



# How To Run The SPSM

This guide describes how to run the Social Policy Simulation Model (SPSM) using Visual SPSM and Classic SPSM.



Statistics  
Canada      Statistique  
Canada

Canada

## Table of Contents

Introduction.....	2
Running SPSM using Visual SPSM .....	2
Introduction.....	2
Creating, Opening, and Saving Simulations .....	2
Creating a new simulation .....	2
Opening an existing simulation .....	4
Saving simulations .....	4
File types.....	5
Control parameters which are changed when saving a simulation .....	5
Backup of output files .....	6
Changing simulation settings.....	6
Editing Parameters .....	6
Copying and pasting parameter values .....	10
Undo / restore parameter values .....	10
Hidden Parameters .....	10
Executing Simulations and Viewing the Log Window.....	11
Contents of Log Window .....	11
Viewing Tables and Distributions .....	12
Viewing Microdata .....	13
Reading and Saving Parameter Files .....	13
Configuring your machine to write include files using Notepad.....	15
Window States .....	15
Visual SPSM Options .....	16
Function Keys .....	18
Help.....	18
Using Tools.....	18
Running SPSM using Classic SPSM .....	18
MS-DOS overview.....	19
SPSD/M Files and Directories .....	20
SPSD Environment Variables.....	23
SPSM Control .....	24
SPSM Dialogue Structure .....	24
Parameter Editing Facility .....	26
Changing Parameter Values Interactively .....	26
Changing Parameter Values Using Include Files .....	27
Configuring your machine to write include files using Notepad.....	27
SPSM Batch Facility.....	27
Command Line Method .....	27
Control File Method.....	28
Changing the language in Classic SPSM.....	28
Running SPSM using other methods .....	29

## Introduction

This guide will describe the operation of the Social Policy Simulation Model. There are two ways of running the model, using the Visual SPSM interface and using the Classic SPSM interface. Both methods are described.

This guide assumes that you are familiar with the content of the [SPSD/M Introduction and Overview](#) guide. If you are not, please read that guide first.

## Running SPSM using Visual SPSM

### INTRODUCTION

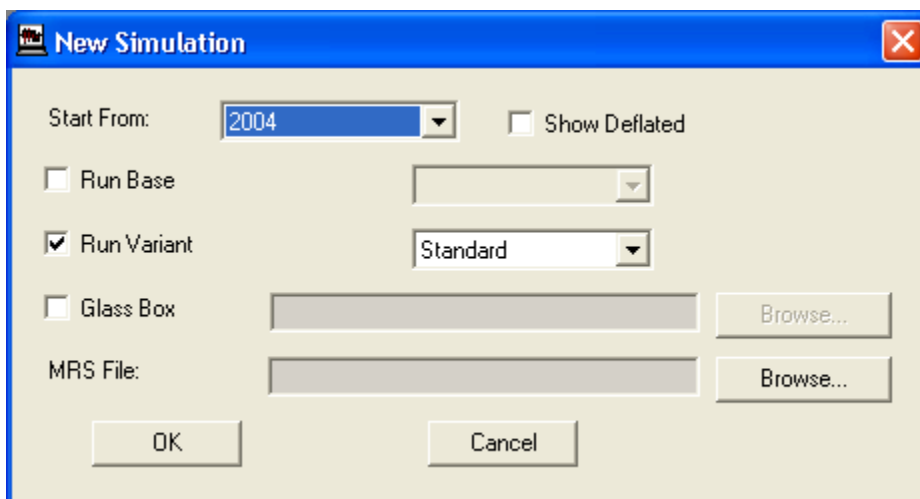
This section will describe how to run the SPSM using the Visual SPSM interface. The Visual SPSM provides a method of editing parameters, executing simulations, and examining output without knowledge of the command prompt environment.

Note that the screen snapshots provided are examples and may not exactly reflect the version you are currently using.

### CREATING, OPENING, AND SAVING SIMULATIONS

#### Creating a new simulation

To start a new simulation, click on **New Simulation** in the **File** menu or from the menu bar. You will get the following box:



This box allows you to choose which year to simulate and also what type of scenario you would like to simulate.

#### Choosing the year

New simulations always start with default parameters which were installed with the

product. You start them by opening a control parameter file. There are two kinds: normal and deflated. By default, the SPSM includes parameter files for various years. The years are listed in the **Start From** box. They represent the standard control parameter files which are called:

`baCY.cpr`

where CY is the last two digits of the year of interest. So `ba00.cpr` would be the default simulation for 2000 which would simulate the 2000 tax and transfer system on a population grown to represent 2000.

You also have the option to open deflated control parameters. Deflated simulations always run the base year population, but then apply other year's tax and transfer systems to them, deflated to the dollars of the base year. In the **Start From** box, they will be labelled something like `2008_06` which means that it's the 2008 tax and system deflated to 2006 dollars, run on the 2006 database. These deflated control parameter files are called:

`baCY_DY.cpr`

Where CY is the year of the tax and transfer system and DY represents the database year.

Lastly, you have the option to start from non standard tax and transfer simulations. These are simulations which have been saved in the default SPSD directory (either by you, or occasionally by the SPSD/M team). These simulations open your own tax and transfer system which you have created and assigned your own name to, for example `ba04new.cpr`. These `cpr` files will appear at the bottom of the drop-down box, under the last tax and transfer year the SPSD/M creates.

### **Choosing the executable**

By default, the Visual SPSM will run in Black-box mode, using the compiled executable which is included in the model. However, Visual SPSM can also be run in Glass-box mode by using executables created by users instead of the default executable which is included with the SPSM. Please see the [Programmer's Guide](#) for more details on how to create glass box executables.

To use a glass box executable instead of a black box executable, tick the **Glass Box** box and navigate to an appropriate glass box executable.

### **Choosing the scenarios**

To turn on the base or the variant scenario, you need to tick the appropriate boxes. There are two types of variant scenarios which are possible:

**Standard:** the standard tax/transfer algorithm will be used to calculate the scenario

**Alternate:** the alternate tax/transfer algorithm will be used to calculate the scenario (using a Glass Box executable)

There are three types of base scenarios which can be created. In addition to the Standard

and Alternate algorithms, the base scenario can also use the option:

**SPSM Results File:** that results output from a previous SPSM simulation will be used as the base scenario

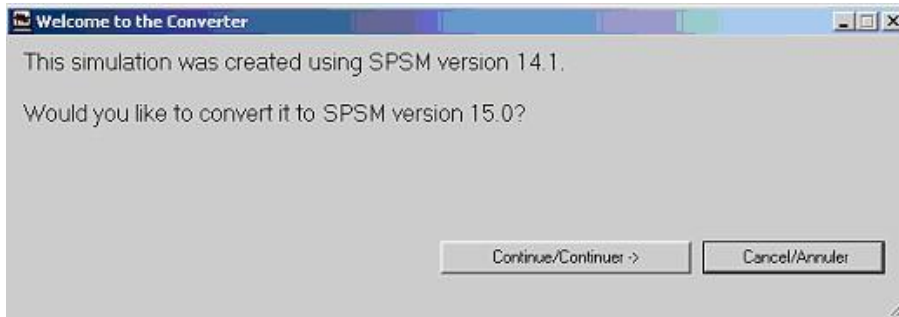
If the SPSM Results File is chosen, you must also indicate the location of the results to be read in. This must be done by setting the file in the **MRS File** box by using the "Browse" button.

These options can also be selected or changed from **File, Simulation Management**. See the [User's Guide](#) for more information on creating variant and base scenarios.

### Opening an existing simulation

To open a previously created simulation, click on **File, Open**.

If the simulation being opened was created in a previous version of Visual SPSM or in the current version of Classic SPSM, a .cpr file warning message will first appear and a Conversion Wizard will open. Note that only Classic SPSM simulations created in a previous version of the SPSM cannot be opened in Visual SPSM.



You will be prompted to save the new simulation under a different name. When the conversion process is finished you can open the new simulation, exit the wizard and return to Visual SPSM or review the log created during conversion. This log describes the conversion steps in detail. It highlights parameters that are omitted from the conversion and parameters which have been converted. Parameters that have different values from the default will be converted.

Once the simulation has successfully been converted into the current version of Visual SPSM, it must be opened and executed in order to update the results. Note that when opening a simulation created using a previous version of Visual SPSM, there may have been some changes to the SPSM which might invalidate your results. Refer to the Addendum to see what has changed in this release of the SPSM or contact the SPSM team for an Addendum for a previous version.

### Saving simulations

To save a simulation, click on either **Save** or **Save As...** in the **File** menu. Note that when you save a simulation, a number of files as well as some control parameters will be

saved.

## File types

When you save a simulation, the four full sets of parameter files (control, database adjustment, base tax/transfer, variant tax/transfer), will always be saved. If parameters have been changed from the default parameters, parameter difference files will also be saved. Finally, the current state of the various open windows will be saved so your interface will look the same the next time you open the simulation. Note that different output files may be created when you run a simulation. But when you use "Save As", the output files are not saved – they are only produced when running a simulation. The following tables list all the files which can be produced when you save a simulation:

### Database Inputs

.SPD	SPSD binary database file
.FXV	SHS household level expenditure data
.WGT	SPSD file containing survey weights

### Parameters

#### Complete parameters

.CPR	Control parameter file
.APR	Database adjustment parameter file
_B.MPR	Base Tax/transfer parameter file
_V.MPR	Variant Tax/transfer parameter file

#### Changed parameters

.CPD	Control parameter difference file
.APD	Database adjustment parameter difference file
_B.MPD	Base Tax/transfer parameter difference file
_V.MPD	Variant Tax/transfer parameter difference file

#### Visual SPSM Window state

.STA	Window state
------	--------------

While the following are the various output files which can be produced when simulations are run:

### SPSM Outputs

#### Printable

.TBL	Output tables result file
.PRN	Case output facility text results file

#### Input to further analysis

.MRS	Binary SPSM results file
------	--------------------------

## Control parameters which are changed when saving a simulation

When you save a simulation, certain control parameters will be changed to reflect the name of the saved simulation. These parameters are hidden, and will not be displayed. They are:

Parameter name	Parameter label	Example value (Simulation = test)
OUTCPR	Name of control parameter file (out)	test.cpr
OUTLOG	Name of log file (out)	test.log

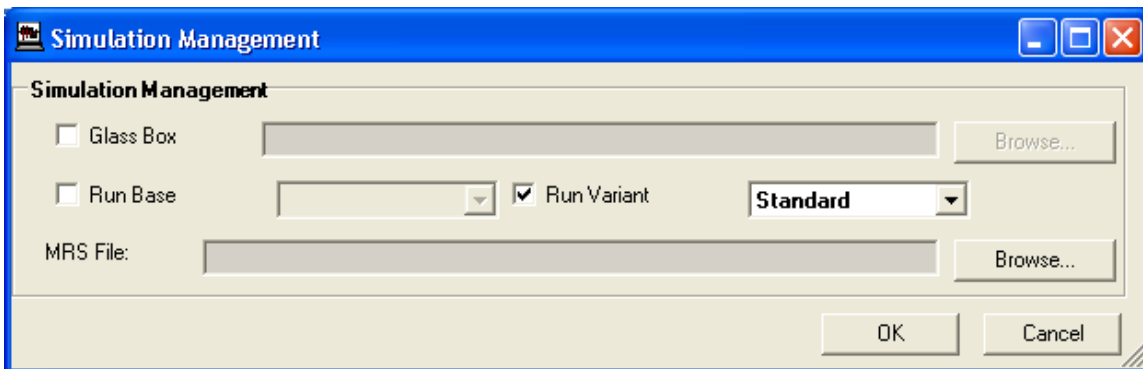
INPAPR	Name of database adjustment parameter file (in)	test.apr
OUTAPR	Name of database adjustment parameter file (out)	test.apr
INPVARMPR	Name of variant tax/transfer parameter file (in)	test_V.mpr
OUTVARMPR	Name of variant tax/transfer parameter file (out)	test_V.mpr
OUTVARMRS	Name of variant results file (out)	test.mrs
INPBASMPR	Name of base tax/transfer parameter file (in)	test_B.mpr
OUTBASMPR	Name of base tax/transfer parameter file (out)	test_B.mpr
OUTASC	Name of text output file (out)	test.prn
OUTTBL	Name of report file (out)	test.tbl

### Backup of output files

Backups of a simulation's .log, .tbl, and .prn files are done automatically in the Simulation Directory. Prior to a simulation being run a second time under the same name, Visual SPSM will browse the Simulation Directory, rename the simulation's existing .log, .tbl and .prn files to .log.bak, .tbl.bak and .prn.bak respectively, and create new files with results from the newly run simulation.

### CHANGING SIMULATION SETTINGS

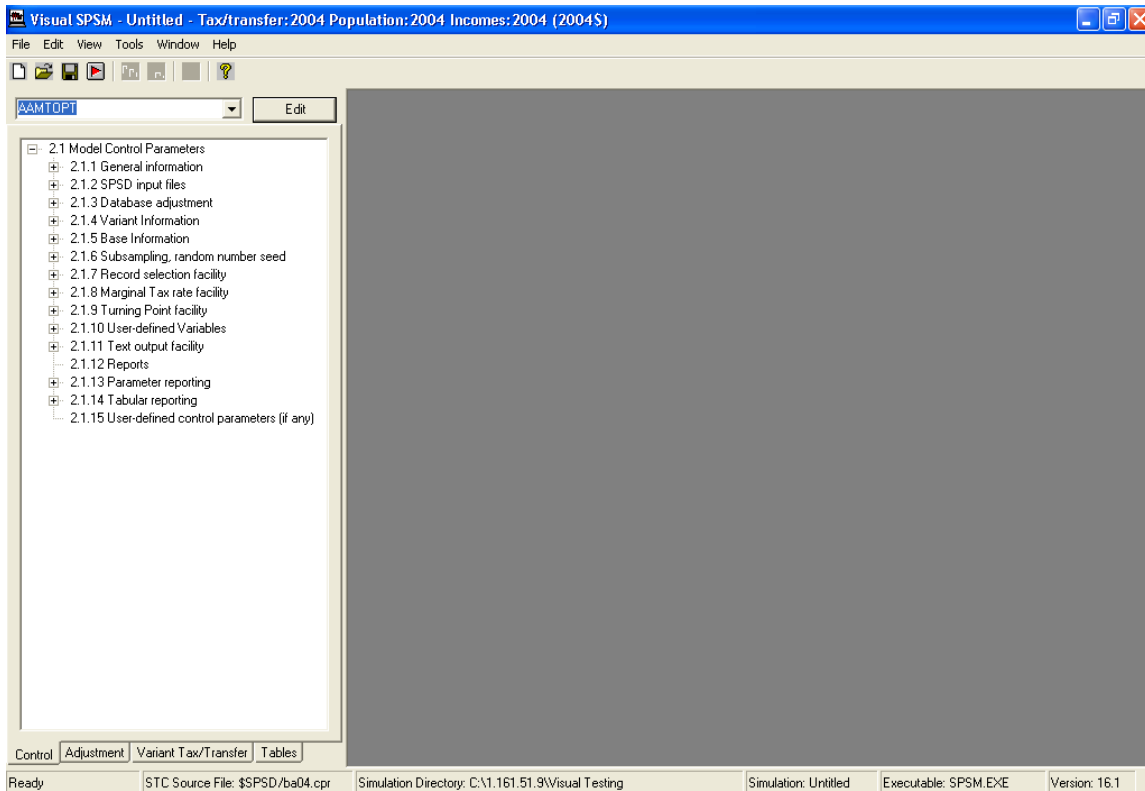
You can change the simulation settings of the current simulation by going to **File, Simulation Management**. The following window will open:



In this window, you can change the settings of the current simulation.

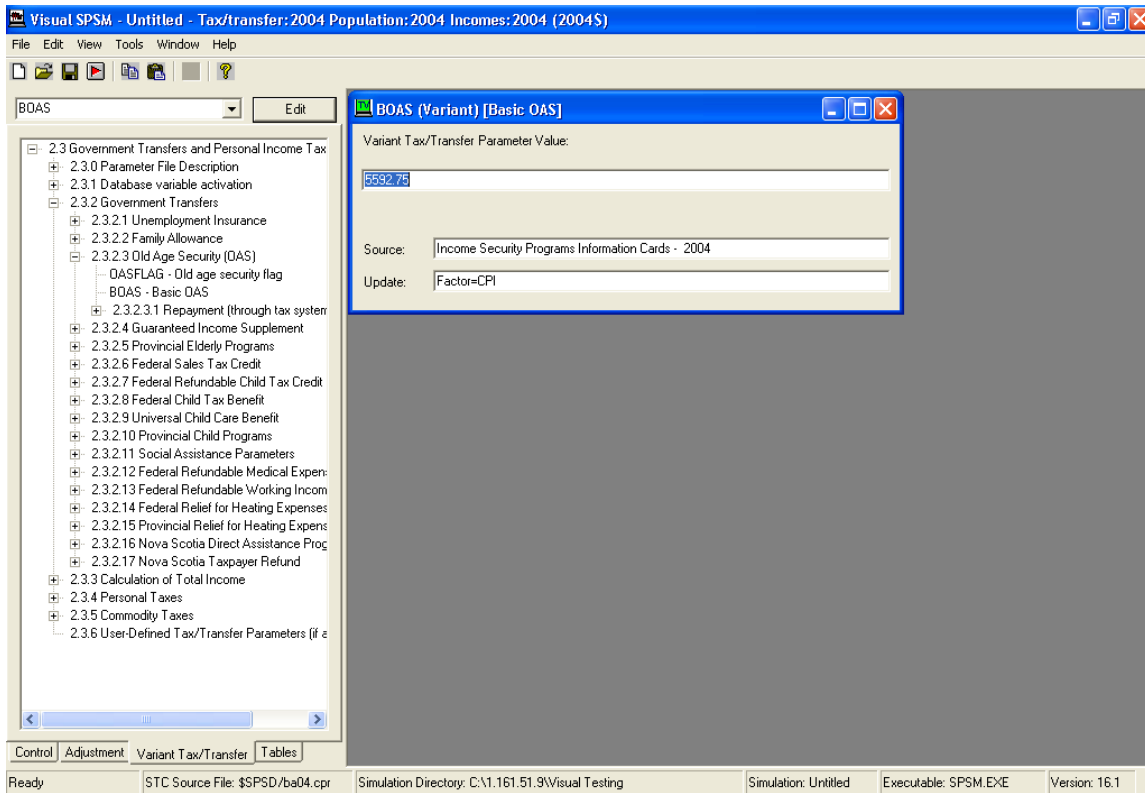
### EDITING PARAMETERS

Once you have created or opened a simulation you can edit the parameters before executing the SPSM. The next figures show the basic windows of Visual SPSM.



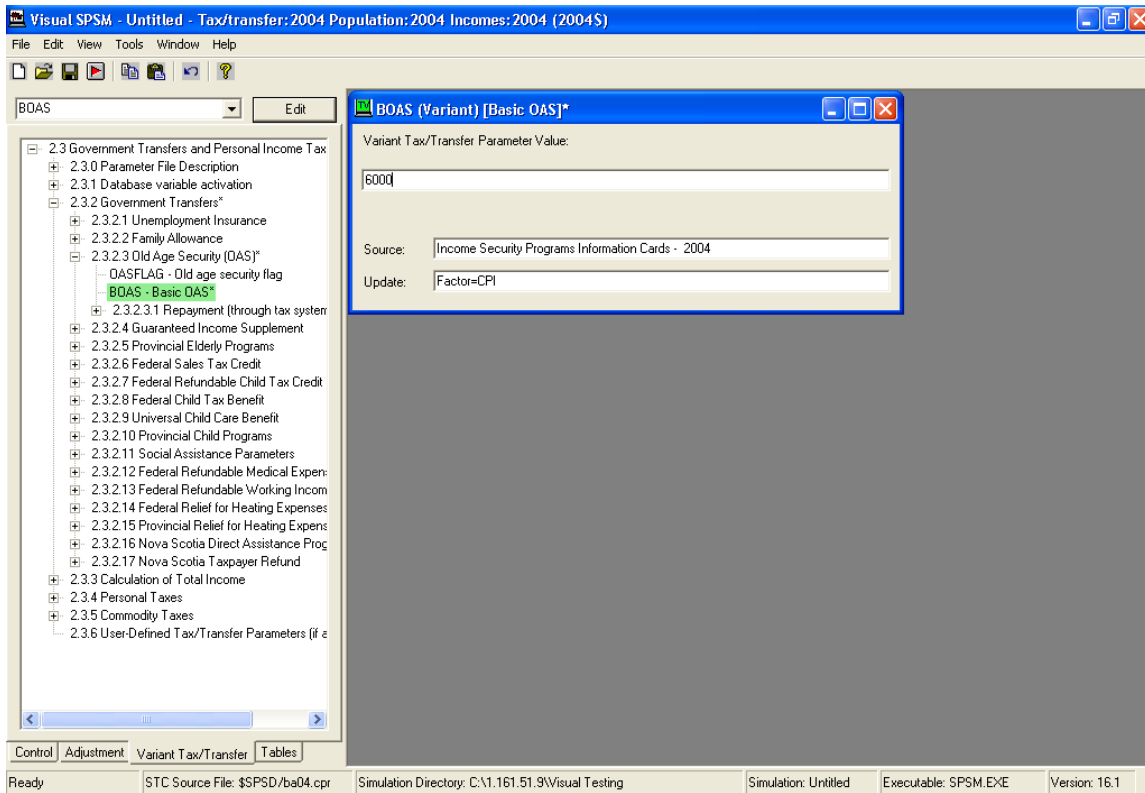
There are four tabs at the bottom left of the window. The first three will open panes with parameter trees: control parameters, adjustment parameters, and variant tax and transfer parameters. **Note that the Base tax/transfer parameters will only appear if a base scenario has been selected, and the Variant tax/transfer parameters will appear if a variant scenario has been selected. Base parameters are only used when BASMETH is set to 2 or 3 and variant parameters are only used when VARMETH is set to 2 or 3.**



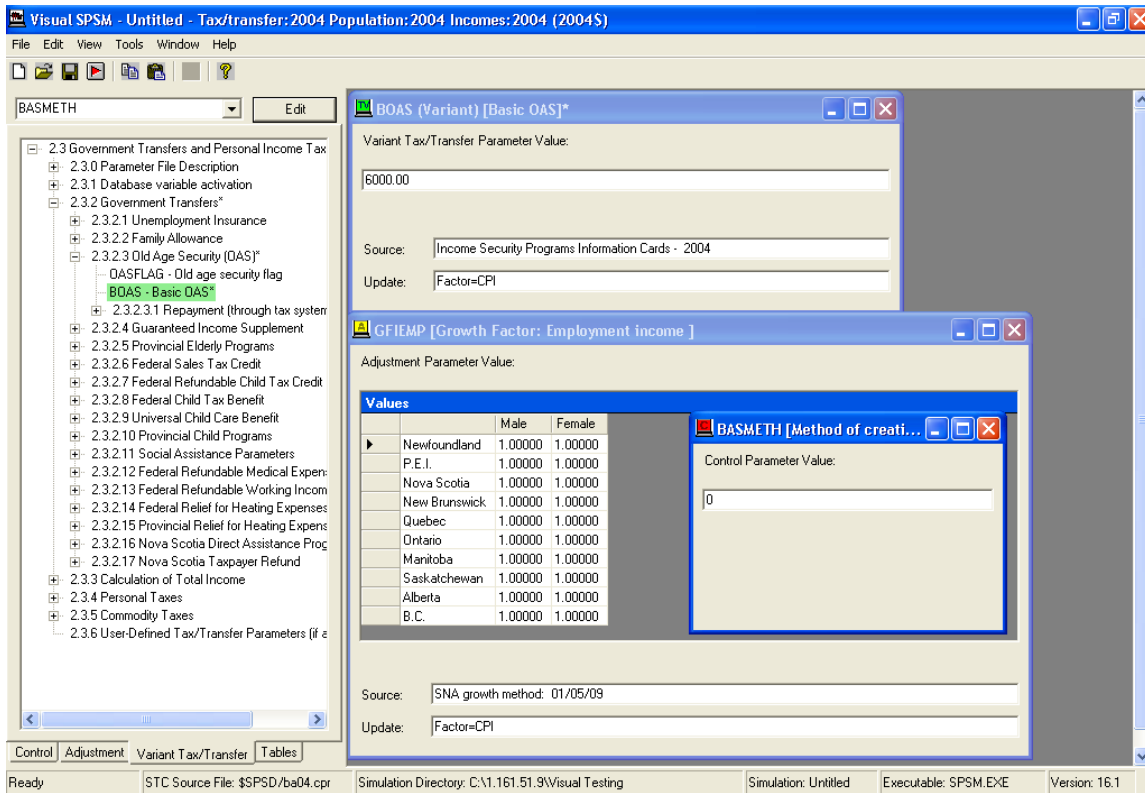


There are two methods to open the parameter you want to edit. One method is to find the parameter in the hierarchical tree. For example, if you wanted to edit the basic amount of Old Age Security (OAS) in the variant scenario, you would first click on the Variant Tax/Transfer tab. Then you'd open the tree by clicking on Government transfers, then Old Age Security, and then finally double click on BOAS which will open the BOAS window. If you already know the name of the parameter that you are looking for, you can also simply type in the name in the box above the tree and then click the Edit button. The hierarchy tree will open to display BOAS in the Variant Tax/Transfer tab and the parameter window will open. Note that you can input any type of parameter from any tab in this edit box. So, for example, if you are in the Control tab and type in GFIEMP, then the parameter window for GFIEMP opens, even though it's an adjustment parameter. Note that when you input a tax/transfer parameter in any tab but the Base Tax/Transfer, the variant value will be opened. You can only open the Base parameters from the Base tab.

Once a parameter window is opened, you can then change the parameter's value, source, and update factor. When you change the value of a parameter, the parameter name in the hierarchy tree is highlighted and an asterisk is placed beside the name. An asterisk will also appear in the title of the parameter box. Once you save the simulation, the highlighting and asterisks will disappear. **You should always save a simulation prior to executing it.**



Note that parameter windows have colour coded icons in the top left hand corner (see figure below). Control parameters have a red icon with a C in it. Database adjustment parameters have a yellow icon with an A in it. All tax and transfer parameters have green icons. The base tax and transfer parameters have a TB in the icon while the variant tax and transfer parameters have a TV in the icon.



## Copying and pasting parameter values

You can copy parameter values into other programs for editing, such as Notepad or Microsoft Excel. This may be useful if you have many changes to make in a parameter. You can do this by clicking on **Edit**, **Copy** or **Edit, Copy Table**. For example, if you open GFIEMP, click on Edit, Copy Table, and paste it in Excel you are now able to edit the values. When finished, copy the table values then in Visual SPSM click on **Edit**, **Paste Table**.

You can also add rows to or delete rows from a parameter using the **Add Row** or **Delete Selected Row** buttons found in each table parameter. If you try to add or delete a row when editing a parameter in Notepad or Excel, these changes won't appear when you paste the parameter back into Visual SPSM.

## Undo / restore parameter values

**Edit, Undo Parameter Edit** will undo the changes, one digit at a time, to a parameter value since the last time the simulation was saved. **Edit, Restore Data** will bring back the entire value of the parameter since the last time the simulation was saved. If the simulation hasn't yet been saved, Restore Data will bring back the default parameter value.

## Hidden Parameters

Hidden parameters are parameters that are not visible in Visual SPSM and not available for editing. BRKFLAG, CLOSEFLAG, ETAFLAG, UPDATEINT, and LICENSEE do

not apply to Visual SPSM and are therefore not visible. OUTAPR, OUTASC, OUTCPR, OUTVARM, OUTBASM, OUTLOG and OUTTBL will be automatically overwritten when saving/running a new simulation. INPAPR, INPBASM, INPVARM should not be used in Visual SPSM -> please use the Read Parameter File facility instead.

## EXECUTING SIMULATIONS AND VIEWING THE LOG WINDOW

Once you have finished editing the parameters you should **save your simulation** (so that your changes are included in the simulation) and then execute the simulation. Visual SPSM will then call spsm.exe which runs the simulation in the same manner as Classic SPSM.

To run a simulation, click on **File, Execute Simulation** or press the Execute button in the toolbar. Once the run is complete, a box will open which will indicate if there were any errors or warnings produced during the simulation, or whether the simulation ran without problems.

If an error or warning is found, the log window is opened. You should look at this window to find the error or warning, which will be highlighted in red or yellow respectively. The following section gives a brief description of what is seen in the log window.

### Contents of Log Window

Classic SPSM, immediately after being invoked, displays a copyright notice and a greeting screen. The dialogue which follows consists basically of six prompts, which are described in turn below.

1) Enter name of input control parameter file ==>

This will be the name of your saved simulation. It will point to all the new parameter files which were created when the simulation was saved.

2) Enter specification for generating output files ==>

This will also be the name of your saved simulation.

3) Do you wish to modify any control parameters ? ==>

The answer will be no. If there were any errors or warnings in the control parameters, they will be highlighted in colour here. An example:

```
Do you wish to modify any control parameters ? ==> N
error(962): 'hdprov' is not an analysis variable in XTSPEC # 1
error(964): no analysis expression in XTSPEC # 1
error(948): parameter error in batch mode
```

4) Any further control parameter changes ? ==>

The answer will always be no. The model will then load the database adjustment parameters. **If there are any errors in them, they will be listed prior to the sample parameters shown.**

5) Do you wish to modify any database adjustment parameters ? ==>

The answer will be no. Note that any errors will be listed before this section. The model will then read the base and variant tax and transfer parameters. **If there are any errors in them, they will be listed prior to the sample parameters shown.**

6) Do you wish to modify any variant tax/transfer parameters ? ==>

Depending on the version of Visual SPSM, it is possible that a parameter may be changed here for technical reasons. This has no impact on results.

The SPSM now performs the simulation. At the conclusion of the simulation, SPSM will output some summary information and list the names of all output files produced.

When the Log file is open, you have the option to Print it from this window.

## VIEWING TABLES AND DISTRIBUTIONS

Once a simulation has been run, you can view any tables or distributions which were produced. They can be found in the Tables tab. The built-in tables are shown first, followed by the user tables and then the distribution.

By default, only Table 0 and Table 1 are produced. See the *User's Guide* for more information on how to turn on the other default tables, create user tables, or use the Distributional Analysis facility.

You are able to copy the values of table rows and columns to the Clipboard, and paste into another application. A 'Copy Table' button can be found in the table header.

The screenshot shows the Visual SPSM software interface. The title bar reads "Visual SPSM - untitled - Tax/transfer: 2004 Population: 2004 Incomes: 2004 (2004\$) - [Table 0]". The menu bar includes File, Edit, View, Tools, Window, and Help. The main window is divided into a left-hand tree view and a right-hand table view.

The left-hand tree view shows a hierarchy of categories under "2.3 Government Transfers and Personal Income Tax":

- 2.3.0 Parameter File Description
- 2.3.1 Database variable activation
- 2.3.2 Government Transfers
  - 2.3.2.1 Unemployment Insurance
  - 2.3.2.2 Family Allowance
  - 2.3.2.3 Old Age Security (OAS)
    - OASFLAG - Old age security flag
    - BOAS - Basic OAS
    - 2.3.2.3.1 Repayment (through tax system)
  - 2.3.2.4 Guaranteed Income Supplement
  - 2.3.2.5 Provincial Elderly Programs
  - 2.3.2.6 Federal Sales Tax Credit
  - 2.3.2.7 Federal Refundable Child Tax Credit
  - 2.3.2.8 Federal Child Tax Benefit
  - 2.3.2.9 Universal Child Care Benefit
  - 2.3.2.10 Provincial Child Programs
  - 2.3.2.11 Social Assistance Parameters
  - 2.3.2.12 Federal Refundable Medical Expenses
  - 2.3.2.13 Federal Refundable Working Income
  - 2.3.2.14 Federal Relief for Heating Expenses
  - 2.3.2.15 Provincial Relief for Heating Expenses
  - 2.3.2.16 Nova Scotia Direct Assistance Program
  - 2.3.2.17 Nova Scotia Taxpayer Refund
- 2.3.3 Calculation of Total Income
- 2.3.4 Personal Taxes
- 2.3.5 Commodity Taxes
- 2.3.6 User-Defined Tax/Transfer Parameters (if e

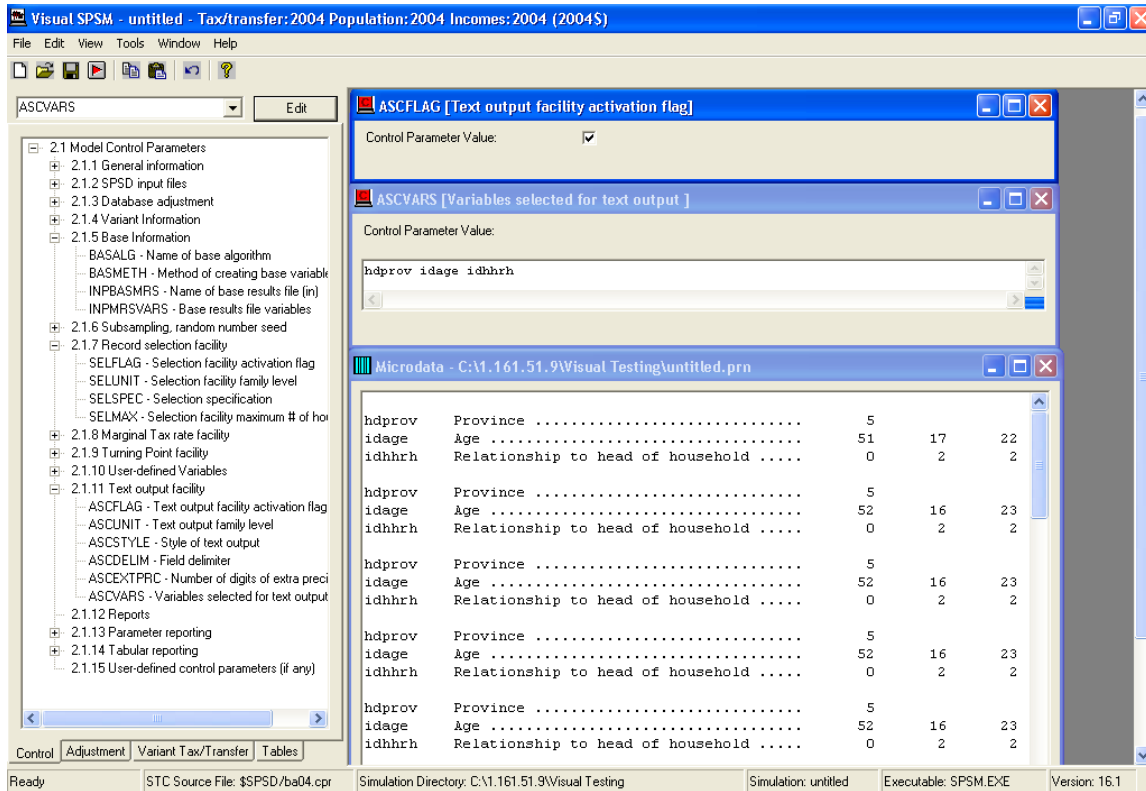
The right-hand table view displays "Results for Census Families" with a "Copy Table" button. The table has two columns: "Variable (x1,000,000)" and "TOTAL".

Variable (x1,000,000)	TOTAL
Family Units (x1000)	15252.9
Persons (x1000)	31510.6
SLID Survey Records	34886.0
SPSD Records	97580.0
Income (Base)	599080.8
Income (Variant)	599080.8
Change	0.0
Number of Gainers (x1000)	0.0
Number of Losers (x1000)	0.0
No Change (x1000)	15252.9
Gainer's Gain	0.0
Loser's Loss	0.0
Total Income	854633.4
Market Income	752302.6
Wages and Salaries	564670.7
Self-Employment Income	52654.9
Investment Income	57670.0
Other Income	77306.9
Transfer Income	102330.8
Total Tax	255552.9
Net Transfers	-153222.1
Disposable Income	672777.9
Consumable Income	599080.8
Federal Taxes	159347.5
Federal Income Tax	95414.2
UI/EI Contributions	7711.0
UI/EI Benefit Recovery	232.0

The status bar at the bottom shows: Ready, STC Source File: \$SPSD/ba04.cpr, Simulation Directory: C:\1.161.51.9\Visual Testing, Simulation: untitled, Executable: SPSM.EXE, Version: 16.1.

## VIEWING MICRODATA

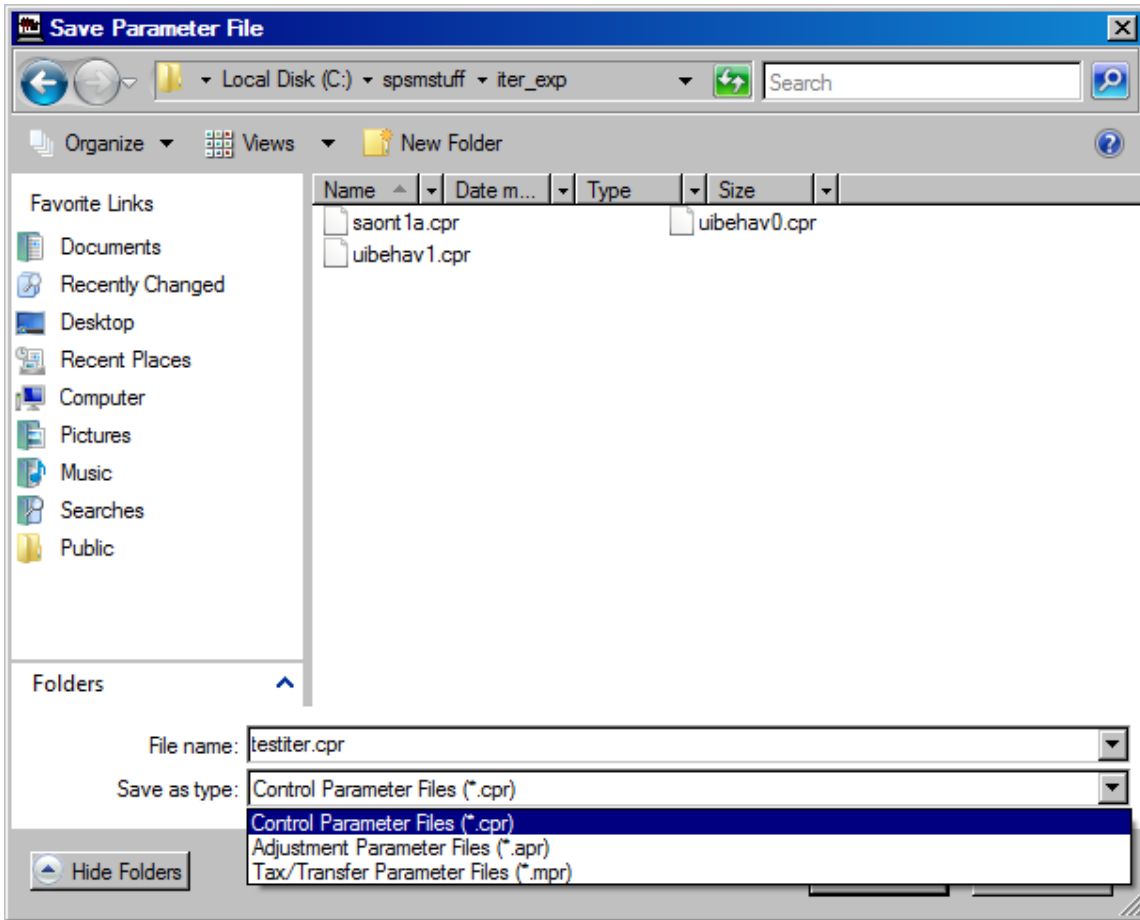
If you have produced textual microdata by using the Text Output Facility or the Turning Point Facility, you can view it by clicking on the **View, Microdata** in the menu bar. Large files may be truncated (a message will be displayed). You also have the option to Print and/or Print Preview the Microdata. Depending on the amount of printed text, the Print and Print Preview may also truncate the data.



## READING AND SAVING PARAMETER FILES

You may wish to use the same parameters files in more than one simulation. One way to do this is to read or save parameter files.

When you click on **File, Save Parameter File**, you get the following box. Pick the parameter type you wish to save. If you pick a tax/transfer parameter type, you will later be asked to specify whether you wish to save the base parameter or the variant parameters.



Two files will be saved. One will contain the full set of parameters for the current simulation and the second will be the parameter difference file which only contains the parameters which are different from the default.

	Full parameter files	Parameter difference files	Optional Parameter include files
Control parameters	test.cpr	test.cpd	test.cpi
Adjustment parameters	test.apr	test.apd	test.api
Base tax/transfer	test_B.mpr	test_B.mpd	test.mpi
Variant tax/transfer	test_V.mpr	test_V.mpd	test.mpi

These files are now available to be read by other simulations. Note that when simulations are saved in Visual SPSM, the full parameter files and the parameter difference files are always saved for all four parameter types.

To read in previously produced parameters, you click on **File, Read Parameter File**. Parameters will be highlighted as changed in the hierarchy tree. In addition to the full parameter files and the parameter difference files, you can also read in partial include files (\*.cpi, \*.api, \*.mpi). These files are created by users with a text processor and contain only selected parameters. They can be very useful since they allow you to mix and match parameters. They can be created using Notepad.

### **Configuring your machine to write include files using Notepad**

We encourage SPSM users to use include files in order to organize their work. In order to write these files, you'll need a text editor. Windows come with a basic text editor called Notepad. You may notice that when using Notepad a .txt extension is automatically added to your include files (\*.api; \*.mpi; \*.cpi).

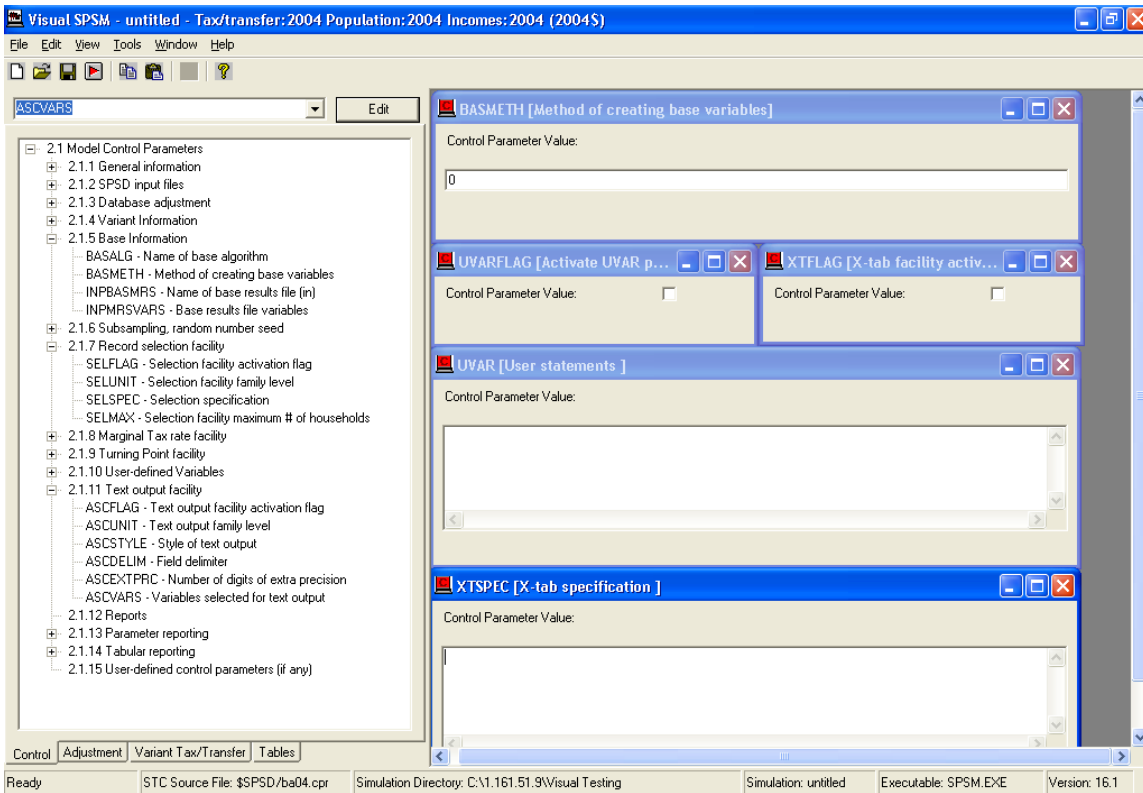
Should you encounter this problem, go to Windows Explorer. On **Windows 2000/XP** machines, open Tools/Folder Options/File Types/New. In the "File Extension" field, type in "CPI" and select "Ok". Repeat the steps for API and MPI files, and you'll find that the automatically generated .txt extensions disappear.

### **WINDOW STATES**

If you frequently modify certain parameters, you may wish to be able to group them together so that you can open them at once. It can be time consuming to do this repeatedly, so you might wish to open and resize the windows you want, and then save the window state. This will create a file which will contain information on which windows were open, as well as the size and location of the windows. It will **not** save the content of the parameters in the windows. By doing this, when working on a new simulation, you can simply load the window state to easily view and edit some of your favorite parameters.

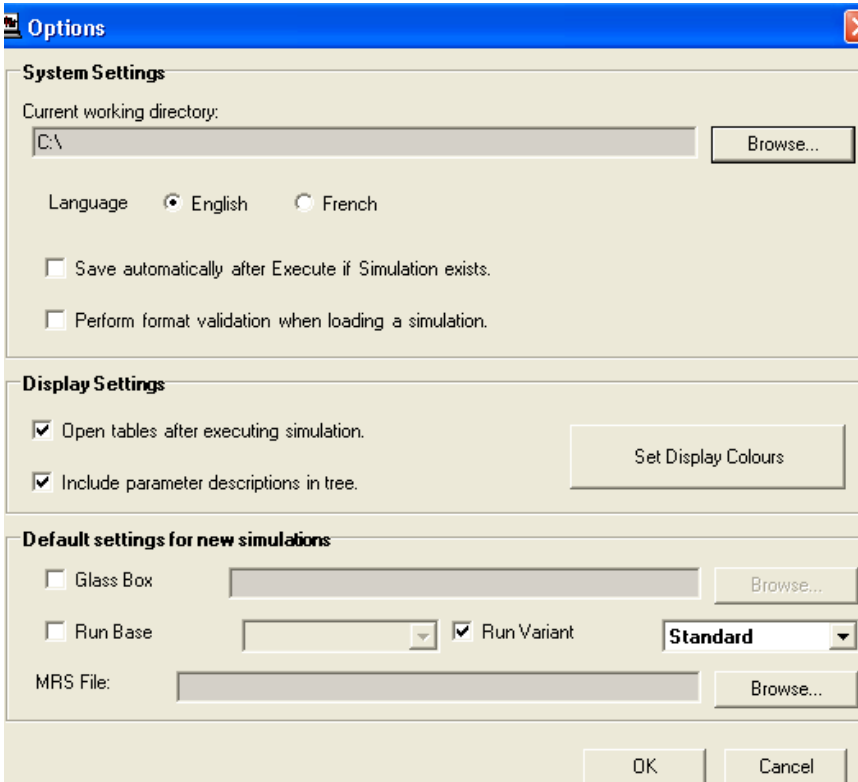
To save a window state, go to **File, Save Window State**. To load a window state, go to **File, Load Window State**.





## VISUAL SPSM OPTIONS

If you click on **Tools, Options**, you can change the default options.



The current working directory is the default location for opening and saving simulations. Note that once you open or save a simulation in a different directory, the current working directory will automatically be changed to that directory.

You can change the Visual SPSM language. It too will only take effect the next time Visual SPSM is opened.

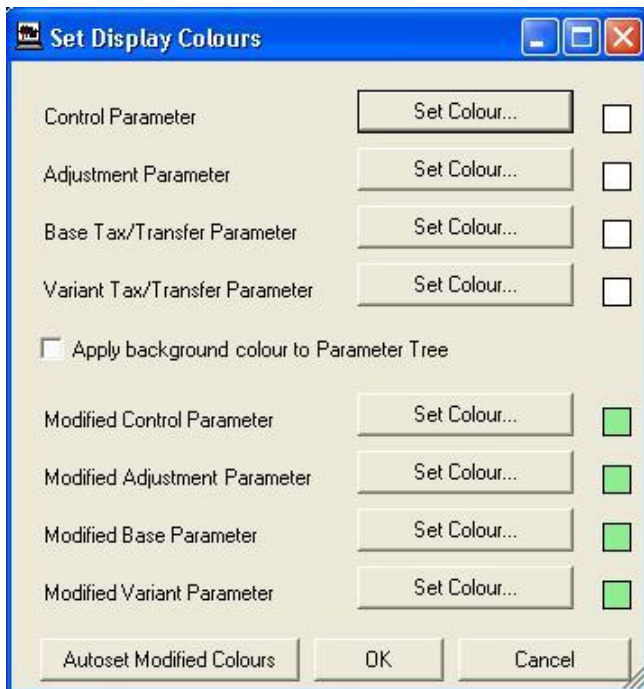
Saving automatically upon execution of your simulation if it exists is useful if you plan to overwrite your simulation each time it is run. If this option is off, you will be prompted to save your simulation upon execution, which gives you the option of renaming it.

You can ask Visual SPSM to perform validation of parameter values when a previously created simulation is loaded or a parameter file is read. This will verify that the values fall within pre-defined formats. If the validation check fails, the simulation will load but a message will appear listing the parameters that failed. The simulation will not run until the values are valid.

There is an option which will open built-in Table 0 and user Table 1 (if available) after a simulation is executed.

You also have the option to display parameter labels in the parameter hierarchy tree. It will only take effect the next time Visual SPSM is opened.

You have the option of changing the default visual display colours. You can add colour to the inside of parameter boxes, or change the colour to the parameter name highlighting in the hierarchy tree, or add colour to the background of the parameter tree.



Finally, you have the option to change the settings for the executable choice (glass box or

black box), and the settings for the base and variant scenarios.

## **FUNCTION KEYS**

Some function keys can be used in Visual SPSM.

F1 – Open the SPSD/M on-line help documentation

F2 – Go to parameter name drop-down box

F3 – Execute Simulation

F4 – Open built-in Table 0

F5 – Open XTFLAG parameter box

F6 – Open UVARFLAG parameter box

F7 - Open SELFLAG parameter box

F8 - Open ASCFLAG parameter box

F9 - Open XTSPEC parameter box

F10 – Open UVAR parameter box

F11 – Open context-specific help documentation

F12 – Hides or displays the hierarchy tree

## **HELP**

Under the Help menu, you can find the SPSM help file, which gives you information about the SPSM (and which includes this guide). Parameter Help provides context-specific information for the parameter which is selected and open for editing. Visual SPSM Release Notes provide information on enhancements that have been made and problems that may be encountered in the current version of the Visual SPSM interface. About Visual SPSM provides information about this version of Visual SPSM as well as information on the spsm executable used.

## **USING TOOLS**

Some SPSM tools can be used with Visual SPSM. Please see the *[Tools User's Guide](#)* for more information about how to use these tools.

## **Running SPSM using Classic SPSM**

Running the SPSM using the Classic SPSM interface means that you will be running the model using an MS-DOS environment. When you click on **Classic SPSM**, a command prompt will open with the correct environment variables and path set.

## **MS-DOS OVERVIEW**

For people unfamiliar with the MS-DOS environment, accessed using the command prompt, this section will serve as a brief refresher.

### **Files**

A file is a collection of stored information. Because you can store many files on disk, naming files is an important task. Good filenames are descriptive. File names for the SPSM should not contain any spaces. A filename may also have an extension, although one is not required. An extension always begins with a period followed by characters (normally three). The extension allows you to group files into categories by giving more information about what is in the file.

### **Directories**

Because many files may be created, you need a method of organizing files so you can efficiently store and retrieve data. A directory (or folder) is used to organize groups of related files. First you determine how your files are related and then you create a directory and add your files to it. Directories are stored on the disk with the files. We recommend that you do not use spaces when naming directories.

The root directory, or simply, the root, is the main directory on a hard or floppy disk. The root directory cannot be deleted. A subdirectory is a directory which is below, and included in, another directory. It is any directory other than the root. The current directory is the directory in which you are currently working. The current directory is the directory which MS-DOS accesses when you specify a file without specifying a directory. When you open Classic SPSM, the current directory is set as your default working folder. A parent directory is the directory one level above the current directory in the hierarchy.

### **Pathnames**

Before MS-DOS can find a file in a directory that is not in the current directory you must tell it where to locate it. You do this by specifying a pathname. A pathname consists of a drive specifier, a series of directory names, and a filename. A backslash separates each of the elements. For example the following pathname specifies that the file README.TXT is located in disk drive D: in the directory SOFTWARE which has the parent directory USERS. The root is always indicated by a single backslash.

```
D:\USERS\SOFTWARE\README.TXT
```

When running an executable which is not in the current directory, you must either specify a pathname as for files, or include the path of the environment variable PATH. The directory where the SPSM executables exist is included automatically in this environment variable when you open the Classic SPSM shortcut.

### **Using Directories**

MS-DOS uses several commands to list, create, locate, change and delete directories. The key actions are to create, delete and change directories.

Command	Short Form	Purpose
MKDIR	MD	Add a new directory to the tree structure
RMDIR	RD	Remove a directory from the tree structure
CHDIR	CD	Display the current directory or change the current directory

All these commands take as their argument a directory name. For example, MKDIR \USERS will create a directory called USERS one level below the root. The command CHDIR \USERS will make \USERS the current directory. If you wish to display the name of the current directory simply enter the CHDIR command with no arguments. The command RMDIR \USERS would then delete the USERS directory. Note that you cannot delete the root or the current directory. The RMDIR command will not remove a subdirectory that contains files or other subdirectories; thus users must delete any contained files before removing a subdirectory.

### Other Useful Commands

Here are other commands which may prove useful:

Command	Purpose
COPY	Copies a file or a directory
DEL	Deletes a file or a directory
DIR	Lists files in directory
EXIT	Exits the command prompt
REM	Indicates that the rest of the line is to be taken as a comment
SET	Displays, sets, or removes environment variables

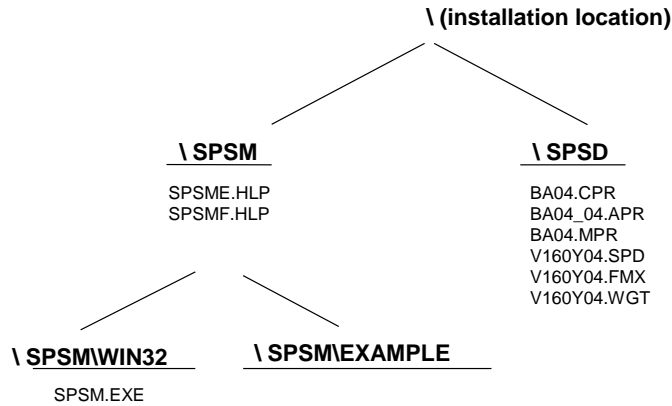
### SPSD/M FILES AND DIRECTORIES

All SPSP/M files are maintained using a directory structure that follows certain conventions. Directories can also be referred to as folders under some operating systems. Users are strongly urged to organize their work in directories as well. Directories provide a structure for organizing files on disk.

Executable files have either a `.exe`, `.com`, or `.bat` extension. They are executed by simply typing the file name (without the extension), followed by a carriage return at the command prompt. The `spsm.exe` program is an example of such a file. Assuming that you have chosen the default installation directory, the full pathname of the executable SPSP/M program for Version 16.0 is something like:

```
c:\program files\statcan\spsdm16.0\spsm\win32\spsm.exe.
```

The following figure shows part of the SPSP/M directory structure and selected file contents.



During the user dialogue several specifications of input and output files are required. If the user does not specify a full pathname when specifying a file, the file will be read from, or written to, the current directory.

When about to begin an analysis using the SPSM, the user should create a new directory to store all the input parameter files and outputs. Any name may be selected for this directory. Directories can be created either in the command prompt using the MKDIR command, or by using "My Computer".

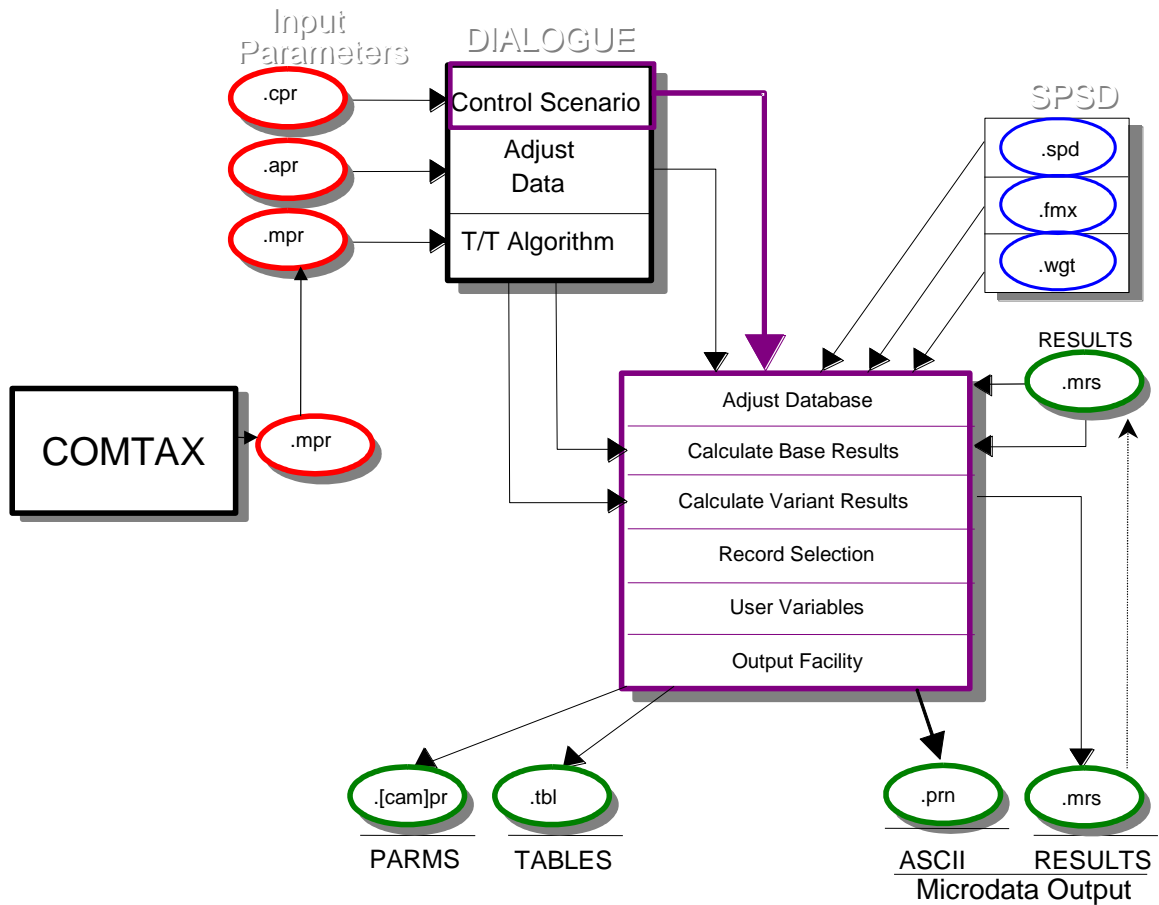
The following table gives the file extension naming conventions used for SPSM input and output files.

Database Inputs		
	.SPD	SPSD binary database file
	.FXV	SHS household level expenditure data
	.WGT	SPSD file containing survey weights
Parameter Inputs		
Complete and mandatory		
	.CPR	Control parameter file
	.APR	Database adjustment parameter File
	.MPR	Tax/transfer parameter file
Partial and optional, created by user		
	.CPI	Control parameter include file
	.API	Database adjustment parameter include file
	.MPI	Tax/transfer parameter include file
Partial, created by Visual SPSM		
	.CPD	Control parameter difference file
	.APD	Database adjustment parameter difference file
	.MPD	Tax/transfer parameter difference file
SPSM Outputs		
Printable	.TBL	Output tables result file
	.PRN	Case output facility text results file
Input to further analysis	.MRS	Binary SPSM results file
Program Run Control		
	.BAT	SPSM Dialogue commands
	.CMD	SPSM Dialogue commands
	.CTL	SPSM Dialogue commands
Visual SPSM state		
	.STA	Visual SPSM state

### SPSD/M File Naming Conventions

The database files are all named by Statistics Canada and should not be changed. There are many types of input parameters. The parameter files with .CPR, .APR and .MPR must be complete and valid SPSM parameter files and values for all three must be specified for any given program run. The .CPI, .API, and .MPI files are created by the user with a text editor and contain subsets of the corresponding .CPR, .APR and .MPR files. The .CPD, .APD and .MPD are similarly subsets of parameters but they are created automatically by Visual SPSM.

The relationships of all these files to the structure of the SPSM is shown in the following diagram. File types with their extension are shown in ellipses.



## SPSD/M Files and Relationships

This diagram can be used as a handy reference while becoming familiar with the files and naming conventions.

### SPSD ENVIRONMENT VARIABLES

Environment variables can be used to generalize the names of various input files used by SPSM. For example, when running Version 18.0, the `.cpr` files contain entries such as the following:

```
INSPSD $SPSD/v180y06.spd
```

The SPSD environment variable is set when you open the Classic SPSM shortcut. If you used the default locations when installing the SPSD, the preceding is equivalent to

```
INSPSD c:/program files/statcan/spsdm18.0/spsd/v160y06.spd
```

This file name expansion only works in the leading position of the path name. If the environment variable is not defined, the name is not expanded and an error message will result.



The Classic SPSM shortcut defines the SPSD and SPSM environment variable and adds the appropriate directory to the PATH environment variables so that when you run the spsm from any subdirectory, it will know where to find the executable.

## **SPSM CONTROL**

Overall operation of SPSM is controlled through a user dialogue which has a number of distinct phases. The overall result of this dialogue, however, is the creation of one or more parameter files, which together provide a complete specification of the SPSM run.

The most important of these files is the control parameter file. Every execution of the SPSM produces one such file, which contains, among other things, a complete description of all other files used or produced in the run. These other files are specified through string parameters in the control parameter file. The conventions used are the same as those of the operating system. For MS-DOS these conventions are as follows:

- a) Names can be given in upper or lower case; case is not significant.
- b) If the drive specifier is absent, the current drive is assumed.
- c) If a path specifier is absent, the default directory for the current drive is used. Either forward or backward slashes may be used to delimit the elements of the path.
- d) File names consist of characters, followed by a dot, followed by an extension (normally three characters). We recommend that spaces not be used in file names.

## **SPSM Dialogue Structure**

SPSM, immediately after being invoked, displays a copyright notice and a greeting screen. The dialogue which follows consists basically of six prompts, which are described in turn below.

1) Enter name of input control parameter file ==>

A control parameter file contains a fair number of parameters. Typically, the user will wish to perform a run similar to one previously performed. He or she should enter the name of an existing control parameter file similar to that desired in response to this prompt. A fully specified file name may be specified (e.g. `$spsd/ba08.cpr`). The `.cpr` file extension will be automatically generated if omitted. SPSM may produce error messages at this point if syntactic problems occur in the specified file.

2) Enter specification for generating output files ==>

The control parameter file has a number of parameters that specify the file names of output files that the SPSM may produce. Specifically, these parameters are OUTCPR, OUTAPR, OUTVARMPR, OUTBASMPR, OUTVARMRS, OUTASC, and OUTTBL. It is unlikely that all of these files will be produced in a given run, although the file given by OUTCPR, which contains the control parameters for the run, is always produced. If the user enters a file name in response to this prompt, names for all output files will be generated from it by changing the file extension. For example, if the user enters `test1`, then the generated output file names will be `test1.cpr`, `test1.apr`, etc. A fully

qualified file name (e.g. /tmp/junk) can also be specified. Using this technique, all output files can be directed to some other directory. If the user just presses ENTER then the values of these parameters will remain unchanged from those of the input control parameter file. Note that any control parameter, including those specifying output file names, can be changed in Step 3) below.

3) Do you wish to modify any control parameters ? ==>

If the user answers YES to this question, then a dialogue allowing changes to control parameters will follow. The form of this dialogue is described in the section titled Parameter Editing Facility. After all changes have been made, the user issues the GO command. At this point SPSM checks the control parameters for consistency and validity, and issues error messages if it finds any problems.

4) Any further control parameter changes ? ==>

If error messages were issued, or if the user remembered additional changes that should have been made, the response to this question should be YES. If this is done, the dialogue will return to Step 2) above, otherwise the new values of the control parameter are written to the file OUTCPR and the dialogue continues with Step 5).

5) Do you wish to modify any database adjustment parameters ? ==>

After reading in the database adjustment parameters from the file specified in the INPAPR control parameter and displaying selected values from that file, this prompt will be displayed. As before, answering YES will activate the parameter editing facility, allowing the user to make changes to the database adjustment parameters. If any database adjustment parameters are changed as a result, SPSM will modify the control parameter INPAPR to be equal to OUTAPR and print a message to that effect. This means that if the resulting control parameter file is used as input in a subsequent run, the correct database adjustment parameter file will be referenced.

6) Do you wish to modify any variant tax/transfer parameters ? ==>

This prompt is very similar to Step 5), except that the user has the opportunity to change variant tax/transfer parameters. If no variant results are being produced (i.e. VARMETH was set to 0) then this prompt will not be displayed.

After these six dialogue steps are complete, SPSM will perform the run. The run may be interrupted at any time by pressing CTRL-BRK or CTRL-C. CTRL-BRK will stop SPSM without producing information while CTRL-C will stop SPSM and will produce information on the number of persons and household process and the size of the sample. The user will then be asked whether the run should be continued or terminated. At the conclusion of the run, SPSM will output some summary information and list the names of all output files produced.

Two general points should be noted. First, the dialogue can be interrupted at any point by pressing either:

CTRL-BRK. If this is done during the dialogue, SPSM will terminate, returning the user to the operating system.

CTRL-C the user will be asked whether the run should be continued or terminated.

Second, the parameter editing facility can be used, in Steps 3), 5), and 6), to examine parameter values as well as change them.

## **PARAMETER EDITING FACILITY**

The parameter editing facility provides a method by which a user may change the values of parameters without using a text editor to modify a parameter input file. The section titled SPSM Dialogue Structure describes how to invoke the parameter editing facility during the SPSM dialogue. This section describes how the facility operates.

After the parameter editing facility has been invoked, a prompt (==>) appears. At this point the user may enter a command, or the name of a parameter. Entering a parameter name will allow the user to examine or modify that parameter. The valid commands that can be given are LIST, which displays the names of currently accessible parameters, and READ, whose effect is described in the section titled Read Facility, and GO, which terminates the parameter editing facility and resumes the main dialogue.

### **Changing Parameter Values Interactively**

If a valid parameter name is entered in response to the ==> prompt, the current value of the parameter will be displayed. If the user starts to enter a new value, the current displayed value will disappear, and the new value being typed will appear. If, on the other hand, an editing key (such as HOME) is the first key pressed, the existing parameter value will be retained and can be modified using the editing keys. The ENTER key is used to indicate that changes are complete. Recognized editing keys and their meanings follow.

HOME	Move cursor to leftmost position.
END	Move cursor to rightmost position.
LEFT	Move cursor one position to left.
RIGHT	Move cursor one position to right.
DEL ARROW	Delete character to left of cursor.
DEL	Delete character under cursor.
INS	Toggle between insert and overstrike mode.
ESC	Discard changes to values, and restart with original value.

If the parameter being edited is a vector, look-up table, or array, the dialogue is slightly different. If the dimension can be changed, then a prompt allowing such change is issued. Next, if the parameter is an array, the column to be modified is requested. Finally, a prompt is issued for each element of the parameter in turn. All the editing keys listed above can be used.

### **Multi-Line Strings**

If the parameter being edited is a long string, the editing facility will generate a prompt indicating which line of the string is being displayed. The following additional editing keys then become available. As before, the ENTER key is used to indicate that changes are complete.

UP Go to previous line of string.  
DOWN Go to next line of string.  
CTRL-x Split line at cursor.  
PGUP Go to first line of parameter.  
PGDN Go to last line of parameter.

### **Changing Parameter Values Using Include Files**

If the READ command is issued, the user will be prompted for the name of a file containing values for one or more parameters. Such a file (called an include file) would typically have been created by modifying another parameter file using a text editor. The named include file will be read, and the values for the parameters given in the file will replace the corresponding current parameter values. The READ command is useful to manipulate small groups of parameters, and can be used to ‘mix and match’ elements of tax/transfer scenarios. It is the recommended method of changing parameters in Classic SPSM.

Though you can include sources and update factors, they are not necessary. An example of the content of a control parameter include file (test.cpi) might be:

```
T0FLAG 0  
T3FLAG 1
```

If the include file name specified to the READ command lacks a file extension, default file extensions will be generated depending on the type of parameters being modified. Specifically, control parameter include files have a default extension of .cpi, database adjustment of .api, and tax/transfer of .mpi. We suggest users follow the same convention for organizing parameter include files.

### **Configuring your machine to write include files using Notepad**

We encourage SPSM users to use include files in order to organize their work. In order to write these files, you’ll need a text editor. Windows comes with a basic text editor called Notepad. You may notice that when using Notepad a .txt extension is automatically added to your include files (\*.api; \*.mpi; \*.cpi).

Should you encounter this problem, go to Windows Explorer. On **Windows 2000/XP** machines, open Tools/Folder Options/File Types/New. In the “File Extension” field, type in “CPI” and select “Ok”. Repeat the steps for API and MPI files, and you’ll find that the automatically generated .txt extensions disappear.

### **SPSM BATCH FACILITY**

Sometimes it is desirable to be able to control the operation of SPSM in an automatic fashion, without needing a human operator to respond to prompts. The SPSM batch facility fulfills this need. Two methods are provided, depending on the complexity of interaction necessary.

#### **Command Line Method**

In the command line method, which is suitable for the simulation of short dialogue

interactions, the responses are indicated in a single string on the command line used to invoke SPSM, with the pound symbol '#' being used to delimit each response from the next. (Note that MS-DOS places a limit of 128 characters on a command line). For example, the line

```
spsm $spsd/ba08#temp#N#N#N#N
```

would invoke SPSM using the control parameter file `$spsd/ba08.cpr` and create output files `temp.cpr`, `temp.tbl`, etc. The Ns indicate 'No' responses to the normal questions that SPSM asks the user. In the more complicated example

```
spsm temp##Y#SAMPLEREQ#.001#read#$spsd/detsum.cpi#go#N#N#N
```

SPSM is run using the control parameter file `temp.cpr`, with a requested sample of 0.1%. The control parameter include file `$spsd/detsum.cpi` is read during the control parameter dialogue. This file will activate the case output facility, and produce a detailed summary report of each household.

You can have multiple such commands in a single file which you can save with a `.bat` or `.cmd` extension. In the command prompt, you then just need to type in the name of this `.bat` file and multiple simulations can be run.

### Control File Method

In the control file method, SPSM is invoked on the command line with a single argument, being the name of a file containing the exact responses SPSM would expect to receive had the dialogue proceeded normally. Each line of the file corresponds to a prompt that the user would have responded to. If the file `temp.ctl` had been edited to contain the following 6 lines:

```
/spsd/ba08  
temp  
N  
N  
N  
N
```

or a single line

```
/spsd/ba08#temp#N#N#N#N
```

and if SPSM had been invoked as follows:

```
spsm temp.ctl
```

then the result would have been identical to the first example given above in the section titled Command Line Method.

### CHANGING THE LANGUAGE IN CLASSIC SPSM

The **SPSM Classic** shortcut will run SPSM in English while the **MSPS classique** shortcut will run SPSM in French. It is possible, though, to change the language by changing the `SPSMLANG` environment variable. From the command prompt, if you type in:

```
set SPSMLANG=E
```

SPSM will run in English.

```
set SPSMLANG=F
```

SPSM will run in French.

## Running SPSM using other methods

We recommend using Visual SPSM or Classic SPSM to run the spsm. However, if you wish to run SPSM using some other software you can do so. You will need to set the environment variables. The method of doing this will vary depending on your operating system and the software used.

Two environment variables must be set, SPSM and SPSD. For example, if you installed SPSM Version 16.0 in the default location, they would be set to the following values:

SPSM	C:\Program Files\StatCan\SPSDM16.0\spsm
SPSD	C:\Program Files\StatCan\SPSDM16.0\spsd

You may also wish to set the language of the SPSM by setting SPSMLANG to either E or F for English or French.

Finally, the path of the executable, assuming that you installed SPSM in its default location, will be:

```
C:\Program Files\StatCan\SPSDM16.0\spsm\win32
```

If you have questions about setting up the model to run in a different environment, please phone us at (613) 951-3774 or email us at [spsdm@statcan.gc.ca](mailto:spsdm@statcan.gc.ca).