample	I	0-999		
2.4		TOTALSEG	Total agric. cattle segments	I	0-999		
3	DISTRICT		Enumeration District				
3.1		DISTRI	District Code	C			
3.2		NDISTRI	District Name	C			
3.3		RAINFALL	Rainfall Average Level	I	1-199		
4	AREA		Urban or Rural for 2000 Census				
4.1		AREA	Urban or Rural Area Code	C			
5	BLOCK		Block for 2000 Census				
5.1		BLOCK	Block Code	C			
6	HOUSIN		Housing for 2000 Census				
6.1		BEDS	Number of Beds in Household	I	0-14		
6.2		BICYCL	Bicycle Availability	I	0-1		
6.3		CAR	Car Availability	I	0-1		
6.4		COLPRI	Collective or Private Housing	I	1-2		

Figure 21. All the variables in the dictionary

2. Displays variables for some entities. Figure 6 in item II.3 shows the output.

Contents	Comment
`[DICCATVIV]`	Section ID
`NODETYPE=Dictionary`	Node Type
`CAPTION=Geographical Variables`	
`MODE=SHORT`	Only the names and labels of variables
`ENTITY=COMUNA DISTRITO AREA MANZAN AGROSEGM`	Entity names to display the variables

3. Displays a page for selecting the variables to list the categories (see Figure 22). If the variables highlighted in blue are selected and the Execute button is pressed, the program will display Figure 23.

Contents	Comment
`[DICGEO]`	Section ID
`NODETYPE=Dictionary`	Node Type
`CAPTION=Categories for Housing Variables`	
`MODE=VARIABLE`	Variables and their categories
`ENTITY=VIVIEN`	Only housing variables
`PROCESSTITLE=-- Select one or more variables (Press Ctrl key) --`	

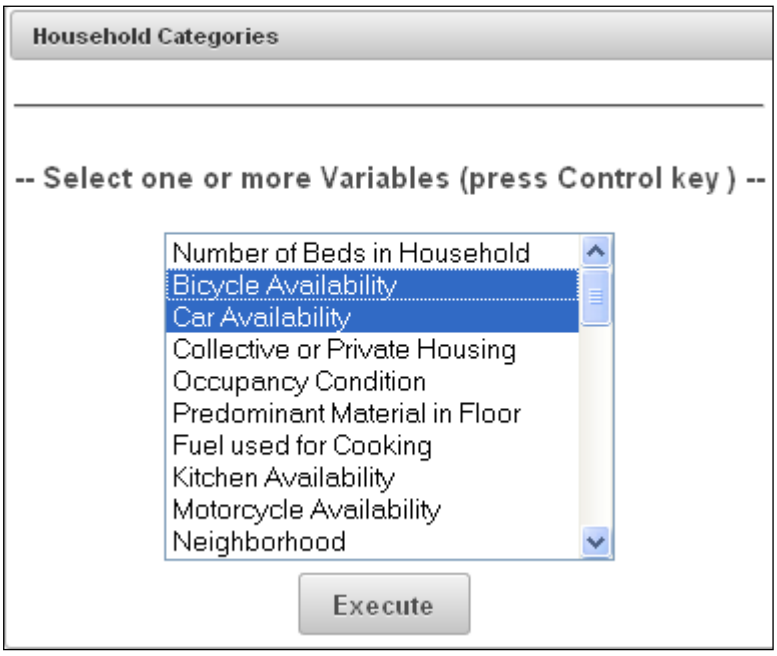


Figure 22. Dictionary Process

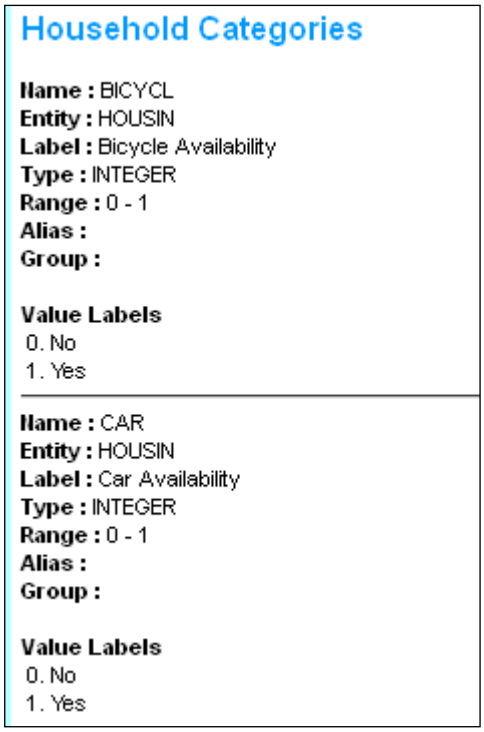


Figure 23. Output from the Categories Process

VI.8 Process:

VI.9 Ratio (Fraction)

Nodetype	**FRACTION**
Purpose	Create an AREALIST using an indicator that is the division of a value (numerator) by another (denominator).
Numerator	A combobox containing a list of categorical variables from which the categories to be used as a filter for counting in the numerator portion of the fraction may be selected.
Denominator	A combobox containing a list of categorical variables from which the categories to be used as a filter for counting in the denominator portion of the fraction may be selected. If no denominator used, then the process is a percentage of the number of cases in the numerator divided by the total number of cases.
Operation	For the numerator and denominator, the system displays the categories of the selected variable. Once the categories of each variable (one in the numerator and one in the denominator) have been selected, the system takes the categories of the variable in the denominator and creates a DEFINE using RECODE at the same level as the entity, to be subsequently used as a filter in a DEFINES using COUNT, to count cases at the level of the output entity. The same process is then applied to the variable in the denominator. Then, at the level of the arealist entity output, the system divides one variable by another. If there is no denominator, then the same entity in the numerator is used as total entity, which means that the indicator will be a percentage.
Particular Controls & Clauses	INDICLABEL Titles NUM Numerator DENUM Denominator OUT Output Level Entities PERCENT Multiplier OUTPUT Display options
Common Controls	Map, Filter, Format Graph, Selection, Tally Weight, TableTitle
Common Clauses	Universe, AltFilter, Caption ProcessTitle, Footnote, Webmaster
Restrictions and Comments	Entities in the numerator and denominator must be of at a lower level than the Arealist output entity

Description of Controls and Clauses

Control/Clause	Type	Comment
INDICLABEL	Text	Label of the indicator column in the Arealist table. Optional.

Description of Controls and Clauses

Combobox listing the alternatives to be used as numerator (top part) of the indicator. Consists of a list of variables from any level.

Control/Clause	Type	Comment
NUMCAPTION	Text	Numerator box label, displayed on top of the box
NUMN	Integer	Number of variables that may be used as numerator. Controls the existence of the NUMi clauses
NUM$_i$	Variable Name	Items in the numerator. They contain the full names of the variables (entity.variable).
NUMWIDTH	Integer	Length of the box containing variable names
NUMVLWIDTH	Integer	Length of the box containing variable categories
NUMLAST	Integer	Shows the number of the variable in the list that is shown in the box
NUM$_i$VLNUM$_j$	RWord NO YES	Preselection of the category of a variable in the box. The default value is NO. When displaying the indicator in the page, the category *j* of the variable *i* will be displayed as selected in the list. Example: **NUM1.VLNUM3=YES** The third category for the first variable in the list will be displayed as preselected

Control: Denominator (DENUM)

The same clauses as for the numerator, replacing the prefix NUM with DENUM.

Control: Multiplier (PERCENT)

A combobox with the options to be used as a multiplier of the division's numerator.

Control/Clause	Type	Comment
PERCENTCAPTION	Text	Multiplier box label, displayed to the left of the box
PERCENTN	Integer	Number of multiplier types. Controls the existence of the PERCENTi clauses
PERCENT$_i$	Numeric and text (optional)	Choice of multiplier. It must strictly follow the syntax of a numeric value followed by any label (optional). Examples:

		100 Multiplied by 100 **1000**
PERCENTFIXED	Numeric	Fixed numeric value, not shown on screen for selection, and directly used to multiply the numerator.

Note

1. If there are no values, there is no multiplication
2. PERCENTFIXED overrides PERCENTN
3. The '(none)' option may be included in PERCENTi. In fact, any text is valid if between brackets
4. The numeric value may be decimal, e.g., 0.1, and in that case, the multiplier works as a divider (by 10 in this example)

Control: Output Type (OUTPUT)

A combobox listing alternatives for output options, whether indicator only or also the numerator and denominator absolute values.

Control/Clause	Type	Comment
OUTPUTCAPTION	Text	Label on the output option screens, displayed to the left of the box
OUTPUTN	Integer	Number of output types, generally 2. Controls the existence of the $OUTPUT_i$ clauses
$OUTPUT_i$	RWord & Text OFF COUNTS	Reserved word symbolizing the output type to be displayed, and a descriptive text which will be displayed in the box on screen. Options include: Only the dependence ratio Also includes absolute values
OUTPUTFIXED	RWord	OFF or COUNTS, fixed, not shown on screen for selection. In fact, it only makes sense as COUNTS, since it suffices to delete the OUTPUT clause for the system to assume OFF as default value.

Note

1. If the control provides no values, then assume OFF
2. OUTPUTFIXED overrides OUTPUTN

Example

Calculation of dependence ratio. Displays the numerator and denominator with the preselected categories, as seen in Figure 24. The execution is in Figure 25.

Contents	**Comment**
`[INDFRACTIONR]`	Section ID
`NODETYPE=Fraction`	Node Type

`OUTPUTN=2`
`OUTPUTCAPTION=Output Options`
`OUTPUT1=OFF Indicator`
`OUTPUT2=COUNTS Include cases`
	Display options

`NUMN=2`
`NUMCAPTION=Numerator:`
`NUM1=PERSON.EDADGRA`
`NUM2=PERSON.EDQUINQ`
	Numerator, with 2 variables

`NUM1.VLNUM1=YES` `NUM1.VLNUM3=YES`	The first and third categories of the EDADGRA variable are shown preselected in the page

`DENUMN=2`
`DENUM1=PERSON.EDADGRA`
`DENUM2=PERSON.EDQUINQ`
	Denominator, with the same variables

`DENUM1.VLNUM2=YES`	The second category of the EDADGRA variable is shown selected

`PERCENTCAPTION=Multiplier`
`PERCENTN=4`
`PERCENT1=(none)`
`PERCENT2=100`
`PERCENT3=1000`
`PERCENT4=10000`
	Ratio Multiplier
	(not in the NMIR example)

`OUTN=OUT_1`	The output levels are defined in the OUT_1 section

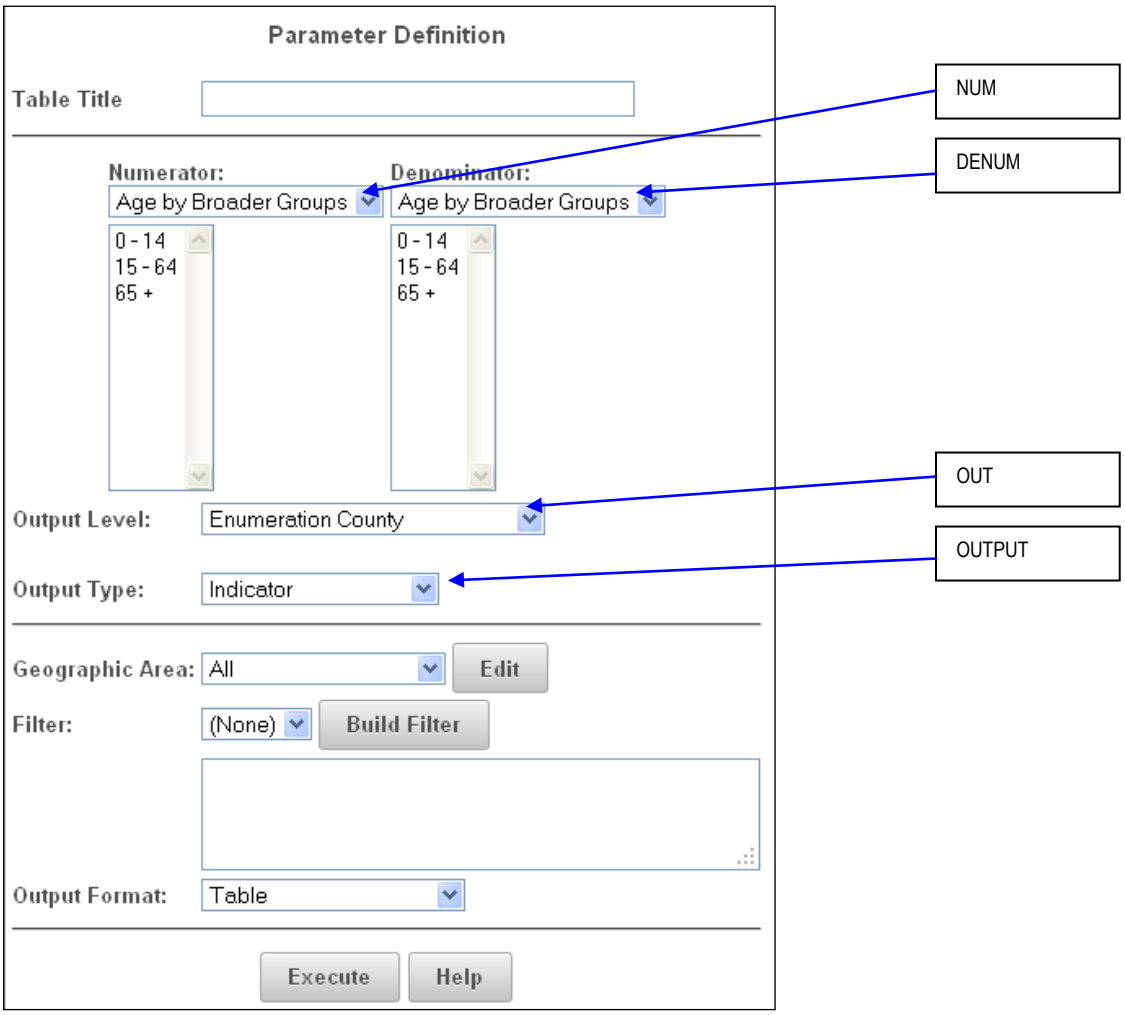


Figure 24. Fraction Process

Figure 25. Fraction Output

VI.10 Process: Frequency

Nodetype	**FREQUENCY**		
Purpose	This process is used to display a box with the distribution of frequencies from one or more variables. This is similar to the Cruz process, but in a single dimension.		
Operation	The user selects one or more variables from the list. The program executes a TABLE FREQUENCY using the selected variables.		
Particular Controls & Clauses	ROW ENTITY	Distribution Variables Distribution Variable Entity	
Common Controls	Areabreak Weight	Filter Selection TableTitle	Format Tally
Common Clauses	Universe ProcessTitle	AltFilter	Caption
Restrictions and Comments	-		

Description of Controls and Clauses

Control/Clause	Type	Comment
ENTITY	Entity Name	Entity for which all the names of variables in the box will be listed.

Control: Distribution Variables (ROW)

Control/Clause	Type	Comment
ROWN	Section Name	Defines a list of variables to select the distribution variables.

Example

Displays a page with the variable box. Figure 26 shows a sample page. Below the figure are some comments on the example.

Contents	Comment
`[FREQPOB]`	Section ID
`NODETYPE=FREQUENCY`	Node Type
`ROWN=ROW_PER1`	See section below
`ABK=ABK_1`	Lists of area breaks, filters, and formats
`FILTER=FILTER_1`	
`FORMAT=FORMAT_1`	

```
...
[ROW_PER1]
ROWCAPTION=Select a Variable
ROWN=17
ROW1=PERSON.TIPOAC
ROW2=PERSON.ASISTE
ROW3=PERSON.CURSO
ROW4=PERSON.EDAD
ROW5=PERSON.ECIVIL
ROW6=PERSON.EDQUINQ
ROW7=PERSON.OCUP
ROW8=PERSON.NHIJOS
```

Independent section, with a list of variables for distribution

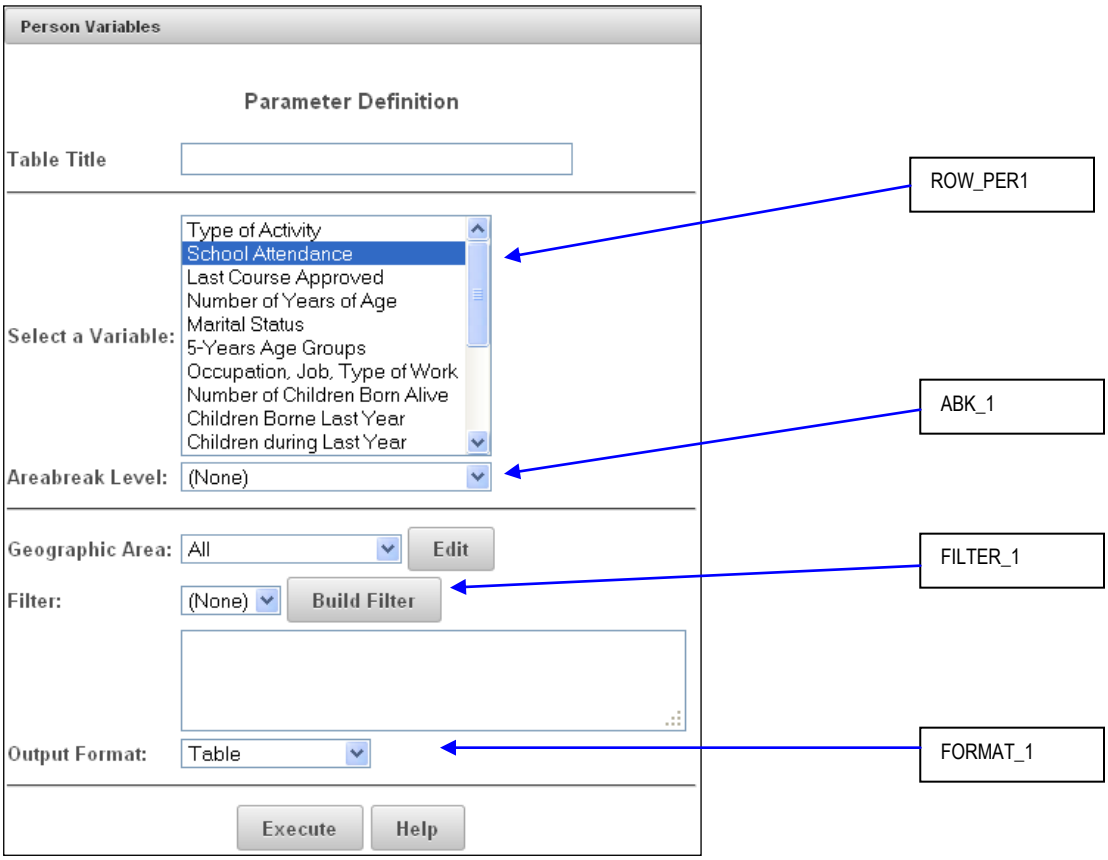
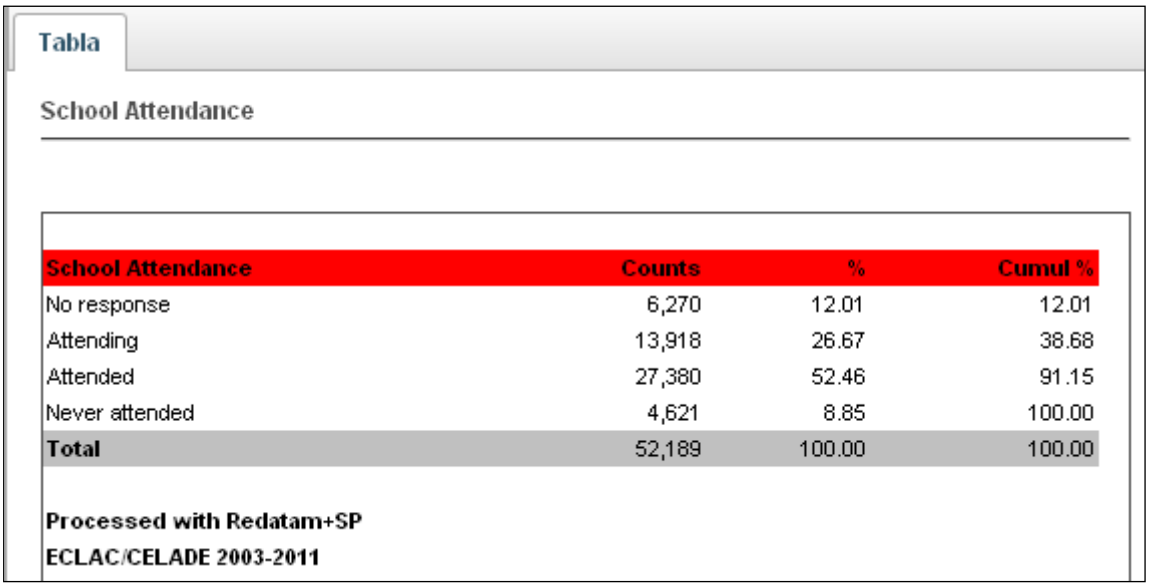


Figure 26. Nodetype Frequency

The Figure shows the variable box, geographical selection, filter, output format, and Execute button. When the 'School Attendance' variable is selected and the Execute button is pressed, the result displays as in Figure 27.



Tabla

School Attendance

School Attendance	Counts	%	Cumul %
No response	6,270	12.01	12.01
Attending	13,918	26.67	38.68
Attended	27,380	52.46	91.15
Never attended	4,621	8.85	100.00
Total	52,189	100.00	100.00

Processed with Redatam+SP
ECLAC/CELADE 2003-2011

Figure 27. Frequency Output

VI.11 Process: InputSPC

Nodetype	**INPUTSPC**		
Purpose	This process is used to execute a preset program, by entering user-defined parameters.		
Operation	The process depends mainly on a base program (template), which is a common program in REDATAM language but including some parameters to be substituted during runtime. The process page is intended for the user to select the values that the parameters must have. This is a very flexible and powerful process, because it can perform many different tasks depending on the contents of the base program. Of course the output from this process could be any type of table that is defined by the base program.		
Particular Controls & Clauses	TEMPLATESPC PARAMS	Distribution Variables Distribution Variable Entity	
Common Controls	Areabreak Weight	Filter Selection TableTitle	Format Tally
Common Clauses	Universe ProcessTitle	AltFilter	Caption
Restrictions and Comments	-		

Description of Controls and Clauses

Control/Clause	Type	Comment
TEMPLATESPC	Text	The name of a REDATAM program with substitution parameters.
PARAMS	Integer	Number of parameters reported in numeric sequence. Must include the controls PARAM1, PARAM2, etc., up to PARAMn

Control: Execution Parameters (PARAMi)

Each of these parameters will be used to populate an item box (combobox). There are no limits to the number of parameters.

Control/Clause	Type	Comment
PARAM$_i$CAPTION	Text	Text that goes with the item box. Example: **PARAM1CAPTION=Select the geographical output level**
PARAM$_i$N	Integer	Number of items in the variable box. Must include the clauses PARAMi1, PARAMi2, etc., up to PARAMin
PARAM$_{ik}$	Item text and descriptive text	Contents of the item that will be entered into the program, replacing the parameter, followed by a descriptive text that will be displayed in the box. **Important: the contents of the item cannot include any blanks.**
PARAM$_i$FIXED	Text	Directly replaces the parameter in the base program, without input from the user (its contents are not shown in the page). This overrides PARAM$_i$CAPTION and PARAM$_{ik}$.

Base Program (Template)

This is an ordinary program in REDATAM which additionally may have parameters written as % n, where n is a number, e.g. %3. These parameters serve as “placeholders” for the actual values, which will be reported by the user when executing the Process.

In addition to the numeric parameters there are fixed name parameters, including:

%SELECTION	for geographical selection
%UNIVERSE	for universe

Examples

This example works with a base program (template) that crates a ratio indicator between a numerator that has been calculated by counting the cases of persons with some type of problem, divided by the total population. Figure 28 shows a page created for this example. Below the Figure is shown the program generated by combining the base program and the parameters entered for selections on screen.

Contents	Comment
[DISABILITYRATIO] NODETYPE=InputSPC CAPTION=Disability Ratio TABLETITLE=Disabled Population (%) PROCESSTITLE=Disabled Population by Area	Section ID, node type, and titles
TEMPLATESPC=RatioGeneric.spc	Name of the Program to be used as a base. It must be in the SITE directory
SELECTION=SELECTION1	In this case the selection refers to an existing section, that is common to several processes
UNIVERSEN=4 UNIVERSECAPTION=Selection Population UNIVERSE1=() No Specific Selection UNIVERSE2=PERSON.AGE>64 Elder Population(65+) UNIVERSE3=PERSON.AGE<15 Young Population(<15) UNIVERSE4=PERSON.AGE>79 Elderly Population(80+)	The Universe has 4 items. Each item is shown through a logical expression (without intervening spaces) and then a descriptive text that will be shown in the box
PARAMS=9	Number of avulsion parameters. It would be more efficient if it were 3 and the parameters would range from 1 to 3
PARAM4FIXED=PERSON	Parameter %4 are fixed with the entity PERSON. It could have been better if this entity were already in the template program, without the need for a parameter, but it was made this way to demonstrate that the program template may be used to count entities other that persons.
PARAM5N=3 PARAM5CAPTION=Output level PARAM51=DISTRICT List by District PARAM52=CTV List by City/Town/Village PARAM53=ED List by ED	Parameter %5 with 3 items, with the process output entities
PARAM9N=10 PARAM9CAPTION=Select Target Population PARAM91=PERSON.SIGHT=1 Sight PARAM92=PERSON.HEARING=1 Hearing PARAM93=PERSON.SPEECH=1 Speech PARAM94=PERSON.MOBILITY=1 Moving/Mobility PARAM95=PERSON.BODYMOV=1 Body Movements PARAM96=PERSON.GRIPP=1 Gripping PARAM97=PERSON.LEARN=1 Learning PARAM98=PERSON.BEHAVIOU=1 Behavioural	Parameter %9 with 10 items. Each item is shown through a logical expression (without intervening spaces) and then a descriptive text that will be shown in the box

```
PARAM99=PERSON.PERSCARE=1 Personal care
PARAM910=PERSON.DISOTHER=1 Other Disability
```

RatioGeneric.spc Base Program

Contents	Comment

`RUNDEF DisabilityRatio`
`  SELECTION %SELECTION`
`  %UNIVERSE`
	Selection with the parameter
	Universe with the parameter

`*  %SELECTION Selection File`
`*  %UNIVERSE Universe Condition`
`*  %4 Entity counted`
`*  %5 Entity output`
`*  %9 Target count`
	Comments on the parameters used to facilitate the display when substituting the parameters with their values (see Program Generated)

`DEFINE %4.CONDITION`
`  AS %9`
`  TYPE INTEGER`
`  OPTIONS DEFAULT 0`
`  RANGE 0-1`
	Filter condition for counting the numerator items. Parameter %9 must be a Boolean expression returning 0 for false and 1 for true

`DEFINE %5.CNTNUM`
`  AS COUNT %4`
`  FOR %4.CONDITION = 1`
`  TYPE INTEGER`
`  VARLABEL "Target Population"`
	Counts the number of cases at the entity level of the parameter %5, filtering cases using the condition defined in the previous DEFINE

`DEFINE %5.CNTDENUM`
`  AS COUNT %4`
`  TYPE INTEGER`
`  VARLABEL "Universe Population"`
	Counts the total of cases

`DEFINE %5.RATIO`
`  AS 1000 * ( %5.CNTNUM/%5.CNTDENUM )`
`  FOR %5.CNTDENUM > 0`
`  TYPE REAL`
`  VARLABEL "Ratio (per 1000)"`
	Calculates ratio

`TABLE List  AS AREALIST`
`  OF %5, %5.LABEL, %5.CNTNUM, %5.CNTDENUM,`
`%5.RATIO 7.2`
`  OPTION OMITTITLE`
	Displays an output

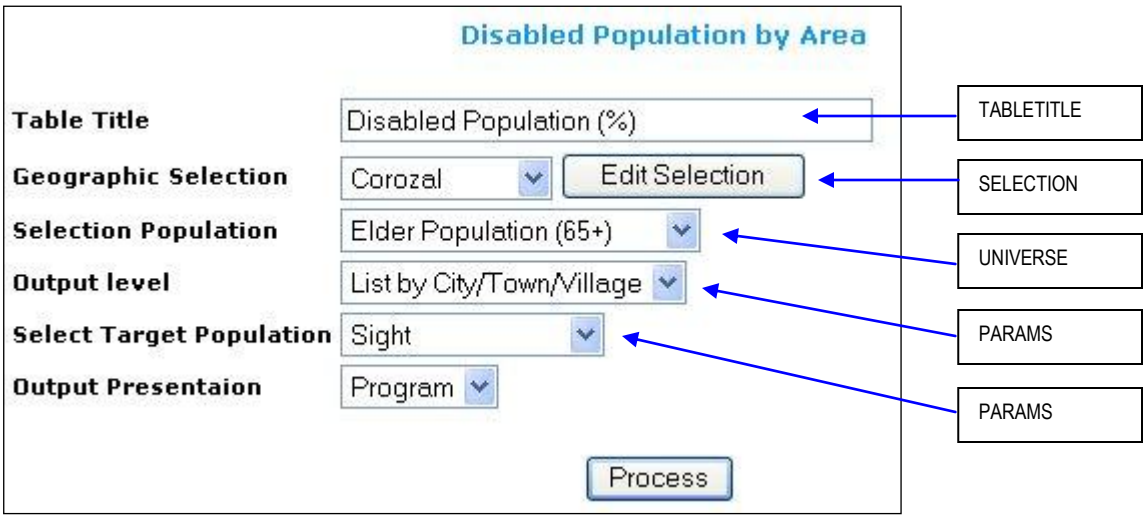


Figure 28. InputSPC Process

Program Generated

Contents	Comment

`RUNDEF DisabilityRatio`
`  SELECTION Corozal.sel`
`  UNIVERSE PERSON.AGE > 64`
	The selection was Corozal, and the universe of cases are 65+ year-old people (see Selection Population)

`* Corozal.sel Selection File`
`* UNIVERSE PERSON.AGE > 64 Universe Condition`
`* PERSON Entity counted`
`* CTV Entity Output`
`* PERSON.SIGHT=1 Target count`
	Comments to verify the values in the selected parameters
	%4 – PERSON
	%5 – CTV
	%9 - PERSON.SIGHT=1

`DEFINE PERSON.CONDITION`
`  AS PERSON.SIGHT = 1`
`  TYPE INTEGER`
`  OPTIONS DEFAULT 0`
`  RANGE 0-1`
	Parameter 4 is fixed for PERSON, and parameter 9 was selected as Sight = 1

`DEFINE CTV.CNTNUM`
`  AS COUNT PERSON`
`  FOR PERSON.CONDITION = 1`
`  TYPE INTEGER`
`  VARLABEL "Target Population"`
	The numerator is found by counting those persons that meet the condition defined in the previous DEFINE. The output level was selected as CTV

`DEFINE CTV.CNTDENUM`
`  AS COUNT PERSON`
`  TYPE INTEGER`
`  VARLABEL "Universe Population"`
	All persons are counted in the denominator

`DEFINE CTV.RATIO`
`  AS 1000 * ( CTV.CNTNUM / CTV.CNTDENUM )`
`  FOR CTV.CNTDENUM > 0`
`  TYPE REAL`
`  VARLABEL "Ratio (per 1000)"`
	The ratio between the numerator and denominator is calculated by multiplying the results by 1000, taking care that there is at least one person in the District (denominator > 0).

```
TABLE List AS AREALIST
   OF CTV, CTV.LABEL, CTV.CNTNUM, CTV.CNTDENUM,
   CTV.RATIO 7.2
   OPTION OMITTITLE
```

Type REAL to obtain decimal results
Program output in AREALIST with variables for numerator, denominator, and ratio. The ratio is 2 decimals long.
TITLE is omitted in the output

Output from this program is an Arealist as in Figure 29.

Code	District Name	Target Population	Universe Population	Ratio (per 1000)
1	Corozal	932	32,209	28.94
2	Orange Walk	975	38,060	25.62
3	Belize	1,254	63,061	19.89
4	Cayo	2,145	51,221	41.88
5	Stann Creek	518	24,443	21.19
6	Toledo	1,064	23,117	46.03

Download Excel format - Download PDF format

Figure 29. InputSPC Process

VI.12 Process: MultiFilter

Nodetype	**MULTIFILTER**	
Purpose	A process specific to databases with aggregated data, where generally indicators have already been calculated. To generate boxes and arealists allowing to map the indicators in a database with various user-selected breakdowns such as Urban/Rural, Sex, Age, Poverty Status.	
Operation	This page allows combining a tri-type selection and 4 additional filters. For the output the user may select a summary box, which is a crosstabs with up to 4 variables. (Generally they are Geographical Location, Period of Time, and Indicator). In this case the values correspond to the sum of the filters chosen by the user. For example, if the indicator were Persons of native background and the user had selected Poor, Males, Rural, and 15-29, and 30 years or more, then the Summary will display a column listing all those persons that recognize themselves as of native background, 15 years old or more, poor, and rural. If Table is selected as output, then the system will generate an arealist that can be associated to the map with a column for each selected condition, i.e., in the previous case a column would display for persons of 15-29 years old and another column for those of 30 and more.	
Particular Controls & Clauses	INDICLABEL TRIVAR1/2/3	Titles Indicator Selections (up to 3 variables)

	VAR	Variables in arealist	
	FILTER1/2/3/4	Filters for indicators (up to 4)	
	ROW	Row Variable	
	COL	Column Variable	
	CTL	Control Variable (3rd dimension)	
	PAN	Panel Variable (4th dimension)	
Common Controls		Filter	Format
		Selection	Graph
	Weight	TableTitle	Map
Common Clauses	Universe	AltFilter	Caption
	ProcessTitle	Footnote	Webmaster
Restrictions and Comments	-		

Description of Controls and Clauses

Control/Clause	Type	Comment
INDICLABEL	Text	Label of the indicator column in the Arealist table. Optional.
FILTERS	Integer	Number of existing filters

Control: Selection of Indicators (TRIVARi)

(Up to 3) comboboxes listing the options to select variables and their categories for the indicators. Shown here are only those clauses for the first TRIVAR (represented by number 1 in the clauses), but the same applies to the second and third TRIVARs (when substituting 1 for 2 and 3, respectively).

Control/Clause	Type	Comment
TRIVAR1WIDTH	Integer	Width of the Name box for Tri Variables
TRIVAR1VLWIDTH	Integer	Width of the Category box for Tri Variables
TRIVAR1CKCAPTION	Text	Text of the checkbox allowing the user to select any options for Tri. Optional.
TRIVAR1N	Integer	Number of Variables that may be placed in the first Tri. Must include the clauses TRIVAR11 up to TRIVAR1n.
TRIVAR1$_j$	Variable Name	Full name (entity.variable) of the variable that will be selected in Tri.

Control: Variables in Arealist (VAR)

This control has no visible parts. It's used to report which of the TRIVAR controls are to be displayed in what outputs, in case that the HTML output is selected (Arealist).

Control/Clause	Type	Comment
VARINDIC	RWord	Placed on TRIVAR1, TRIVAR2, or TRIVAR3, depending on which TRIVAR has been placed the **Indicator**.
VARCONTROL	RWord	Placed on TRIVAR1, TRIVAR2, or TRIVAR3, depending on which TRIVAR has been placed the **Control**.
VARFILTER	RWord	Placed on TRIVAR1, TRIVAR2, or TRIVAR3, depending on which TRIVAR has been placed the **Geographic Location.**
GRAPHVAR	RWord	Placed on TRIVAR1, TRIVAR2, or TRIVAR3, depending on which TRIVAR the **Graphs** are to be created. Generally this will be the **Geographic Location**.

Control: Filters for the Indicators (FILTER)

(Up to 4) comboboxes listing the options to define filter variables and their categories for the indicators. Shown here are only those clauses for the first FILTER (represented by number 1 in the clauses), but the same applies to the second, third, and fourth FILTER (when substituting 1 for 2, 3, and 4, respectively).

Control/Clause	Type	Comment
FILTER1WIDTH	Integer	Width (in pixels) of the Filter 1 box.
FILTER1VLTOTAL	Integer	Code of the category meaning totals.
FILTER1VLWIDTH	Integer	Width (in pixels) of the Filter 1 label.
FILTER1VLHEIGHT	Integer	Height (in lines) of the Filter 1 label.
FILTER1N	Integer	Number of Variables that may be placed in Filter 1. Must include clauses FILTER11 up to FILTER1n.
FILTER1$_j$	Variable Name	Full name (entity.variable) of the Variable that will be placed on Filter 1.

Control: Row Variable (ROW)

A combobox listing the options to define the output variables in rows, in case that the selected output is a summary.

Control/Clause	Type	Comment
ROWCAPTION	Text	Row box caption
ROWOMITTOTAL	RWord YES/NO	To decide whether totals per row are omitted or not. The default value is NO.
ROWN	Integer	Number of Variables that may be selected to show per row. Must include the clauses ROW1 up to ROWn.

ROW$_i$	Variable Name	Variable to be listed in the row.

Control: Variable for Column and 3rd Dimension (COL, CTL)

A combobox listing the options to define the output variables in columns and third dimension, in case that the selected output is a summary. The same ROW options apply as well for COL (Column), CTL (3rd dimension), PAN (4th dimension), and GRD (5th dimension).

Example

Youth Indicator database, taking as indicator the population size, regions as geographic area, and census years as the time period. Since the screen contains much information, it was divided into two, as shown in Figures 30 and 31. Figure 32 is the output from the example, when the options selected in the Figures are chosen.

Contents	**Comment**
`[TAMANIO]`	Section ID and node type
`NODETYPE=MultiFilter`	
`TRIVAR1N=2`	2 variables in TRIVAR1
`TRIVAR11=REGION.REGION`	Both variables in TRIVAR1
`TRIVAR12=COMUNA.COMUNA`	
`TRIVAR1WIDTH=35`	Width of the variable name box
`TRIVAR1VLWIDTH=25`	Width of the variable category box
`TRIVAR1CKCAPTION=Sel. All`	Text for the TRIVAR1 checkbox
`TRIVAR2N=1`	TRIVAR2 has only one variable.
`TRIVAR21=INDICA.ANCENSOS`	
`TRIVAR2WIDTH=10`	
`TRIVAR2VLWIDTH=10`	
`TRIVAR2CKCAPTION=Sel. All`	
`TRIVAR3N=1`	TRIVAR3 also has only one variable.
`TRIVAR31=INDICA.VARIA1`	
`TRIVAR3LOGIC=NONE`	
`TRIVAR3WIDTH=54`	
`TRIVAR3VLWIDTH=84`	
`TRIVAR3CKCAPTION=Sel. All`	
`VARINDIC=TRIVAR3`	Defines TRIVAR3 to be used as Indicator
`VARCONTROL=TRIVAR2`	Defines TRIVAR2 to be used as a control dimension
`VARFILTER=TRIVAR1`	Defines TRIVAR1 to be used as Filter
`GRAPHVAR=TRIVAR1`	Defines TRIVAR1 to be used in the Graph variable
`FILTERS=4`	There are 4 filters
`FILTER1WIDTH=29`	Width of Filter 1
`FILTER1N=1`	Only one variable in Filter 1, and its name
`FILTER11=INDICA.ZONA`	
`FILTER1VLWIDTH=20`	Width of the category box
`FILTER1VLHEIGHT=3`	The box is 3 lines in height

```
FILTER1VLTOTAL=3                    Total Category of Filter 1
FILTER2WIDTH=22                     Filter 2, with 2 variables
FILTER2N=2
FILTER21=INDICA.GEDAD
FILTER22=INDICA.GEDADJ
FILTER2VLWIDTH=40
FILTER2VLHEIGHT=3
FILTER3WIDTH=29                     Filter 3, with a single variable
FILTER3N=1
FILTER31=INDICA.SEXO
FILTER3VLTOTAL=3
FILTER3VLWIDTH=20
FILTER3VLHEIGHT=3
FILTER4WIDTH=29                     Filter 4, with a single variable
FILTER4N=1
FILTER41=INDICA.POBREZA
FILTER4VLTOTAL=3
FILTER4VLWIDTH=32
FILTER4VLHEIGHT=3
ROWCAPTION=On (row):                Row Variables
ROWN=3
ROW1=TRIVAR1
ROW2=TRIVAR2
ROW3=TRIVAR3
COLCAPTION=Crossed by (column):     Column Variables
COLOMITTOTAL=YES
COLN=3
COL1=TRIVAR2
COL2=TRIVAR3
COL3=TRIVAR1
CTLCAPTION=Control:                 3rd Dimension Variables
CTLOMITTOTAL=YES
CTLN=3
CTL1=TRIVAR3
CTL2=TRIVAR1
CTL3=TRIVAR2
```

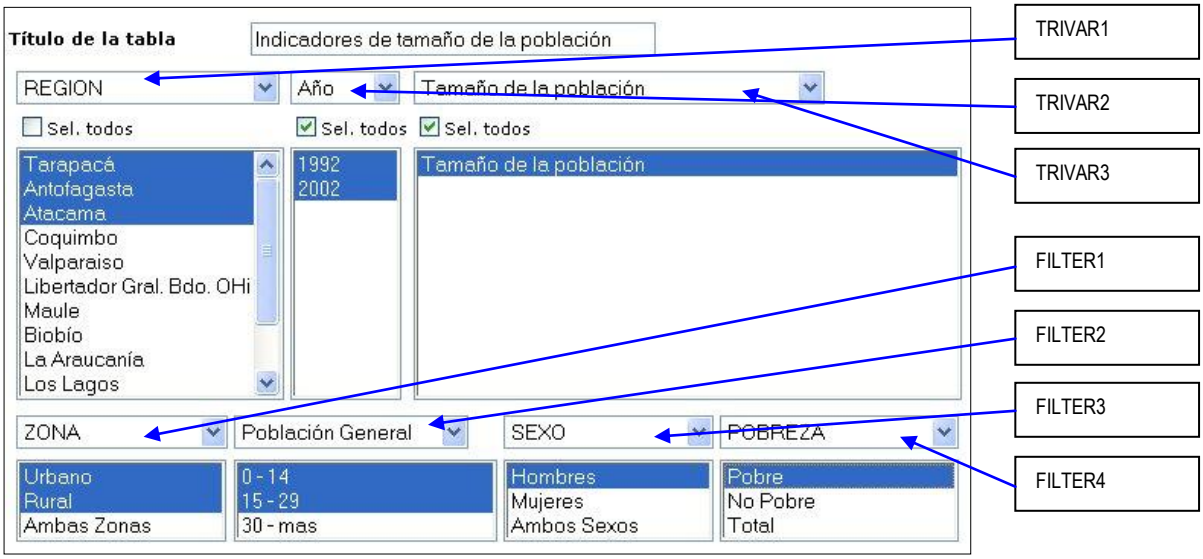


Figure 30. Multifilter Process (1st Part)

Figure 31. Multifilter Process (2nd Part)

Código	REGION	Tamaño de la población 1992 Urbano 0 - 14 Hombres Pobre	Tamaño de la población 1992 Urbano 15 - 29 Hombres Pobre	Tamaño de la población 1992 Rural 0 - 14 Hombres Pobre	Tamaño de la población 1992 Rural 15 - 29 Hombres Pobre	Tamaño de la población 2002 Urbano 0 - 14 Hombres Pobre	Tamaño de la población 2002 Urbano 15 - 29 Hombres Pobre	Tamaño de la población 2002 Rural 0 - 14 Hombres Pobre	Tamaño de la población 2002 Rural 15 - 29 Hombres Pobre
01	Tarapacá	25,665	22,894	2,520	3,491	21,027	19,780	2,384	4,216
02	Antofagasta	31,950	27,948	1,053	1,995	21,631	21,897	728	2,054
03	Atacama	19,796	15,381	2,329	3,181	16,274	13,183	2,155	2,442
TOTAL		77,411	66,223	5,902	8,667	58,932	54,860	5,267	8,712

Figure 32. Multifilter Output

VI.13 Process: QTS

Nodetype	**QTS**	
Purpose	This process is used to calculate a ratio indicator between the categories of a variable, as selected by the user, and the total of cases.	
Operation	The user selects the variable to be used for calculation, and then selects one or more categories that will be used as inputs for the ratio numerator. For the output level wanted, the program counts the cases in the numerator. It also counts all cases, and then divides the first item by the second. At the end the program executes a TABLE AREALIST using the indicator.	
Particular Controls & Clauses	QTS OUT	Process Variables Output Level Entities

	OPT	Display options	
Common Controls		Filter Selection	Format
	Weight	TableTitle	
Common Clauses	Universe ProcessTitle	AltFilter	Caption
Restrictions and Comments			

Description of Controls and Clauses

Controls below may also be displayed separately as sections. In this case, use as a clause the name of the control followed by the letter N and a section name, as in most controls. For example:

```
QTSN=QTS1_SECTION
OUTN=ENTITY1_SECTION
OPTN=OPTION1_SECTION
```

Control: Process Variables (QTS)

Control/Clause	Type	Comment
QTSCAPTION	Text	Text that goes with the variable box.
QTSN	Integer	Number of items in the variable box. Must include the clauses QTS1, QTS2, etc., up to QTSn
QTS_i	Variable name and text (optional)	Name of the variable to calculate the ratio. By default the system shows the variable label in the box, unless there is any descriptive text following the name of the variable in the clause.
QTSLAST	Integer	Number of the variable that will be displayed in the box when the system shows the page. The default value is 1
$QTS_i.VL_j$	RWord YES	To preselect the category j (valuelabel) in variable i in the list, when the variable has been selected by the user or is the variable in QTSLAST upon the initial display of the page.

Control: Output Level Entities (OUT)

Control/Clause	Type	Comment
OUTCAPTION	Text	Text that goes with the entity box.
OUTN	Integer	Number of items in the entity box. Must include the clauses OUT1, OUT2, etc., up to OUTn.
OUT_i	Entity name and text (optional)	Name of the output level entity. The entity label appears in the entity box and may be replaced with the (optional) text following the entity name.

Control: Display Options (OPT)

Control/Clause	Type	Comment
OPTCAPTION	Text	Text that goes with the display option box.
OPTTOT	Text	This clause has two functions: 1) indicates that the option Total will display in the box; and 2) shows the text to be displayed in the Total row. For displaying the ratio denominator (D).
OPTSEL	Text	The same applies to the selected categories of the variable, i.e., the ratio numerator (N).
OPTPC	Text	The same applies to percentage, i.e. the ratio itself, i.e. R = N / D.
OPTPCR	Text	The same applies to a relative percentage, which is the calculation of the numerator divided by the categories not selected, i.e., P = N / (D – N).
OPTTOTSEL **OPTSELSEL** **OPTPCSEL** **OPTPCRSEL**	RWord YES/NO	To automatically select the display option when displaying the page. If desired, the user may deselect this option at runtime. The default value is NO.

Example

Displays a page with the variable box and a list of categories for the variable shown. Displays the output entity box, and then a list of display options. Figure 33 shows a sample page. Below the figure are some comments on the example.

Contents	**Comment**
`NODETYPE=QTS`	Node Type
`QTSCAPTION=Select one or more conditions:`	Variable box caption
`QTSN=4`	Number of variables
`QTS1=PERSON.EDQUINQ Age in groups of 5 years`	Variables in the list, the first one with a changed label for display
`QTS2=PERSON.ECIVIL`	
`QTS3=PERSON.CURSO`	
`QTS4=PERSON.ALFAB`	
`QTSLAST=2`	Automatically display the second variable (ECIVIL)
`QTS2.VL2=YES`	Preselect the second and third categories of the second variable
`QTS2.VL3=YES`	
`OUTCAPTION=Output Level:`	Output level entities
`OUTN=3`	
`OUT1=COMUNA`	
`OUT2=DISTRITO`	
`OUT3=AREA`	
`OPTCAPTION=Options:`	Displays ratio only
`OPTTOT=Total`	Enables 4 display options
`OPTSEL=Selected Total`	
`OPTPC=Percentage`	

```
OPTPCR=Relative Percentage

OPTSELSEL=YES                                    Preselects all display options
OPTTOTSEL=YES
OPTPCSEL=YES
OPTPCRSEL=YES
```

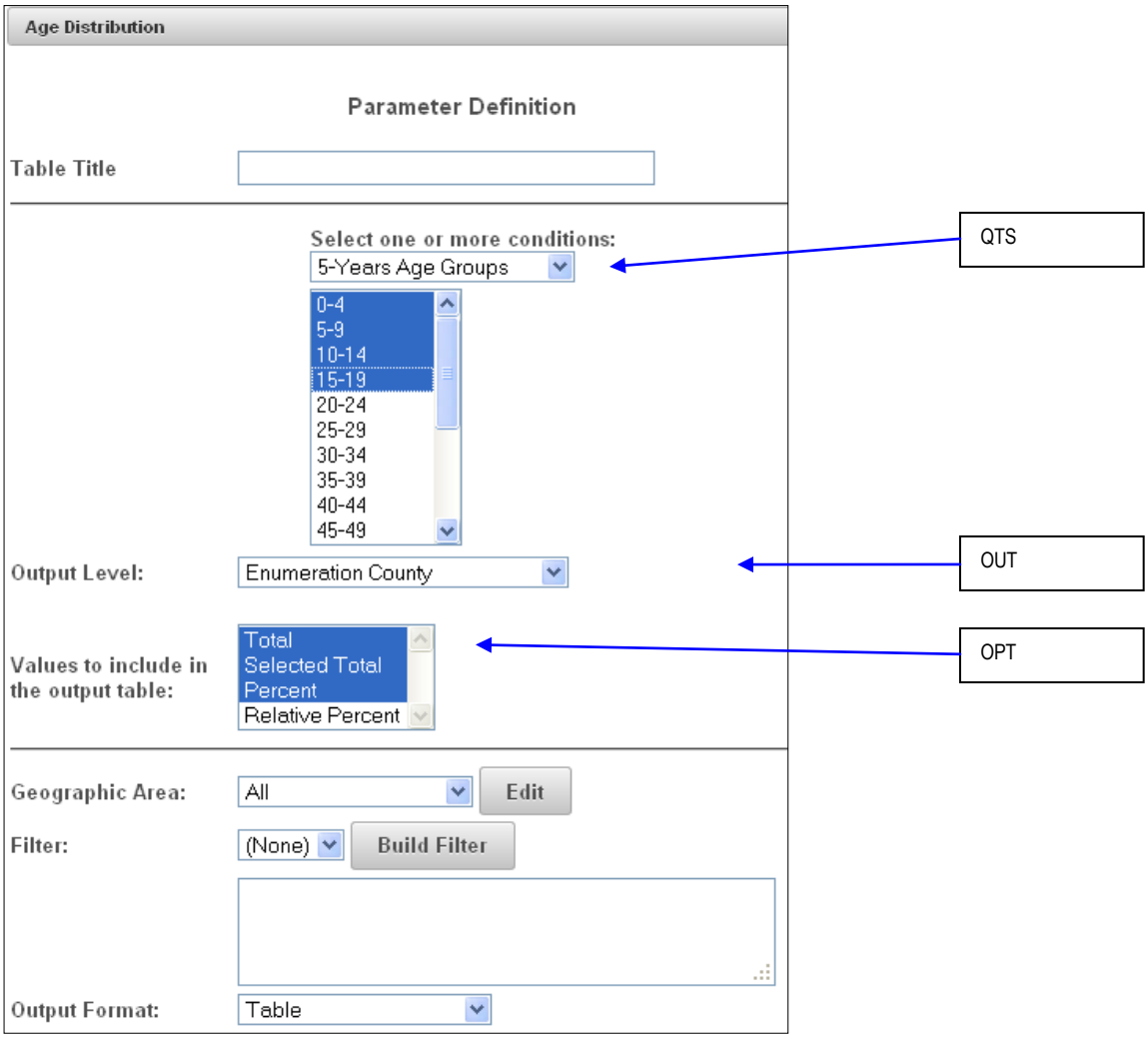
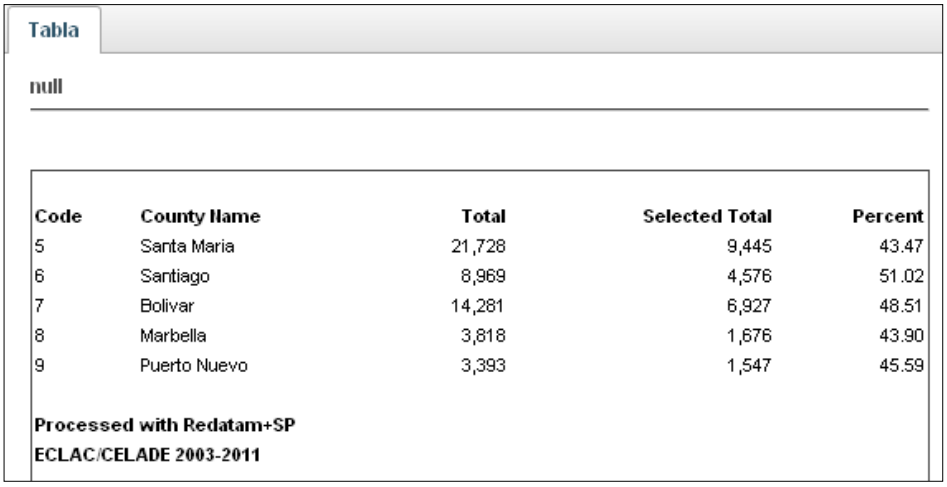


Figure 33. QTS Process

This sample Figure starts with a CAPTION. Then the variable box is shown with the variable “5-Year Age Groups” selected, and the categories “0-4”, “5-9”, “10-14” and “15-19” highlighted. The selected output level is Enumeration County, and all display options have been selected. Then the geographic and output format selections and the Execute button are displayed. When pressing the Execute button, the result is displayed as in Figure 34.



Tabla

null

Code	County Name	Total	Selected Total	Percent
5	Santa Maria	21,728	9,445	43.47
6	Santiago	8,969	4,576	51.02
7	Bolivar	14,281	6,927	48.51
8	Marbella	3,818	1,676	43.90
9	Puerto Nuevo	3,393	1,547	45.59

Processed with Redatam+SP
ECLAC/CELADE 2003-2011

Figure 34. QTS Output

VI.14 Process: Sex Ratio (SexRatio)

Nodetype	**SEXRATIO**		
Purpose	This process is used to calculate the Gender Ratio of the population.		
Operation	For each item at the output level, the program calculates the male and female populations, divides males by females, and then executes a TABLE AREALIST using the Gender Ratio.		
Particular Controls & Clauses	SEXVAR ROW OUTPUT	Definition of the sex variable Output entities Display options	
Common Controls	Weight	Filter Selection TableTitle	Format
Common Clauses	Universe ProcessTitle	AltFilter	Caption
Restrictions and Comments	-		

Description of Controls and Clauses

Controls below may also be displayed separately as sections. In this case, use as a clause the name of the control followed by the letter N and a section name, as in most controls. For example:

```
SEXVARN=SEX1_SECTION
ROWN=ENTITY1_SECTION
OUTPUTN=OUTPUT1_SECTION
```

Control: Definition of the Sex Variable (SEXVAR)

Control/Clause	Type	Comment
SEXVAR	Variable Name	Variable in the database (or calculated in DEFINEs) with data on persons' sex.
MALECODE	Integer	Code for males in the SEXVAR variable
FEMALECODE	Integer	Code for females in the SEXVAR variable

Control: Output Entities (ROW)

Control/Clause	Type	Comment
ROWCAPTION	Text	Text that goes with the entity box.
ROWN	Integer	Number of items in the entity box. Must include the clauses ROW1, ROW2, etc., up to ROWn
ROW$_i$	Entity name and text (optional)	Name of the output level entity. The entity label appears in the entity box and may be replaced with the (optional) text following the entity name.

Control: Display Options (OUTPUT)

Control/Clause	Type	Comment
OUTPUTCAPTION	Text	Text that goes with the display option box.
OUTPUTN	Integer	Number of items in the display box. Must include the clauses OUTPUT1, OUTPUT2, etc., up to OUTPUTn
OUTPUT$_i$	RWord & Text OFF COUNTS	Reserved word symbolizing the output type to be displayed, and a descriptive text which will be displayed in the box on screen. Options include: Only sex ratio Also includes absolute values

Example

Displays a page with the filter boxes and the output entity box. Figure 35 shows a sample page.

Contents	**Comment**
`SEXVAR=PERSONA.SEXO`	Sex Variable
`MALECODE=1`	Male & female codes
`FEMALECODE=2`	
`ROWN=3`	Number of entities
`ROWCAPTION=Output Level`	Entity box caption
`ROW1=DISTRITO District`	Output entities
`ROW2=CTV Town/City`	
`ROW3=ED Enumeration Sector`	

`OUTPUTN=4`	Number of output options
`OUTPUTCAPTION=Output Options`	Option box caption
`OUTPUT1=OFF Ratio Only`	Displays ratio only
`OUTPUT2=COUNTS Include absolute values`	Displays the numerator, denominator, and ratio values

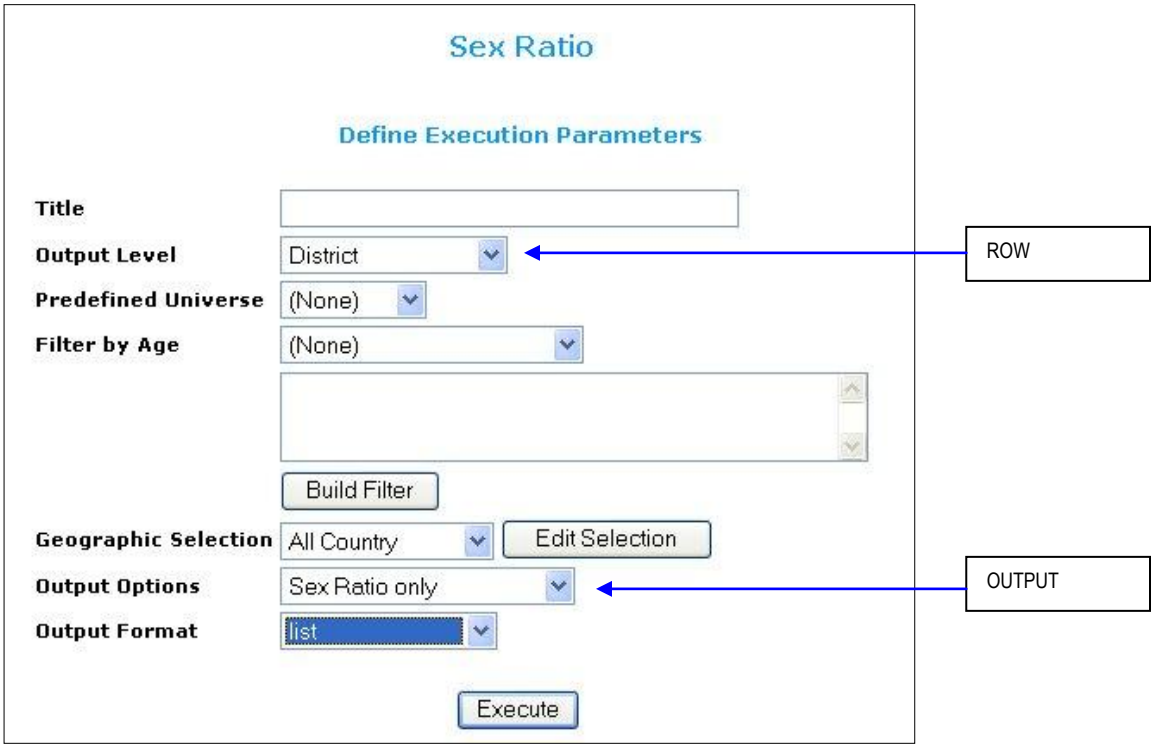


Figure 35. SexRatio Nodetype

This sample Figure starts with a CAPTION. Then the universe box (optional) and an Age filter (optional) are shown. Then the geographical selection, the output options and output format, and the Execute button. If District is selected as the output entity and the Execute button pressed, the results display as in Figure 36.

Code	District Name	Sex Ratio
1	Corozal	1.01
2	Orange Walk	1.05
3	Belize	0.97
4	Cayo	1.00
5	Stann Creek	1.08
6	Toledo	1.01

Download Excel format - Download PDF format

Figure 36. SexRatio Output

VI.15 Process: TriRecode

Nodetype	**TRIRECODE**
Purpose	A process specific to databases with aggregated data, where generally indicators have already been calculated. To generate boxes and arealists allowing mapping the indicators of a database with various user-selected breakdowns such as Urban/Rural, Sex, Age, Poverty Status, etc.
Operation	This is very similar to the Multifilter process in terms of TRI, but it doesn't include a filter portion. For the output the user may select a summary box, which is a crosstabs with up to 4 variables (generally they are geographic location, period of time, and indicator). In this case the values will correspond to the sum of the filters selected by the user. For example, if the indicator were Persons of native background and the user had selected Poor, Males, Rural, and 15-29, and 30 years or more, then the Summary will display a column listing all those persons that recognize themselves as of native background, 15 years old or more, poor, and rural. If the Total box output is selected, the system will generate an arealist that can be associated to the map with a column for each selected condition, i.e., in the previous case a column would display for persons of 15-29 years old and another column for those of 30 and more.

Particular Controls & Clauses	INDICLABEL	Titles
	TRIVAR1/2/3	Indicator Selections (up to 3 variables)
	VAR	Variables in arealist
	ROW	Row Variable
	COL	Column Variable
	CTL	Control Variable (3rd dimension)
	PAN	Panel Variable (4th dimension)

Common Controls		Filter	Format
		Selection	Graph
	Weight	TableTitle	Map
Common Clauses	Universe	AltFilter	Caption
	ProcessTitle	Footnote	Webmaster

Restrictions and Comments	-

Description of Controls and Clauses

Control/Clause	Type	Comment
INDICLABEL	Text	Label of the indicator column in the Arealist table. Optional.

Control: Selection of Incicators (TRIVARi)

(Up to 3) comboboxes listing the options to select variables and their categories for the indicators. Shown here are only those clauses for the first TRIVAR (represented by number 1 in the clauses), but the same applies to the second and third TRIVARs (when substituting 1 for 2 and 3, respectively).

Control/Clause	Type	Comment
TRIVAR1WIDTH	Integer	Width of the Name box for Tri Variables
TRIVAR1VLWIDTH	Integer	Width of the Category box for Tri Variables
TRIVAR1CKCAPTION	Text	Text of the checkbox allowing the user to select any options for Tri. Optional.
TRIVAR1N	Integer	Number of Variables that may be placed in the first Tri. Must include the clauses TRIVAR11 up to TRIVAR1n.
TRIVAR1$_j$	Variable Name	Full name (entity.variable) of the variable that will be selected in Tri.

Control: Variables in Arealist (VAR)

This control has no visible parts. It's used to report which of the TRIVAR controls are to be displayed in what outputs, in case that the HTML output is selected (Arealist).

Control/Clause	Type	Comment
VARINDIC	RWord	Placed on TRIVAR1, TRIVAR2, or TRIVAR3, depending on which TRIVAR has been placed the **Indicator**.
VARCONTROL	RWord	Placed on TRIVAR1, TRIVAR2, or TRIVAR3, depending on which TRIVAR has been placed the **Control**.
VARFILTER	RWord	Placed on TRIVAR1, TRIVAR2, or TRIVAR3, depending on which TRIVAR has been placed the **Geographic Location.**
GRAPHVAR	RWord	Placed on TRIVAR1, TRIVAR2, or TRIVAR3, depending on which TRIVAR the **Graphs** are to be created. Generally this will be the **Geographic Location**.

Control: Row Variable (ROW)

A combobox listing the options to define the output variables in rows, in case that the selected output is a summary.

Control/Clause	Type	Comment
ROWCAPTION	Text	Row box caption

ROWOMITTOTAL	RWord YES/NO	To decide whether totals per row are omitted or not. The default value is NO.
ROWN	Integer	Number of Variables that may be selected to show per row. Must include the clauses ROW1 up to ROWn.
ROW$_i$	Variable Name	Variable to be listed in the row.

Control: Column and Third Dimension Variable (COL, CTL)

A combobox listing the options to define the output variables in columns and third dimension, in case that the selected output is a summary. The same ROW options apply to COL (column), CTL (3rd dimension), and PAN (4th dimension).

Example

The Database of the International Conference on Population and Development, Cairo 1994 (ICPD), where the population by sex is selected as the indicator, two countries as geographic area, and census years as the time period. The screen is shown in Figure 37. Figure 38 is the output from the example when the options shown in the Figure have been selected.

Contents	**Comment**
`[POBLA1]`	Section ID
`NODETYPE=TriRecode`	Node Type
`TRIVAR1CKCAPTION=Sel. All`	Display the checkbox for all TRI, with the specified text
`TRIVAR2CKCAPTION=Sel. All`	
`TRIVAR3CKCAPTION=Sel. All`	
`TRIVAR1NVAR=1`	
`TRIVAR11=PAIS.Pais6`	
`TRIVAR1WIDTH=45`	
`TRIVAR1VLWIDTH=25`	
`TRIVAR2NVAR=1`	
`TRIVAR21=PERIODO.Periodo5`	
`TRIVAR2WIDTH=10`	
`TRIVAR2VLWIDTH=13`	
`TRIVAR3NVAR=2`	
`TRIVAR31=VARIABLE.Varia3`	
`TRIVAR32=VARIABLE.Varia42`	
`TRIVAR3WIDTH=54`	
`TRIVAR3VLWIDTH=44`	
`TRIUNIVERSE=PAIS.Pais2<38 AND (VARIABLE.Cairo<>0)`	
`FILTERS=0`	
`ROWCAPTION=On (row):`	
`ROWTOP=30`	
`ROWLAST=1`	
`ROWNVAR=3`	
`ROW1=PAIS.Pais6`	
`ROW2=PERIODO.Periodo5`	
`ROW3=TRIVAR3`	
`COLCAPTION=Crossed by (column):`	
`COLTOP=60`	
`COLNVAR=3`	

```
COLLAST=1
COL1=PERIODO.Periodo5
COL2=TRIVAR3
COL3=PAIS.Pais6
CTLCAPTION=Control:
CTLTOP=90
CTLNVAR=3
CTLLAST=1
CTL1=TRIVAR3
CTL2=PAIS.Pais6
CTL3=PERIODO.Periodo5
MAP=YES
MAPENTITY=PAIS
VARINDIC=TRIVAR3
VARCONTROL=TRIVAR2
VARFILTER=TRIVAR1
```

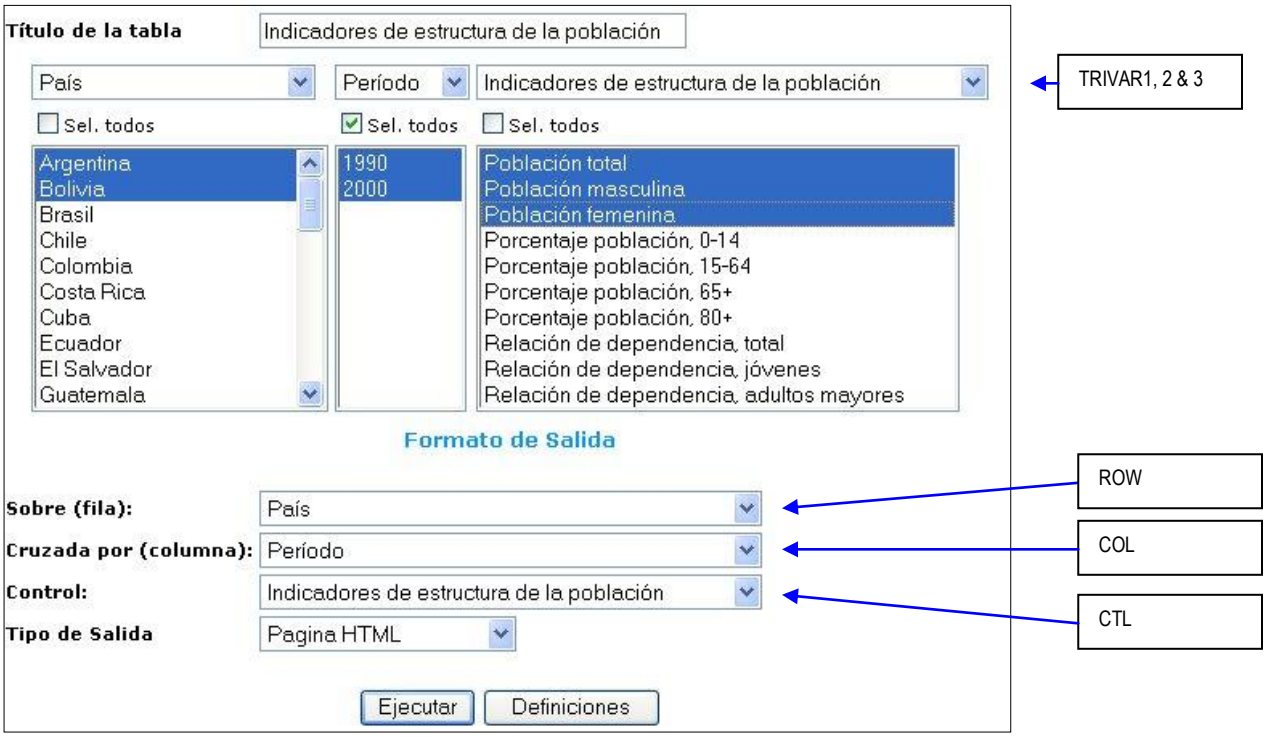
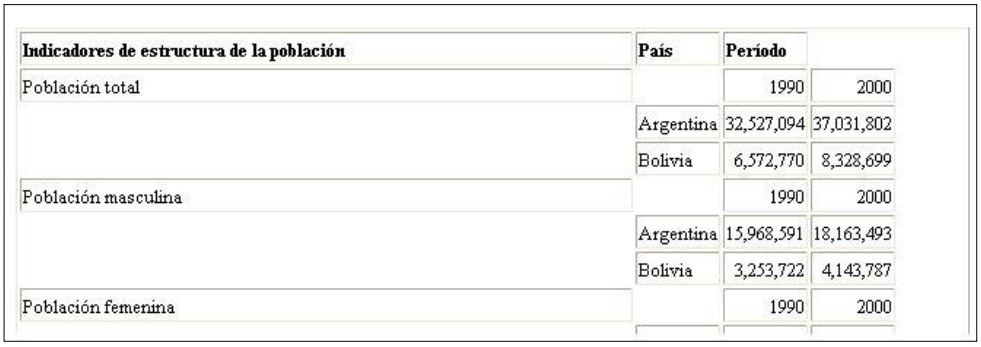


Figure 37. TRIRECODE Process



Indicadores de estructura de la población	País	Período	
Población total		1990	2000
	Argentina	32,527,094	37,031,802
	Bolivia	6,572,770	8,328,699
Población masculina		1990	2000
	Argentina	15,968,591	18,163,493
	Bolivia	3,253,722	4,143,787
Población femenina		1990	2000

Figure 38. TRIRECODE Output

VII. Auxiliary Nodes

The first four types of nodes (*, Group, Database, and DisplayHTML) are the only nodes that may be used in the WebServerMain.INL file. The other nodes may be used only in the INLs of the Guest type (that are triggered by WebServerMain). Nodes are identified by the clause NODETYPE.

VII.1 Nodetype *

This type of node is used to display a title in the Index, in order to organize a list of entries. The difference between this type and the GROUP type is that * organizes entries in the Index, while GROUP organizes entries in a list displayed in the OUTPUT. This type has only the CAPTION clause to display in the Index.

Control/Clause	Type	Comment
NODETYPE	Symbol	*
CAPTION	Text	Text to be displayed in the Index.

Example

Figure 39 below is a part of the Index built using sections with Nodetype =* that are shown here. This list displays in the Index panel, on the left portion of the screen.

Contents	**Comment**
`NODE17=OTHERGRP`	This node points to the [OTHERGRP] section below
`NODE18=NMIRGROUP`	This node points to the [NMIRGROUP] section below
`NODE19=*`	Shows a blank line
`NODE20=ONLINEGROUP`	
`NODE21=*`	
`NODE30=SPECIALGROUP`	
`NODE31=*`	
`[OTHERGRP]`	Shows only the title
`NODETYPE=*`	
`CAPTION=Other Databases`	
`[NMIRGROUP]`	A group of entries to be shown in the Output panel (when NODETYPE=GROUP)
`NODETYPE=GROUP`	
`CAPTION=Nueva Miranda`	
`NODES=3`	
`NODE1=NMIRESP`	
`NODE2=NMIRENG`	
`NODE3=NMIRPRT`	

Figure 39. Results from NODETYPE *

VII.2 BASE Nodetype

This type of node is used to connect to a INL file type that will take control to call a REDATAM database.

Control/Clause	Type	Comment
NODETYPE		BASE
CAPTION	Text	Text to be displayed in the list
INL	Filename	Name of the file that will take control of the program. Directory markers may be used.

Contents	Comment
`[NMIRENG]`	Section ID
`NODETYPE=BASE`	Node Type
`CAPTION=English`	Node Text
`INL=RpBases\NMIR\NMIRANDACENSO_ENG.inl`	Guest file that will take control in case that this option is selected

VII.3 CmdSet Nodetype

This type of node is used to directly program in REDATAM, by writing commands in SPC language. When triggered, the node shows Figure 40, where two empty boxes display to the left, and one box displays to the right with the database dictionary. The user may write commands in the top box and then press the Execute button.

Control/Clause	Type	Comment
NODETYPE		CMDSET
CAPTION	Text	Text to be displayed in the list
TITLE	Text	Process title in the OUTPUT page. If there is no such clause, then the title will be the contents of CAPTION. In order to delete a title from the output, this clause must be set to null (TITLE=).

Example

Displays the screen in Figure 40.

Contents	**Comment**
`[PROGRED]`	Section ID
`NODETYPE=CmdSet`	Node Type
`CAPTION=Redatam Statistical Processor`	Node Text
`TITLE=Write Redatam  Program`	Title displayed on screen

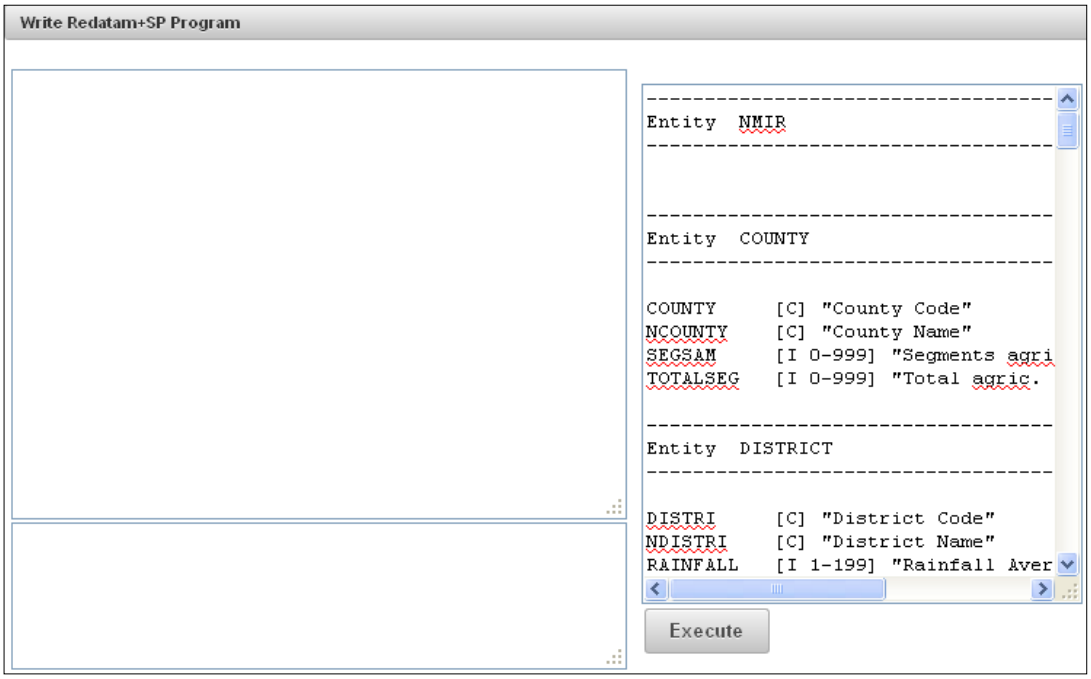


Figure 40. NODETYPE CMDSET

VII.4 STRUCTURE Nodetype

This type of node is used to display a list of nodes in the Index page. This is similar to the GROUP NODETYPE, except that GROUP shows the list in the Output panel, while STRUCTURE shows the list in the Index panel.

Control/Clause	Type	Comment
NODETYPE		STRUCTURE
CAPTION	Text	Text to be displayed as a list title in the Index
NODES	Integer	Maximum number of nodes.
NODE$_i$	Section Name	Name of a section or asterisk (“*”). If an asterisk, then a blank line appears in the list.

VII.5 DATABASE Nodetype

This type of node is intended to define the database to be used in the Guest file.

Control/Clause	Type	Comment
NODETYPE		DATABASE
NAME	Filename	Name of the database dictionary. You may use the “% INLPATH” keyword to signify the directory where the Guest file resides. Example: **NAME=%INLPATH\BaseR\NmirEsp.dic**

VII.6 DisplayHTML Nodetype

This type of node is used to display the contents of a file in html format, a page from a remote site, or to connect to a REDATAM server in a different site

Control/Clause	Type	Comment
NODETYPE		DISPLAYHTML
CAPTION	Text	Text to be displayed in the list
FILENAME	Filename	Name of the file to be displayed. It may be an HTM type file, an Internet address, or an INL parameter file to be used by the Webserver in the server site (see examples shown above in GROUP NODETYPE)
NEWPAGE	RWord YES/NO	A parameter to open a new page in the server (NEWPAGE=YES). By default it's shown in the same page.

VII.7 Download Nodetype

To download a file to the user's machine. It may be any type of file, e.g., a REDATAM data dictionary, a Word document, a text file, a PDF format document, etc.

Control/Clause	Type	Comment
NODETYPE		DOWNLOAD
CAPTION	Text	Text to be displayed in the list
FILENAME	Filename	Full name of the file to be downloaded. It may contain directory markers.

VII.8 GROUP Nodetype

This type of node is used to display an entry structure that will show in the Output panel on screen (to the right of the Index).

Control/Clause	Type	Comment
NODETYPE		GROUP
CAPTION	Text	Text to be displayed in the Index. Also displayed as title of the item list in the OUTPUT
NODES	Integer	Maximum number of nodes. Below there must be the clauses NODE1, NODE2, up to the maximum number of nodes.[13]
NODE$_i$	Section Name	Name of a section or asterisk ("*"). If an asterisk, then a blank line appears in the list.

Example

The right section in Figure 41 below was produced using the following commands

Contents	Comment
`[ONLINEGROUP]`	Section ID
`NODETYPE=GROUP`	This section produces the entry noted at the bottom left of the Figure, in the Index list.
`CAPTION=Sitios Online`	When the "Online Sites" title is clicked, the program displays a group list to the right, in the Output panel.
`NODES=34`	
`NODE1=AMLAT`	
`NODE2=*`	
`NODE11=OLBOL`	
`NODE12=OLCHL`	
`NODE13=OLCRI`	
`[AMLAT]`	
`NODETYPE=*`	
`CAPTION=Latin American Countries`	

[13] Just like in the WebServerMain list, numbers may be omitted without effecting the program's execution, but the execution will be slower because the program will be looking for nonexistent NODEs.

```
[OLBOL]
NODETYPE=DisplayHtml
CAPTION=Bolivia - 2001 Population and
Housing Census
FILENAME=http://www.../PortalAction?&MODE=MAIN&BASE=TallCreac&MAIN=WebS
erverMain.inl
NEWPAGE=YES
[OLCHL]
NODETYPE=DisplayHtml
CAPTION=Chile - 2002 National Population and Housing Census - Regional
Level
FILENAME=http://.../PortalAction?&MODE=MAIN&BASE=CPCHL2KREG&MAIN=WebSer
verMain.inl
NEWPAGE=YES
[OLCRI]
NODETYPE=DisplayHtml
CAPTION=Costa Rica - Population and Housing Censuses, Homes &
Indicators Survey
FILENAME=http://www.../REDATAM/01REDATAMSet.htm
NEWPAGE=YES
```

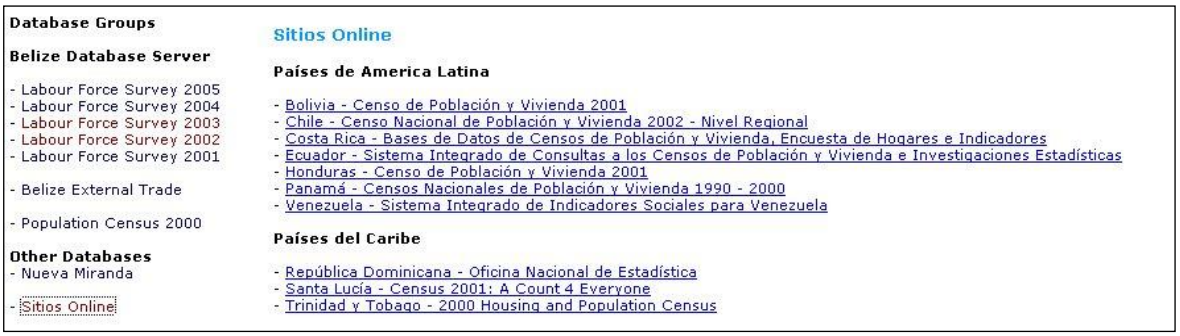


Figure 41. Results from the GROUP Nodetype

VII.9 Map Nodetype

This node is used to define a map section that may be displayed using an output format of the MAP type. In order for them to work, MAP type nodes must be included in the appropriate list, i.e., in any MAPi clause with the section name.

Control/Clause	Type	Comment
NODETYPE		MAP
CAPTION	Text	Text to be displayed in the Index. Also displayed as title of the item list in the OUTPUT
MXP	Filename	Name of the map definition file that will control the display of the map in the output. It may contain directory markers. Example: **MXP=%INLPATH\MAPAS\AMAZONAS-91.mxp** That file is built by the REDATAM Process module, and shall be connected to DBF and SHP

		files with the map items and polygon definition.
MAPSECTION	RWord	Always fixed **MAPSECTION=MAP**
IMAGE	RWord	Always fixed **IMAGE=CUSTOM**
HEIGHT	Integer	Map height
WIDTH	Integer	Map width

Example

Figure 42 below was produced by a QTS type node, executed at the "Enumeration County" output level, and using "Antioquia" as geographic selection.

Contents	**Comment**
`[ANTIOQUIA-05]`	Section ID
`NODETYPE=MAP`	
`CAPTION=Colombia : Antioquia Enumeration Counties`	
`MXP=%INLPATH\MAPAS\ANTIOQUIA-05.mxp`	Map definition file
`MAPSECTION=MAP`	
`IMAGE=CUSTOM`	
`HEIGHT=430`	Map Height & Width
`WIDTH=500`	

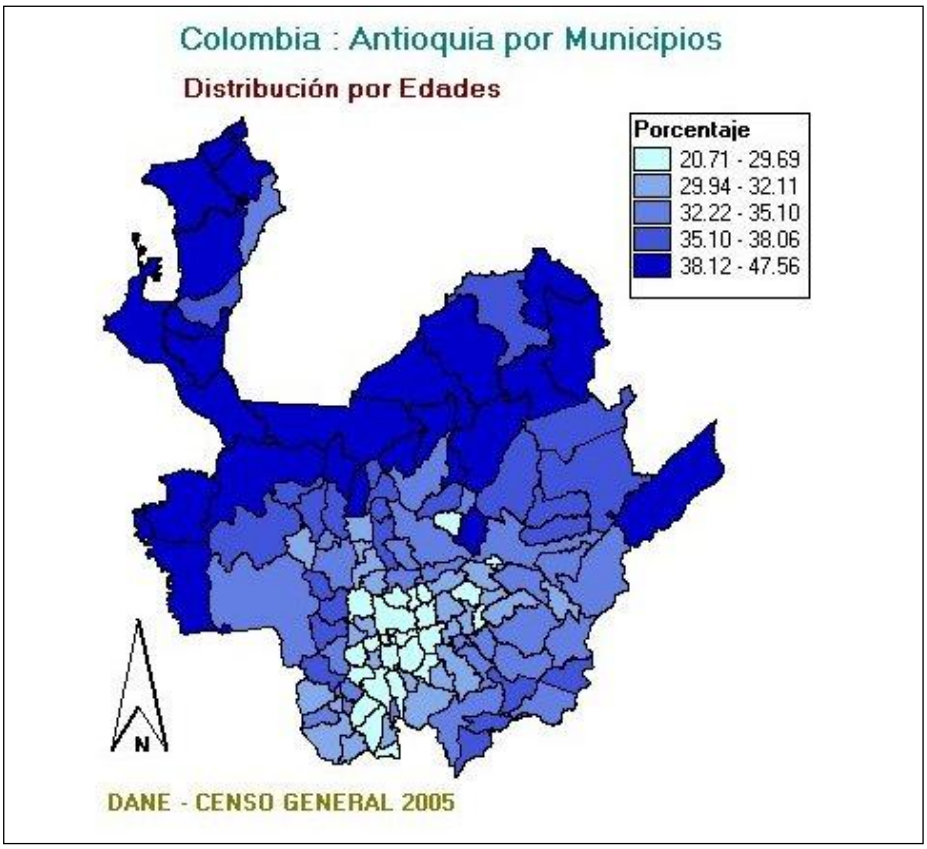


Figure 42. Map of the Antioquia Department

VII.10 NodeStyle Nodetype

This node is used in clauses having the same name in the processes, in order to define a series of common values to be used by all processes. For example, instead of defining for each process the clause values that will always be the same, such as ROWCAPTIONTOP, ROWCAPTIONLEFT, etc., they are defined only once in the Style node. This also has the advantage that, if anything must be changed in all processes in terms of positioning any controls, then this can be done in the style node, and all controls using this style will be modified.

Another advantage of using NODESTYLE type nodes is that the Peripheral Properties, i.e., the less significant clauses in the Control, can be taken out and collected in this node. Thus, the Process programming is made to look 'cleaner' by showing only the Main Properties, that in fact define the execution of the program.

A section stated as NODESTYLE may contain any Property (clause) from any Control (Proprietary or Common) from any Process. To invoke a style, you just have to state the NODESTYLE=section_name clause in the main section

Restrictions and Comments

The name of a Style section must be xxxx.DEFAULT

Example

Typical NODESTYLE for a Frequency Process

Contents	**Comment**
[FREQVIV]	Main Frequency node, where the FREQUENCY.DEFAULT section is stated as style
CAPTION=Housing Variables	
NODETYPE=FREQUENCY	
NODESTYLE=FREQUENCY.DEFAULT	
[FREQUENCY.DEFAULT]	Section ID
NODETYPE=NODESTYLE	Node type
INDICHEIGHT=150	Height of the variable list to extract the frequency
ROWCAPTION=By Row:	Variable box caption
ABKCAPTION=Area Break:	Areabreak box caption

VIII. Common Controls

These types of Controls may be used in any Main Processes (Frequency, Cruz, etc.). In fact, they are entirely standard sections but are called Controls because their clauses are always associated to a common topic, e.g. geographic selections or filters. They are not considered Proprietary Controls for the Main Processes because, first, they are used in several Processes, and second, they don't generate results by themselves, but rather support the Proprietary Controls for each Process.

VIII.1 Areabreak

The objective of these controls (there may be several of them in a Guest file) is to define the entities that will generate the potential area breaks (AREABREAK) to be used in the Processes. The name of an AREABREAK section must have been referenced by an ABKN clause in the Process. An AREABREAK may be used in two ways: first, by setting it as fixed, i.e., results will ALWAYS be broken down by the entity. The second way is by setting it as an entity list in a box, so that the user may select which of those entities will be used as area break.

VIII.1.a Fixed Areabreak

In this case, the information to be entered is the name of the entity, and the single clause is ABKFIXED. The break entity would not display in the process.

Control/Clause	Type	Comment
ABKFIXED	Entity Name	Name of the entity to be used as area break.

Example: Always Fixed Area Break

Contents	**Comment**
`ABKFIXED=DISTRICT`	Results will always be displayed with the area break at the district level, and the control will not be shown on screen

Note

This clause may be directly placed in the main execution processes (Frequency, Cruz, etc.). There is no need to state

```
ABKN=ABKCLAUSULA
```

in the main process, and then to create a section

```
[ABKCLAUSULA]
```

```
ABKFIXED=DISTRICT
```

VIII.1.b Varying Areabreak

In this case, the information that must be entered is the list of entities that may be used. They may be several controls of the Areabreak type.

Control/Clause	Type	Comment
CAPTION	Text	Text to be shown with the entity box
ABKN	Integer	Number of entities (items) in the list to be shown in the box. There must be the clauses ABK1, ABK2,…, up to ABKn, where *n* is the number in ABKN.
ABK_i	Entity Name	Name of the entity to be used as a break. Optionally, text between brackets may be used to signify that an area break won't be used in calculation.

Example: Entity List

In this example no breaks may be selected, or breaks by Enumeration County, District, or Area[14].

Contents	**Comment**
`[ABK_1]`	Section ID
`ABKCAPTION=Area Break:`	Areabreak box caption
`ABKN=4`	Four break options
`ABK1=(No%20Break)`	As a rule, if one of the items in the list begins with brackets, then the system assumes that that Control won't be used. That is to say that if the user chooses that option, then the results will have no area breaks
`ABK2=COMUNA`	
`ABK3=DISTRITO`	
`ABK4=AREA`	
`ABKLAST=2`	Shows the break by Enumeration County at the start

VIII.2 Filters

[14] The %20 symbol is to leave a non breaking space in the 'no break' title, which is not shown on screen but used to connect both words, so that only one word is displayed for the program.

This control is used so that the user may select the cases to be processed, according to a predefined logical expression. The name of the FILTER section must have been referenced by a FILTER clause in the process.

Filter options are shown in a box with a dynamic list of the “combobox” type. The logical expression may also be defined just before executing the process, by using a text editor with a wizard for constructing expressions in Redatam .

Control/Clause	Type	Comment
CAPTION	Text	Text to be shown along with the filter dynamic box
FILTERN	Integer	Number of expressions (items) in the list to be shown in the box. There must be the clauses FILTER1, FILTER2, …, up to FILTERn, where *n* is the number in FILTERN.
FILTER$_i$	Text	Logical expression (without blanks), followed by a text describing the expression (to be displayed in the filter box). Alternatively it may contain text in brackets meaning ‘no filter', and the text will be displayed in the filter box list.

Example

Contents	**Comment**

`[FILTER_1]`
`FILTERN=3`
`FILTERCAPTION=Filter:`
`FILTER1=(None)`
`FILTER2=AREA.URBRUR=1 Urban`
`FILTER3=AREA.URBRUR=2 Rural`
	Section ID
	Displays a dynamic box (combobox) listing 3 entries.

The screen with the “open” box (after clicking on the arrow) can be seen in Figure 43.

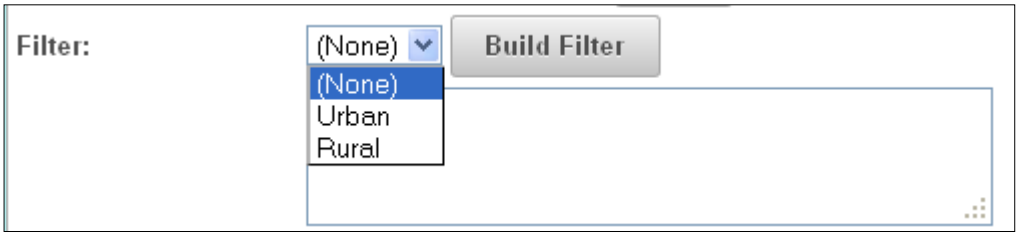


Figure 43. Filter type control

VIII.3 Output Format

Using this control, the user may select the output format type for the process. The OUTPUT FORMAT section name must have been referenced by a FORMAT clause in the definition of the Process (see item V.1.3).

Format options are shown in a box with a dynamic list of the “combobox” type.

Control/Clause	Type	Comment
FORMATCAPTION	Text	Text to be shown along with the format dynamic box
FORMATN	Integer	Number of formats (items) in the list to be shown in the box. There must be the clauses FORMAT1, FORMAT2, …, up to FORMATn, where *n* is the number in FORMATN.
FORMAT$_i$	RWord & Text	Reserved word, followed by a text describing the format (to be displayed in the filter box). The reserved word may be a combination of reserved words, which will mean that more than an output format is desired.

Format options include:

HTML	for Tables
GRAPH	for Graphs
MAP	for Maps
SPC	to show the Redatam program

Or any combination of the above for more than one format. For example:

HTMLMAP	for Table & Map
HTMLGRAPH	for Table & Graph[15]

Example

Contents	**Comment**
`[FORMAT_2]`	Section ID
`FORMATCAPTION=Output Format:`	Eight potential output format types for selection
`FORMATN=8`	
`FORMAT1=HTML Table`	
`FORMAT2=GRAPH Graph`	
`FORMAT3=MAP Map`	
`FORMAT4=SPC R+ Program`	
`FORMAT5=GRAPHHTML Table and Graph`	
`FORMAT6=MAPHTML  Table and Map`	
`FORMAT7=GRAPHMAP  Map and Graph`	
`FORMAT8=MAPGRAPHHTML Table, Graph and Map`	

The screen with a format dynamic box “open” can be seen in Figure 44.

[15] The writing order for combinations is not relevant; it might as well be GRAPHMAP or MAPGRAPH

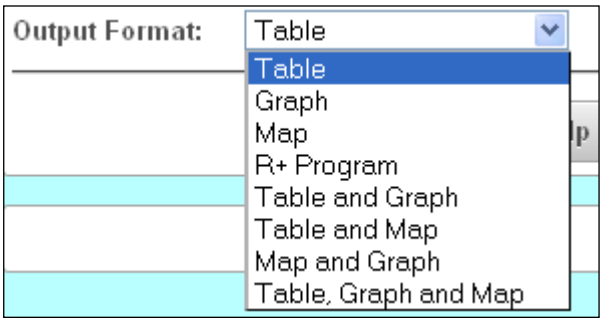


Figure 44. Format type control

VIII.4 Geographic Selection

This control is intended to define the available geographic selections. The section name must have been stated in a SELSET clause in [STRUCTURE] (see item V.1.3).

Control/Clause	Type	Comment
CAPTION	Text	Label of the selection box
SELECTIONS	Integer	Number of selection files that will be in the selection list. Must include as many SELNAME and SELCAPTION clauses as the number is in SELECTIONS.
SELNAME$_i$	File Name	Either the name of a geographic selection file, or the reserved word ALL meaning that the whole database will be processed. The file name may have directory markers.
SELCAPTION$_i$	Text	Text to be displayed in the Selection box to specify the appropriate SELNAME file.

Example

Contents	**Comment**
`[SELSET1]`	Section ID
`CAPTION=Geographic Area:`	A list containing 4 geographic selections
`SELECTIONS=4`	
`SELNAME1=ALL`	
`SELCAPTION1=The entire database`	
`SELNAME2=%INLPATH\SantaMaria.sel`	
`SELCAPTION2=Santa Maria`	
`SELNAME3=%INLPATH\Santiago.sel`	
`SELCAPTION3=Santiago`	
`SELNAME4=%INLPATH\Bolivar.sel`	

VIII.5 Tally

This control is used to multiply the results from the process with the value of a variable, such as the TALLY OPTION in REDATAM . TALLY may be used in 2 ways: either using a fixed setting, i.e., results will ALWAYS be increased by the variable. Or it can be set as a variable list in a box, so that the user may select which of those variables will be used as increasers.

Note

Just as in the TALLY OPTION in REDATAM , here also a numeric value instead of a variable is allowed for increments.

VIII.5.a Fixed Tally

In this case, the information to be entered is the name of the variable, and the single clause is TALLYFIXED. The increment variable will not be displayed in the process.

Control/Clause	Type	Comment
TALLYFIXED	Variable Name	Full name (entity.variable) of the variable to be used for increments.

Example

Using always a fixed multiplier

Contents	Comment
`TALLYFIXED=PERSONA.HIJOS`	Results will always be increased by the value of this variable. This control will not be shown on screen.

Note

This clause may be directly placed in the main execution processes (Frequency, Cruz, etc.). There is no need to state

```
TALLY=TALCLAUSULA
```

in the main process, and then to create a section

```
[TALCLAUSULA]
TALLYFIXED=PERSONA.HIJOS
```

VIII.5.b Varying Tally

In this case, the information that must be entered is the list of variables that may be used. There may be several controls of the Tally type.

Control/Clause	Type	Comment
CAPTION	Text	Text to be shown with the variable box
TALLYN	Integer	Number of variables (items) in the list to be shown in the box. There must be the clauses TALLY1, TALLY2,…, up to TALLYn, where *n* is the number in TALLYN.
TALLY$_i$	Variable Name	Full name (entity.variable) of the variable to be used as increment. Optionally, text between brackets may be used to signify that an increment won't be used in calculation.

Example

Variable list. In this case you may choose between counting the Total Children (HIJOST), Surviving Children (HIJOSS), or Mothers (case count only).

Contents	Comment
`[SECTALLY]`	Section ID
`TALLYCAPTION=Count Children / Mothers`	Increment box caption
`TALLYN=3`	3 increment options
`TALLY1=PERSONA.HIJOST`	
`TALLY2=PERSONA.HIJOSS`	
`TALLY3=(Madres)`	As a rule, if one of the items in the list begins with brackets, then the system assumes that that Control won't be used. That is to say that if the user chooses this option, then the results will have no increments. In such case only the mothers will be counted.

Note

The TALLYFIXED clause overrides other clauses.

VIII.6 Universe & AltFilter

UNIVERSE and ALTFILTER are two clauses (not sections) that are used to select relevant cases in a process. For example, 15 year-old and above women for a process on fertility, or persons aged 65 years and above for an indicator on aging. These clauses are NOT shown on screen to the user, since they are internal filters for the process.

The difference between both clauses is that UNIVERSE is embedded in the program using the RUNDEF command and increases the efficiency in execution. However, UNIVERSE would not allow the use of derived variables in its expression, which is solved using ALTFILTER.

Control/Clause	Type	Comment
UNIVERSE	Text	Filter expression using only database variables
ALTFILTER	Text	Filter expression using derived variables calculated in the DEFINEs.

Example 1

65 year-old and above

```
UNIVERSE=(PERSON.EDQUINQ > 13)
```

Example 2

Households where 65 year-old and above persons reside

```
ALTFILTER=VIVIEN.TOTELD > 0
```

Comments on UNIVERSE, ALTFILTER, & FILTER

1. All these are intended to filter cases and may be used combined in the same process, but they operate as 'UNIVERSE AND ALTFILTER AND FILTER' (there is no way to combine them using OR).
2. UNIVERSE and ALTFILTER are not shown to the user. If they exist in a process then they are always applied upon execution and don't have any explanatory texts. On the other hand, FILTERs may be user selected and have explanatory texts.
3. UNIVERSE and ALTFILTER are single expression clauses, while FILTER is a box with a list of expressions.

VIII.7 Weight

This control is used to weigh the results from the process in the value of a variable, such as the WEIGHT OPTION in REDATAM . WEIGHT may be used in two ways: either using a fixed setting, i.e., results will ALWAYS be weighted by the variable. Or it can be set as a variable list in a box, so that the user may select which of those variables will be used as weight.

Note

Just as in the WEIGHT OPTION from REDATAM , here also a numeric value instead of a variable is allowed for weight.

VIII.7.a Fixed Weight

In this case, the information to be entered is the name of the variable, and the single clause is WEIGHTFIXED. The weight variable will not display in the process.

Control/Clause	Type	Comment
WEIGHTFIXED	Variable Name	Full name (entity.variable) of the variable to be used for weight.

Example

Always fixed weight

Contents	Comment
`WEIGHTFIXED=DISTRICT.WEIGHT`	Results will always be weighed by the value of this variable. This control will not be shown on screen.

Note

This clause may be directly placed in the main execution processes (Frequency, Cruz, etc.). There is no need to state

```
WEIGHT=WEICLAUSULA
```

in the main process, and then to create a section

```
[WEICLAUSULA]
WEIGHTFIXED=DISTRICT.WEIGHT
```

VIII.7.b Varying Weight

In this case, the information that must be entered is the list of variables that may be used. There may be several controls of the Weight type.

Control/Clause	Type	Comment
CAPTION	Text	Text to be shown with the variable box
WEIGHTN	Integer	Number of variables (items) in the list to be shown in the box. There must be the clauses WEIGHT1, WEIGHT2,…, up to WEIGHTn where *n* is the number in WEIGHTN.
WEIGHT$_i$	Variable Name	Full name (entity.variable) of the variable to be used as weight. Optionally, text between brackets may be used to signify that a weight won't be used in calculation.

Example

Variable list.

Contents	Comment
[WEIGHT1]	Section Name
WEIGHTCAPTION=Expansion Factor	Weight box caption
WEIGHTN=3	3 weight options
WEIGHT1=DOMICIL.PESO	Weight variables
WEIGHT2=PERSONA.PESO	
WEIGHT3=(none)	As a rule, if one of the items in the list begins with brackets, then the system assumes that that Control won't be used. That is to say that if the user chooses that option, then the results will have no weight

Note

The WEIGHTFIXED clause overrides other clauses.

IX. Special Sections

In a Guest file, these sections fulfill quite specific purposes and don't follow a common form. Each of them has its own pattern and function. The name of each is set through programming, except the SELSET section.

IX.1 Defines

The names of these sections MUST be DEFINE1, DEFINE2,…, DEFINEn, in a sequence WITHOUT OMISSIONS, and there must be as many sections as is the value of the DEFINEs clause in [STRUCTURE] (see item V.1.3). These sections are intended to define the derived variables that will be required by REDATAM during the execution of processes.

The clauses to be "written" in a DEFINE are the PVL commands from the REDATAM intermediate language. Refer to Annex V for the construction of these DEFINEs.

Example

Contents	Comment
`[DEFINE12]`	Section Name
`NAME=DISTC`	Name of the variable to be created
`ENTITY=DISTRITO`	Entity for which the variable will be created
`TYPE=INTEGER`	Type of variable (the default is INTEGER)
`AS=EXPRESSION`	Variable defined as an expression
`EXPRESSION=VAL( CMPCODE ( DISTRITO ) )`	Numeric value of the full code for the DISTRICT entity

However, if there is dependence between derived variables, two additional clauses may be required.

Control/Clause	Type	Comment
DEPENDENCIES	Integer	Number of derived variables from which this variable depends. This number must match the number of DEPENDENCYi clauses included.
DEPENDENCY$_i$	Variable Name	Full name (entity.variable) of the variable from which this variable depends.

Example

Contents	Comment
`[DEFINE14]`	Section ID
`NAME=EDADNUMER`	Name of the variable to be created

`ENTITY=DISTRITO`	Entity for which the variable will be created
`AS=COUNT`	Variable defined as count
`EXPRESSION=PERSON`	Person count
`DEPENDENCIES=1`	Has 1 dependent
`DEPENDENCY1=PERSON.EDADGRA`	Name of the variable from which this new variable depends

IX.2 Footnotes

There must be as many footnote sections as the value in the FOOTNOTES clause in [STRUCTURE], and the names of these sections will ALWAYS be FOOTNOTE1, FOOTNOTE2, etc.

Control/Clause	Type	Comment
LINES	Integer	Number of footnote lines in the section. There must be as many clauses LINE1, LINE2, etc. as is the value in LINES
LINE$_i$	Text	Text with the contents of line *i* in the footnote.

Example

Contents	Comment
`[FOOTNOTE1]`	Section Name
`LINES=2`	
`LINE1=Processed with Redatam`	
`LINE2=ECLAC/CELADE 2003-2006`	

IX.3 Glossary

The section name MUST be GLOSSARY. This section is intended to define the texts that are common to the Processes in a Guest file. In general they are texts that will not change between one application and another. Just like with the PREFERENCES or GRAPH sections, these sections are placed in a file of the type #include.

Control/Clause	Type	Comment
BUTTONFILTER	Text	Filter button text
BUTTONFREQUENCY	Text	
BUTTONHELP	Text	Help button text
BUTTONSELECTION	Text	Edit Selection button text
BUTTONSUBMIT	Text	Execute button text

PROCESSTITLE	Text	Process text
TABLETITLECAPTION	Text	Table title text
WEBMASTER	Text	Text to be shown in footers from results.
COPYRIGHT	Text	Text to be shown in footers from results.

Example

Figure 45 shows a process screen with Glossary items and the webmaster and copyright footnotes. The contents of the webmaster line appear centered on screen because it was so designed in the htm file.

Contents	**Comment**
`[GLOSSARY]`	Section Name
`BUTTONFILTER=Build Filter`	Filter button text
`BUTTONHELP=Help`	Help button text
`BUTTONSELECTION=Edit`	Edit Geographic Selection button text
`BUTTONSUBMIT=Execute`	Submit (Execute) button text
`PROCESSTITLE=Parameter Definition`	Process text
`TABLETITLECAPTION=Table Title`	Text in the definition label for the Table Title
`WEBMASTER=redatam@cepal.org`	Contents of the webmaster line in output
`COPYRIGHT=Copyright © 2002-2006 Latin American and Caribbean Demographic Center (CELADE), ECLAC, Santiago, Chile`	Contents of the Copyright line in output

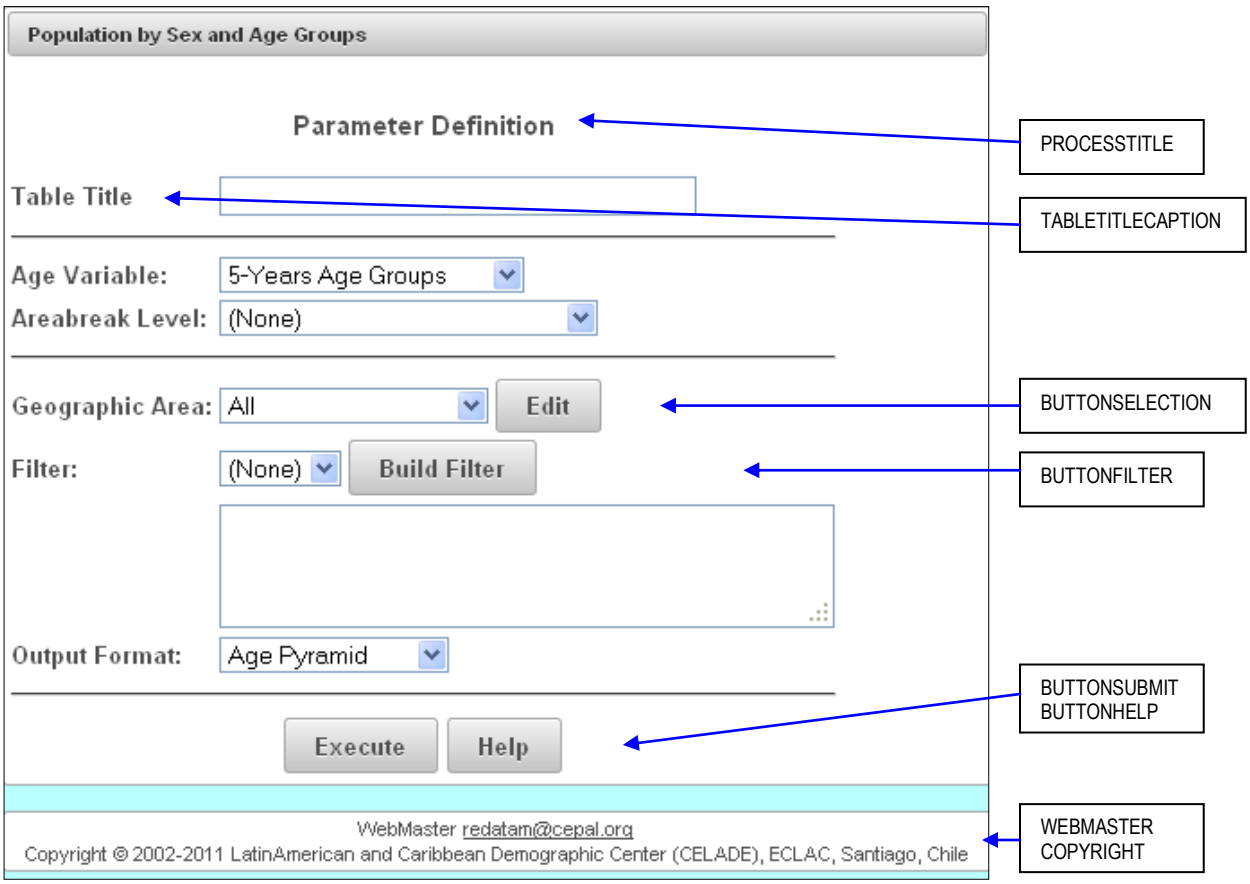


Figure 45. Input screen for a CRUZ type process

IX.4 Graph

The section name MUST be GRAPH. This section contains the general clauses to be applied to an output in the GRAPH format (see item VIII.3).

Control/Clause	Type	Comment
DIMENSION	RWord	2D and 3D for 2 or 3 dimensions, respectively, or else DEFAULT (by default, it takes on the preset value for the type of results to be displayed)
SORT	RWord	Order of data in the output. May be ASCENDING, DESCENDING, or NONE (default)
COLORSCHEME	RWord RAMP	Color scheme. It may be RAMP for orderly color, starting with a COLORMIN value and finishing with a COLORMAX value; or it may be UNIQUE for an unique color, or DEFAULT (by default, it takes on the preset value for the type of results to be displayed)
COLORMIN	RGB or RWord	Initial color in the color ramp.
COLORMAX	RGB or RWord	Final color in the color ramp

COLORSINGLE	RGB or RWord	Whether the graph will have only one color
COLORLEFT	RGB or RWord	Color in the left section of a graph in pyramid format (an age pyramid is the most common case)
COLORRIGHT	RGB or RWord	Color in the right section of a graph in pyramid format

Example

Contents	**Comment**
`[GRAPH]`	Section Name
`DIMENSION=3D`	
`SORT=ASCENDING`	
`COLORSCHEME=RAMP`	
`COLORMIN=BLUE`	
`COLORMAX=YELLOW`	
`COLORSINGLE=GREEN`	
`COLORLEFT=BLUE`	
`COLORRIGHT=RED`	

IX.5 Panels

This is a group of similar sections, intended to define the environment for each of the three panels (Header, Index, and Input). The names of these sections MUST consist of the prefix PANEL followed by the name of the corresponding panel (e.g. PANELHEADER).

IX.5.a PanelHeader

Defines the environment in the top section of the page.

Control/Clause	Type	Comment
HEIGHT	Integer	Height of the panel in pixels
COLOR	RGB or RWord	Background color
PICTURES	Integer	Determines the number of pictures in the panel. May contain up to 2 pictures, which will have their own control (PICTURE1 and PICTURE2), defined as follows.

Control: PICTURE

Control/Clause	Type	Comment
PICTURETOP$_i$	Integer	Distance (in pixels) from the picture to the top of

		the panel
PICTURELEFT$_i$	Integer	Distance (in pixels) from the picture to the left margin of the panel
PICTUREHEIGHT$_i$	Integer	Height (in pixels) of the picture
PICTUREWIDTH$_i$	Integer	Width (in pixels) of the picture
PICTUREFILE$_i$	Filename	Contains the image in the picture. May be a .gif, .jpg, or .bmp file, and must be in the directory pointed to by WORK

Example

Contents	**Comment**
`[PANELHEADER]`	Section Name
`HEIGHT=50`	Height of the panel
`COLOR=255.255.255`	Background color of the panel (white)
`PICTURES=2`	2 pictures in the panel
`PICTURETOP1=0`	Location, size, and file for picture 1
`PICTURELEFT1=2`	
`PICTUREHEIGHT1=50`	
`PICTUREWIDTH1=512`	
`PICTUREFILE1=%INLPATH\ESP\Titulo_ESP.bmp`	
`PICTURETOP2=0`	Location, size, and file for picture 2
`PICTURELEFT2=600`	
`PICTUREHEIGHT2=50`	
`PICTUREWIDTH2=209`	
`PICTUREFILE2=%INLPATH\ESP\Celade_ESP.BMP`	

IX.5.b PanelIndex

Defines the environment for the left section of the page.

Control/Clause	Type	Comment
WIDTH	Integer	Length of the panel in pixels
COLOR	RGB or RWord	Background color
FONTNAME	RWord	Font name. Any font name that is accepted by Windows
LINES	RWord YES/NO	

Example

Contents	Comment
[PANELINDEX]	Section Name
WIDTH=300	Width of the panel
COLOR=185.255.255	Background color of the panel (a little darker than aqua)
FONTNAME=ARIAL	Arial font
LINES=NO	

IX.5.c PanelInput

Defines the environment for the central section of the page, upon receiving the parameters from the user.

Control/Clause	Type	Comment
HEIGHT	Integer	Height of the panel in pixels
COLOR	RGB or RWord	Background color
FONTSIZE	Integer	Font size for displaying all texts in the panel.
TITLESIZE	Integer	
TITLEBOLD	RWord YES/NO	
COMBOSIZE	Integer	
COMBOSELSIZE	Integer	

Example

Contents	Comment
[PANELINPUT]	Section Name
HEIGHT=200	Width of the panel
COLOR=AQUA	Background color of the panel (0.255.255)
FONTSIZE=9	9-point font
TITLESIZE=8	8-point title (smaller than the fonts for displaying options)
TITLEBOLD=YES	
COMBOSIZE=105	
COMBOSELSIZE=109	

IX.6 Preferences

The section name MUST be PREFERENCES. This section contains the general clauses of a Guest file, in terms of the parameters that will be entered into REDATAM when the file takes control of the process execution.

Control/Clause	Type	Comment
HEADERPANEL	RWord	
COLOR	RGB o RWord	
FONTCOLOR	RGB RWord	Font color, in a RGB format or reserved word defining a known color in the system
FONTNAME	Text	Font to be used
FONTSIZE	Integer	Font size
ROOTFONTSIZE	Integer	
PRESENTATION	RWord	
PROGRESS	RWord	
LANGUAGE	Integer	Language for displaying pages, results, and system error messages. The default value is 1. Options: 1 English 2 Spanish 3 Portuguese 4 French 5 Bahasa
DECIMALS	Integer	Number of decimals in the output
GRAPH	RWord	Inclusion of graphs, options YES|NO, default is NO
MAP	RWord	Inclusion of maps, options YES|NO, default is NO
MAPLASTOUTPUT	RWord	
OMITTITLE	RWord	Omission of the title in the output, options YES|NO, default is YES
DECIMAL	Character	Format for displaying decimals. “.” (point) or “,” (comma), default is point
MILES	Character	Format for displaying thousands. “.” (point) or “,” (comma), default is comma
LOADLAST	RWord	
USEWEIGHT	RWord	
SAFETY	RWord	
WARNING	Integer	

Example

Contents	Comment
[PREFERENCES]	Section Name
HEADERPANEL=YES	Show Header Panel
COLOR=SILVERLITE	General color is light gray
FONTCOLOR=0.0.0	Color for texts is black
FONTNAME=ARIAL	Font type
FONTSIZE=8	Font size
ROOTFONTSIZE=12	Font size for the first entry in the Index
PRESENTATION=RICH	
PROGRESS=FAST	
LANGUAGE=2	Language Spanish
DECIMALS=2	2 decimal places
GRAPH=YES	Graphs and maps displayed
MAP=YES	
MAPLASTOUTPUT=YES	
OMITTITLE=YES	Omit table titles (show only values)
DECIMAL=.	Decimal point
MILES=,	Comma for the thousands place
LOADLAST=YES	
USEWEIGHT=NO	Automatic weight not used
SAFETY=YES	Show process errors
WARNING=2	

Annex I: Description of an INL file

Generally, a file of the INL type[16] has "INL" as extension and may be created using any ASCII text editor. Its contents are organized in data blocks that make up sections and clauses, very similarly to an INI file in Windows 3.1.

Each section is identified by a name in square brackets (e.g. [PREFERENCES]), and the clauses within each section are in uppercase, each in a separate line, followed by an equal sign ("=") and the value that the clause must have (e.g., MAP=YES). These values for clauses are also called parameters[17]. No two sections can have the same name, but clauses with the same name may exist in different sections.

There may not be blank spaces between the clause, the equal sign ("="), and the parameter value. If the parameter is a text string, then it may contain blank spaces (it's not required to place the text string between quotes). Examples of clauses:

/valid
```
NODETYPE=STRUCTURE
CAPTION=Basic Features
```

/invalid
```
NODETYPE   =STRUCTURE
NODETYPE=   STRUCTURE
CAPTION="Basic Features"
```

Blank lines may be used to provide increased clarity and to separate sections and clauses.

Unrecognized clauses are ignored, and thus may be used as comments for documentation. However, it's suggested that a comment line would start with a special character (e.g., "/"), in order to differentiate them from the clauses themselves.

The "#include" command may be used to call those definitions that are contained in other INL files. This is useful to better organize the commands in an INL file. The "#include" command is NON-recursive, i.e., those files being called by the "#include" command CANNOT in turn have other "#include" commands. Example:

```
#include WORK\MAPS_ESP.Mxp
```

The order of sections within the file is not significant. The order of clauses within a section is also not significant. A standard block in an INL file would look like:

```
/comment
(blank line)
[SECCION1]
CLAUSULA1=PARAMETRO1
CLAUSULA2=PARAMETRO2
……
```

[16] Comes from INdicator Language, for constructing indicators.
[17] An "entry" is a line containing the "clause=parameter" set

```
[SECCION2]
CLAUSULA1=PARAMETRO1
```

Annex II: Using Directory Markers

In order to avoid potential problems caused by bad addressing to the files used (perhaps because the directories were moved), the directory marker of the Guest file through the %INLPATH parameter may be used. At runtime, this parameter is replaced with the full ("path") step for the Guest.INL file of the application.

In order to refer to a file in the same directory as Guest.inl, you just may use %INLPATH[18], a backslash, and the file name, e.g.:

```
NAME=%INLPATH\utilities.INL
```

In this case, if Guest.inl is in the directory c:\servers\redatam\cgibin\EXAMPLE, then the clause will be:

```
NAME=c:\servers\redatam\cgibin\EXAMPLE\utilities.INL
```

To refer to a file in a directory at a lower level than Guest.inl, use the file step STARTING FROM the Guest directory, e.g.:

```
NAME=%INLPATH\BaseR\NmirEsp.dic
```

Other examples:

```
WORKPATH=%INLPATH\Site\
MXP=%INLPATH\ESP\Comunas_ESP.mxp
PICTUREFILE2=%INLPATH\ESP\Celade_ESP.BMP
```

[18] INLPATH must be uppercase

Annex III: Using Colors

Colors in REDATAM may be stated in two ways: either using their reserved name in the REDATAM system, or using their reference code in the RGB (Red, Green, and Blue) international notation system. In summary, this notation numerically represents the colors available from combining Red, Green, and Blue in a concentration scale ranging from 0 (smaller) to 255 (greater), and separating the values for each concentration by a '.' (period). For example, black is 0.0.0 (total absence of color), while white is 255.255.255.

Some colors have been preset in REDATAM, such as AQUA (0.255.255), Dark Lenin (120.96.88), etc.

The following table shows some of the colors in REDATAM and their correspondences in RGB notation.

REDATAM Name	RGB	Color
BLACK	0.0.0	
AQUA	0.255.255	
BLUE	0.0.255	
YELLOW	255.255.0	
OLIVE	128.128.0	
RED	255.0.0	
SILVER	192.192.192	
WHITE	255.255.255	

The list below includes the names of the colors that are accepted by REDATAM. Some colors have English and Spanish names; both are acceptable.

AQUA	AGUA		NAVY	
BLACK	NEGRO		OLIVE	OLIVA
BLUE	AZUL		PURPLE	PURPURA
DARKGRAY			RED	ROJO
FUCHSIA	FUCCIA		SILVER	PLATA
GRAY			SILVERLITE	
GREEN	VERDE		TEAL	
LIME	LIMA		WHITE	BLANCO
MAROON			YELLOW	AMARILLO
LENIN			DARKLENIN	
CHILEINE			DARKCHILEINE	

Annex IV: Dynamic Definition of Output Entities

Sometimes, an indicator that has been calculated at several geographic levels (entities) must be built. This may imply, e.g., counting persons of 65 years and above, counting all persons, and then dividing one by another, thus obtaining the ratio of seniors over the total population at the geographic level. That is to say, 3 derived variables must be created for each of the geographic levels to be used at the output level of the indicator. For District, REDATAM's common language would look like

```
DEFINE DISTRICT.PERS65
      AS COUNT PERSON
      FOR PERSON.AGE >= 65

DEFINE DISTRICT.TOTPERS
      AS COUNT PERSON

DEFINE DISTRICT.RAZONP
      AS DISTRICT.PERS65 / DISTRICT.TOTPERS
      TYPE REAL
```

Then we would copy and paste the same commands for other levels, such as Enumeration County, etc. If they are many geographic levels involved, then it's more efficient to duplicate those DEFINEs by using a generic entity name (%outent), which then will be "resolved" upon execution of the program. REDATAM's common language would look like

```
DEFINE %outent.PERS65
      AS COUNT PERSON
      FOR PERSON.AGE >= 65

DEFINE %outent.TOTPERS
      AS COUNT PERSON

DEFINE %outent.RAZONP
      AS %outent.PERS65 / %outent.TOTPERS
      TYPE REAL
```

And the definition of the indicator would be something like

```
VARN=1
VAR1=%OUTENT.RAZONP
```

However, the Program Editor will NOT accept such syntax and will return an error. The %outent parameter is directly accepted only in the Redatam Webserver program, that translates and transforms it into the name of the appropriate variable that is selected by the user as the output entity (this is what is meant by %outent, i.e., output entity). Then, a workaround for this limitation is writing the program just like we did for DISTRICT, converting it to the PVL syntax using the Command Editor in REDATAM , and then editing the PVL by converting the word DISTRICT into %outent.

Annex V: Writing DEFINE Commands

In this version of Redatam Webserver, the DEFINEs syntax is in PVL (pivot language), which is REDATAM 's internal language. A very experienced user may be able to directly write programs in this language, but the easier way to create those DEFINEs is using the Redatam Process, i.e. writing the DEFINE command in SPC language in the Program Editor, and then pressing the "Open Intermediate Language window" button () in the command bar.

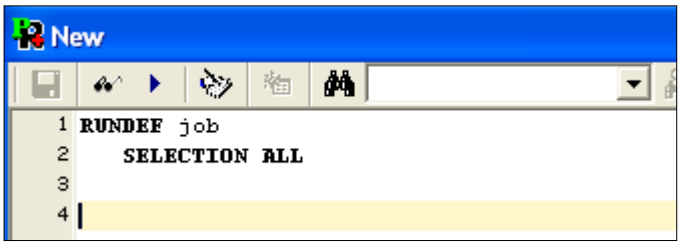


Figure 46. SPC – PVL Button

For example, let's assume that we need to create a variable to count persons younger than 5 years of age, at the Housing level. In SPC, that program would be:

```
RUNDEF program
   SELECTION ALL
DEFINE HOUSIN.NUMPERS
   AS COUNT PERSON
   TYPE INTEGER
   FOR PERSON.AGE < 5
```

Taking that program and using the button above mentioned, the Program Editor would convert that to:

```
[DEFINE1]
NAME=NUMPERS
ENTITY= HOUSIN
TYPE=INTEGER
LINE=5
AS=COUNT
FOR=PERSON.AGE < 5
EXPRESSION=PERSON
```

Now we must change the DEFINE number (in the example this is [DEFINE1]) into the sequential define number that will be placed in the Guest file (or in an #include to be called by the Guest file), such as [DEFINE7]. LINE=5 clause in the example is not required; it's used by REDATAM as an auxiliary in error messages. In our Redatam Webserver case, this line may be commented (//) or just deleted.

Special care must be used with variables that derive from derived variables, i.e. a variable that is created starting from a previously created variable. In this version of Redatam Webserver, such dependence must be reported. For example, we may take the variable

HOUSIN.NUMPERS and create another, derived from this variable, with a value of 0 and 1 if there are any persons younger than 5 years old. The program would be:

```
DEFINE VIVIEN.INDIC AS VIVIEN.NUMPERS > 0
   TYPE BOOL
```

The PVL converter would change it into:

Contents	Comment
[DEFINE2]	Section ID
NAME=INDIC	Variable, and the entity to which it belongs
ENTITY=VIVIEN	
LINE=10	Can be deleted
AS=EXPRESSION	Definition expression for the variable
EXPRESSION=VIVIEN.NUMPERS > 0	
TYPE=INTEGER	REDATAM converts internally a variable of the BOOL TYPE to an INTEGER variable with the range characteristics 0-1 and the appropriate labels
VALUELABELS=2	
VL1=0 False	
VL2=1 True	
RANGES=1	
RANGEMIN1=0	
RANGEMAX1=1	

To use it in the Guest DEFINEs, the number of the DEFINE [DEFINE2] must be changed (also, remember to increase the number of DEFINES in [STRUCTURE]). But if the variable to be used in the Guest indicators is HOUSIN.INDIC, then the Guest file must be informed that it won't suffice to create the program using that variable, since the variable depends on VIVIEN.NUMPERS. In order to show this dependence, two clauses must be added to [DEFINE2], the first to count how many dependences there are (it may depend on more than one), and the second to report the name of the dependency variable.

```
DEPENDENCIES=1
DEPENDENCY1=VIVIEN.NUMPERS
```

Then, the full section to be included in the Guest file would be:

Contents	Comment
[DEFINE8]	Section name, already changed for the sequence number in the Guest file
NAME=INDIC	
ENTITY=VIVIEN	
TYPE=INTEGER	
//LINE=10	Comment clause, may be deleted
AS=EXPRESSION	
EXPRESSION=VIVIEN.NUMPERS > 0	
VALUELABELS=2	
VL1=0 False	
VL2=1 True	
RANGES=1	
RANGEMIN1=0	
RANGEMAX1=1	
DEPENDENCIES=1	Dependency clauses
DEPENDENCY1=VIVIEN.NUMPERS	

Annex VI: Restricted Access Databases

In order to protect a database so that only authorized persons may access it, the AUTHENTICATE=YES clause must be used in the STRUCTURE section of the Guest file. In that case, before showing the main screen with the process Index, the system shows a screen as in Figure 9 in Item II.4.3, so that the user may enter his ID and password.

Also, in the directory %INLPATH must be a file called USERS.PWD with a list of the authorized users and passwords. This is a text type file in INL format, with a line (clause) for each user. There may be as many users as required, each with a clause (user id), and a password. The password is reported encrypted in the file after using the ENCRYPT.EXE utility that can be found in the CELADE website. Upon execution, the program shows the screen in Figure 47.

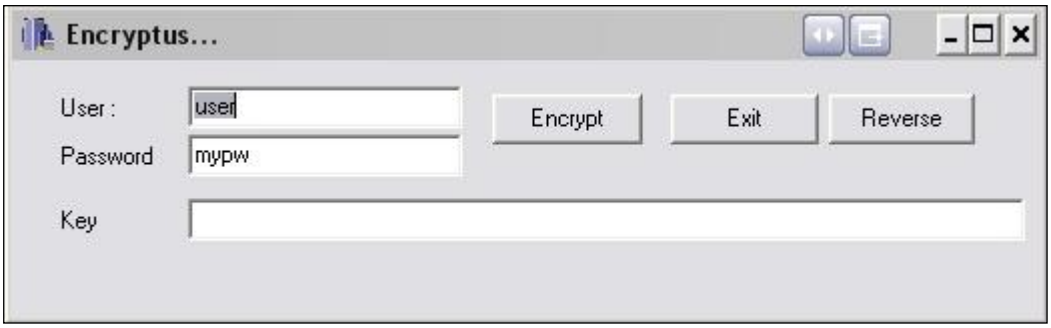


Figure 47. Encrypt Program

To generate an encrypted password, the user's name and password must be entered as usual. For example, let's assume that the user is "LUNES" and the password is "work" (passwords and users are case-sensitive, i.e., "Work" it is not the same as "WORK"). When entering these values into the program and pressing the "Encrypt" button, the program will show the encrypted code, as in Figure 49, in the USERS.PWD file.

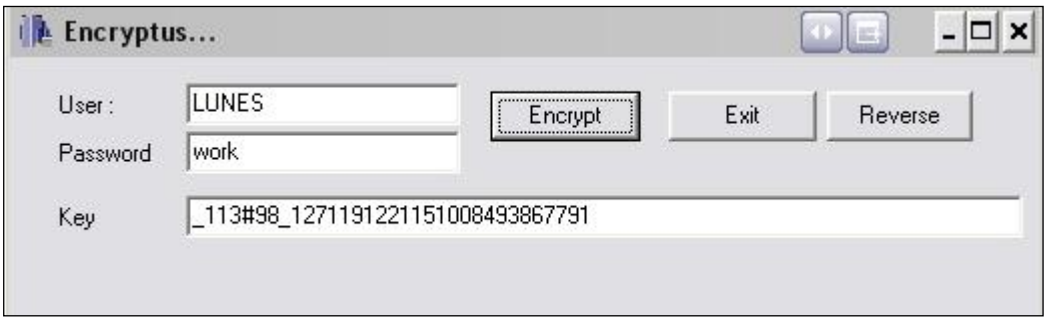


Figure 48. Encrypt with password

Copy this code (Key on screen) using [Ctrl]-C, and generate an entry in the USERS.PWD file, with the user's name (LUNES, in this example)

LUNES=_113#98_127119122115100849386779 1

Annex VII: Clause List

The following list is made up of sets of similar clauses in alphabetical order. Within each group, clauses have been listed in order to make them more understandable. For example, the group of ABK clauses comes before the group of AGEMAX clauses, but within the ABK group, the ABKN clause is listed before ABKi.

Clause	Comment
#include	Call a file so that its contents are included in the **Guest.INL**
ABK	References the name of a section having **AREABREAK** clauses
ABKN	Number of **AREABREAK** items (entities) that will be defined below
ABKi	Name of an entity in the list of potential **AREABREAKs** (i ranges from 1 to the number defined in ABKN)
ABKCAPTION	Caption text in the **AREABREAK** box
ABKFIXED	Name of an entity to serve as **AREABREAK**, that is non-user selectable and that will not display on screen
ABKLAST	Number of the item to be initially shown in the **AREABREAK** box
AGEMAXN	Number of items to define the maximum age in the **DEPRATIO** node
AGEMAXi	One of the maximum age limits (i ranges from 1 to the number defined in AGEMAXN) in **DEPRATIO**
AGEMAXCAPTION	Caption text for the maximum ages box in **DEPRATIO**
AGEMINN	Number of items to define the minimum age in the **DEPRATIO** node
AGEMINi	One of the minimum age limits (i ranges from 1 to the number defined in AGEMINN) in **DEPRATIO**
AGEMINCAPTION	Caption text for the minimum ages box in **DEPRATIO**
AGEVAR	Age variable in **DEPRATIO**
ALTFILTER	Expression for a process filter when derived variables are used
AVG	References the name of a section having **AVERAGE** clauses
AVGN	Number of **AVERAGE** items (variables) that will be defined below
AVGi	Name of a variable in the list of potential **AVERAGE**s (i ranges from 1 to the number defined in AVGN)
AVGCAPTION	Caption text for the **AVERAGE** box
AVGFIXED	Name of an entity to serve as **AVERAGE**, that is non-user selectable and that will not display on screen
AVGLAST	Number of the item to be initially shown in the **AVERAGE** box
BUTTONFILTER	Text for the Build Filter button
BUTTONFREQUENCY	Text for the Execute Frequency button

BUTTONHELP	Text for the Help button
BUTTONSELECTION	Text for the Edit Geographic Selection button
BUTTONSUBMIT	Text for the Execute button
CAPTION	Caption text for an item, either a control or a node
CNT	References the name of a section that has clauses for item counting (**COUNT**)
CNTN	Number of **COUNT** items (entities) that will be defined below
CNTi	Name of an entity in the list of potential **COUNT**s (i ranges from 1 to the number defined in CNTN)
CNTCAPTION	Caption text for the **COUNT** box
CNTFIXED	Name of an entity to serve as **COUNT**, that is non-user selectable and that will not display on screen
CNTLAST	Number of the item to be initially shown in the **COUNT** box
COL	References the name of a section that has clauses for the **COLUMN** of a process (second dimension)
COLN	Number of items (variables) in the **COLUMN** that will be defined below
COLi	Name of a variable in the list of potential **COLUMNs** (i ranges from 1 to the number defined in COLN)
COLCAPTION	Caption text for the **COLUMN**s dimension box of a process
COLFIXED	Name of a variable to serve as **COLUMN**, that is non-user selectable and that will not display on screen
COLLAST	Number of the item to be initially shown in the **COLUMN** box
COLOR	RGB or color ID for an item
COLORLEFT	Color for the left side of a pyramid
COLORMAX	Maximum color in a color ramp
COLORMIN	Minimum color in a color ramp
COLORRIGHT	Color for the right side of a pyramid
COLORSCHEME	Color scheme ID
COLORSINGLE	Single color, if a single color scheme is used
COPYRIGHT	Text for the copyright section
CTL	References the name of a section that has the clauses for the **CONTROL** of a process (3rd dimension)
CTLN	Number of **CONTROL** items (variables) that will be defined below
CTLi	Name of a variable in the list of potential **CONTROL**s (i ranges from 1 to the number defined in CTLN)
CTLCAPTION	Caption text for the **CONTROL**s dimension box of a process
CTLFIXED	Name of a variable to serve as **CONTROL**, that is non-user selectable and that will not display on screen
CTLLAST	Number of the item to be initially shown in the **CONTROL** box

DATAPATH	Database dictionary path
DATASETLABEL	Database label
DATASETS	Number of available databases
DECIMAL	Character to be used as decimal positioner
DECIMALS	Number of decimal positions
DEFAULTFOOTNOTE	Number of the footnote to be shown by default
DEFAULTSELSET	Set of geographic selections to be used by default
DEFAULTVALUE	Text for display subtitling of a DICTIONARY process
DEFINES	Number of derived variable definitions (**DEFINE**s)
DENUM	References the name of a section that has the clauses for the **DENOMINATOR** of a process
DENUMN	Number of **DENOMINATOR** items (variables) that will be defined below
DENUMi	Name of a variable in the list of potential **DENOMINATOR**s (i ranges from 1 to the number defined in DENUMN)
DENUMi.VLDENUMj	YES to preselect the category j of variable i in a process with a **DENOMINATOR** (j ranges from 1 to the number of categories of the variable i)
DENUMCAPTION	Caption text for the **DENOMINATOR** box of a process
DENUMLAST	Number of the item to be initially shown in the **DENOMINATOR** box
DEPENDENCIES	Number of dependencies (other derived variables) in the definition of a derived variable
DEPENDENCYi	Name of a derived variable (i ranges from 1 to the number defined in DEPENDENCIES)
DIMENSION	Definition of the number of dimensions in graphs
FEMALECODE	FEMALE category for the SEX variable in a **SEXRATIO**
FILENAME	File name for nodes having external references (HTML, etc.)
FILTER	References the name of a section that has the clauses for the **FILTER** of a process
FILTERN	Number of items (expressions) of the **FILTER** that will be defined below
FILTERi	Expression in the list of potential **FILTER**s (i ranges from 1 to the number defined in FILTERN)
FILTERCAPTION	Caption text for the **FILTER** box of a process
FILTERFIXED	Expression to serve as a **FILTER**, that is non-user selectable and that will not display on screen
FILTERLAST	Number of the item to be initially shown in the **FILTER** box
FIRSTOUTPUT	Display sequence for output types
FMTN	Number of items (output **FORMAT** types) that will be defined below
FMTi	Format type in the list of potential **FORMAT**s (i ranges from 1 to the number defined in FMTN)
FMTCAPTION	Caption text for the **FORMAT** box of a process
FMTLAST	Number of the item to be initially shown in the **FORMAT** box

FONT	Specification (BOLD, ITALIC) for the font of an item
FONTCOLOR	Font RGB or color ID for an item
FONTNAME	Name of the font for an item
FONTSIZE	Size of the font for an item
FOOTNOTES	Number of footnotes
FORMAT	References the name of a section that has the clauses for the output **FORMAT** of a process
FORMATN	Number of items (output **FORMAT** types) that will be defined below
FORMATi	Format type in the list of potential **FORMAT**s (i ranges from 1 to the number defined in FORMATN)
FORMATCAPTION	Caption text for the **FORMAT** box of a process
FORMATFIXED	**FORMAT** type that is non-user selectable and that will not display on screen
FORMATLAST	Number of the item to be initially shown in the **FORMAT** box
GRAPH	YES to allow output in graph format
GRAPHCROS	Graph type for a crossing
GRAPHFREQ	Graph type for a frequency
GRD	References the name of a section that has the clauses for the **GRID** of a process (5th dimension)
GRDN	Number of **GRID** items (variables) that will be defined below
GRDi	Name of a variable in the potential **GRID**s list (i ranges from 1 to the number defined in GRDN)
GRDCAPTION	Caption text for the **GRID** dimension box of a process
GRDFIXED	Name of a variable to serve as **GRID**, that is non-user selectable and that will not display on screen
GRDLAST	Number of the item to be initially shown in the **GRID** box
IMAGE	Image type for a map
INDICTYPE	Indicator type, AGEBYSEX, DEPRATIO, or SEXRATIO
LANGUAGE	Working language
LASTDATABASE	Working database
LASTSELECTION	Default Geographic Selection section
LOADLAST	YES to load the last instance executed of the GUEST file
MALECODE	MALE category for the SEX variable in a **SEXRATIO**
MAP	YES to allow output in map format
MAPS	Number of map sets (MXP)
MAPi	Name of a Map section (i ranges from 1 to the number defined in MAPS)
MAPFIELD	Field to be shown in the map
MAPLASTOUTPUT	YES to show the last map
MAPSECTION	MAP ID name in the MXP file
MAXIMUM	YES to show the maximum value for the variable
MAXLABELSIZE	Maximum size for variable boxes

MILES	Character to be used as positioner for thousands
MINIMUM	YES to show the variable minimum value
MODE	Dictionary display type
MXP	Map composition file in MXP format
NAME	Dictionary file
NODES	Number of nodes in the Index
NODEi	Name of a process section (i ranges from 1 to the number defined in NODES)
NODESTYLE	Name of a style section for the process
NODETYPE	Process node type
NUM	References the name of a section that has the clauses for the **NUMERATOR** of a process
NUMN	Number of **NUMERATOR** items (variables) that will be defined below
NUMi	Name of a variable in the potential **NUMERATOR** list (i ranges from 1 to the number defined in NUMN)
NUMi.VLNUMj	YES to preselect the category j of variable i in a process with a **NUMERATOR** (j ranges from 1 to the number of categories of the variable i)
NUMCAPTION	Caption text for the **NUMERATOR** box of a process
NUMLAST	Number of the item to be initially shown in the **NUMERATOR** box
OMITCASES	YES for not showing the case counter
OMITFREQPC	YES for not showing the case frequency counter
OMITMISSING	YES for not showing missing values
OMITNOTAPPLICABLE	YES for not showing not applicable values
OMITTITLE	YES for not showing the title lines of tables
OMITTOTAL	YES for not showing the aggregate total of the table
OMITTOTALCOLUMN	YES for not showing the column total
OMITTOTALROW	YES for not showing the row total
OPT	Name of a section with the options to display percentages
OPTTOT	Text for the Total option
OPTSEL	Text for the Selected Total option
OPTPC	Text for the Percentage option
OPTPCR	Text for the Relative Percentage option
OPTSELSEL	YES to preselect the Selected Total option
OPTTOTSEL	YES to preselect the Total option
OPTPCSEL	YES to preselect the Percentage option
OPTPCRSEL	YES to preselect the Relative Percentage option
OPTNAME	Text for the option of including area names
OPTNAMESEL	YES to preselect the option of including area names
OUT	Section name to define the geographic output **LEVEL**
OUTN	Number of items (entities) in the geographic output

	LEVEL that will be defined below
OUTi	Name of a variable in the list of geographic output **LEVEL**s (i ranges from 1 to the number defined in OUTN)
OUTCAPTION	Geographic output **LEVEL** box caption
OUTFIXED	Entity to serve as a geographic output **LEVEL** that is non-user selectable and that will not display on screen
OUTLAST	Number of the item to be shown initially in the geographic output **LEVEL** box
OUTPUTN	Number of items (entities) in the geographic output **LEVEL** that will be defined below
OUTPUTi	Name of an entity in the list of potential geographic output **LEVEL**s (i ranges from 1 to the number defined in OUTPUTN)
OUTPUTCAPTION	Caption text for the geographic output **LEVEL** box of a process
PAN	References the name of a section that has the clauses for the **PANEL** of a process (4th dimension)
PANN	Number of **PANEL** items (variables) that will be defined below
PANi	Name of a variable in the potential **PANEL** list (i ranges from 1 to the number defined in PANN)
PANCAPTION	Caption text for the **PANEL**s dimension box of a process
PANFIXED	Name of a variable to serve as **PANEL**, that is non-user selectable and that will not display on screen
PANLAST	Number of the item to be initially shown in the **PANEL** box
PICTUREFILEi	Name of a picture file (i ranges from 1 to PICTURES) for the panel
PORTALBACKGROUNDHEADERIMAGE	Name of a file for the background image of the HEADER panel
PORTALBACKGROUNDINDEXIMAGE	Name of a file for the background image of the INDEX panel
PORTALBACKGROUNDINPUTIMAGE	Name of a file for the background image of the INPUT panel
PORTALBACKGROUNDOUTPUTIMAGE	Name of a file for the background image of the OUTPUT panel
PORTALTITLE	Text for the portal title
PORTALSUBTITLE	Text for the portal subtitle
PROCESSTITLE	Text for the process title
QTS	References the name of a section that has the clauses for the **QTS** process
QTSN	Number of items (variables) for **QTS**
QTSi	Name of a variable in the list of potential **QTS**s (i ranges from 1 to the number defined in QTSN)
QTSi.VLj	YES to preselect the category j of variable i in a process with a **QTS** process (j ranges from 1 to the number of categories of the variable i)
QTSCAPTION	Caption text for the **QTS** box
QTSLAST	Number of the item to be initially shown in the **QTS** box

ROW	References the name of a section that has clauses for the **ROW** of a process (second dimension)
ROWN	Number of items (variables) in the **ROW** that will be defined below
ROWi	Name of a variable in the list of potential **ROW**s (i ranges from 1 to the number defined in ROWN)
ROWCAPTION	Caption text for the **ROW**s dimension box of a process
ROWFIXED	Name of a variable to serve as **ROW**, that is non-user selectable and that will not display on screen
ROWLAST	Number of the item to be initially shown in the **ROW** box
SELECTION	References the name of a section that has the clauses for the control of **GEOGRAPHIC SELECTION**
SELN	Number of **GEOGRAPHIC SELECTION** items that will be defined below
SELi	Text for the geographic selection to display in the **GEOGRAPHIC SELECTION** box
SELNAMEi	File name type (SEL or SLW) for the **GEOGRAPHIC SELECTION**s (i ranges from 1 to the number defined in SELN)
SELCAPTION	Caption text for the **GEOGRAPHIC SELECTION** box
SELFIXED	Name of the **GEOGRAPHIC SELECTION** that will be used in the process, that is non-user selectable and that will not display on screen
SELSETS	Number of **GEOGRAPHIC SELECTION** sets
SELSETi	References the name of a section that has clauses for each of the **GEOGRAPHIC SELECTION** sets (i ranges from 1 to the number defined in SELSETS)
SEXVAR	Name of the SEX variable in **SEXRATIO**
SHOWSEL	NO for not showing the **GEOGRAPHIC SELECTION** box (default is YES)
TABLETITLE	Text to be used as title for the process output table
TABLETITLECAPTION	Caption text for the table title box
TAL	References the name of a section that has the clauses for the **TABULATOR** item of a process
TALN	Number of **TABULATOR** items (variables) that will be defined below
TALi	Name of a variable of the **TABULATOR** item (i ranges from 1 to the number defined in TALN)
TALCAPTION	Caption text for the **TABULATOR** box
TALFIXED	Name of a variable to serve as **TABULATOR**, that is non-user selectable and that will not display on screen
TALLAST	Number of the item to be shown initially in the **TABULATOR** box
TEMPLATESPC	Name of a program in SPC format for the **INPUTSPC** process
TITLE	Text to be used as title of the process on screen
TITLEBOLD	YES to output the title in bold
TOTAL	YES to output the **AREALIST** total

UNIVERSE	References the name of a section that has the clauses for the **UNIVERSE** control
UNIVERSEN	Number of items for the **UNIVERSE** that will be defined below
UNIVERSEi	Expression of one of the potential **UNIVERSE**s (i ranges from 1 to the number reported in UNIVERSEN)
UNIVERSEFIXED	Expression to serve as a **UNIVERSE**, that is non-user selectable and that will not display on screen
UNIVERSECAPTION	Caption text for the **UNIVERSE** box
USEWEIGHT	YES/NO to use automatic weight or not
VARN	Number of items (entities or variables) for **AREALIST** or **COUNT** processes that will be defined below
VARi	Name of an entity or variable for the **VAR**s box (i ranges from 1 to the value reported in VARN)
VARSELi	YES to preselect the entity or variable i in the **VAR** list
VARCAPTION	Caption text for the **VAR** box
VARFIXED	Name of an entity or variable for **AREALIST** or **COUNT**, that is non-user selectable and that will not display on screen
WARNING	Number of the warning level for potential execution errors
WEBMASTER	Text to display as the site manager's address
WEIGHT	References the name of a section that has the **WEIGHT** clauses, or the name of a variable that will be used as fixed **WEIGHT**, that is non-user selectable and that will not display on screen
WEIGHTN	Number of items in the **WEIGHT** box, that will be defined below
WEIGHTi	Name of a variable of potential **WEIGHT**s (i ranges from 1 to the value reported in WEIGHTN)
WIDTH	Width of the item (panel)
WORKPATH	Name of the working directory for picture files, selections
WORKSPACE	Name of the database dictionary file