SPSD/M

User's Guide

This guide contains a comprehensive treatment of the facilities provided by SPSD/M in the 'black box' mode. It also contains a general description of variables and parameters. Please see the *SPSD/M Introduction and Overview* for an introductory treatment of these topics

October 31, 1997

Statistics Canada Statistique Canada

Canada

Table of Contents

Introduction

This document describes in detail how to use the SPSM in `black box' mode. The reader is assumed to have read the *SPSD/M Introduction and Overview*,and to be familiar with the concepts introduced there. The use of the SPSM in `glass box' mode is presented in the *SPSD/M Programmer's Guide*.

1. Section 2 describes the use of various types of parameters employed during an SPSM run.
2. Section 3 describes variables, which contain information from the SPSD and results from the application of the tax/transfer model for individuals and families.
3. Section 4 describes the overall execution of an SPSM simulation via the User Dialogue.
4. Section 5 provides a brief description of the types of database files supplied by Statistics Canada as input to the SPSM.
5. Section 6 describes the process of database adjustment used by the model to allow the SPSD to represent different calendar years.
6. Section 7 explains the specific features that allow the comparison of two scenarios by the SPSM.

Subsequent sections discuss, in turn, each of the major SPSM facilities. The order of their presentation in this guide follows the order in which the parameters that control them appear in the SPSM control parameter files.

Appendix A contains a listing of a parameter file that the reader may wish to consult while reviewing this manual. For those users with a specific interest in the cross-tabulation facility, the section titled X-tab Facility provides a description of the SPSM cross-tabulation facility in a complete but highly concise manner. A fuller and more tutorial treatment, with worked examples, can be found in the *SPSD/M X-tab User's Guide*.

Parameters

It is through parameters that the user controls all aspects of the SPSM. A parameter is a value specified prior to the actual running of the model that controls how the model executes. It is specified in one of three distinct kinds of parameter input files read by the SPSM. It is represented in these files as an uppercase symbol followed by a value. The three kinds of parameter file are discussed in the section of this guide titled Parameter Files.

Parameters can contain many different kinds of information, ranging from a simple numeric value to a complicated multi-line string used to specify a cross-tabulation request. The different forms of parameters are discussed in the section titled Types of Parameters. Parameters can be modified by using a text editor to change the values in the appropriate parameter file. Alternatively, parameter values can be modified interactively using the parameter editing facility, described in the section titled Parameter Editing Facility. The definitive source on each SPSM parameter is the *SPSD/M Parameter Guide*, which can be access in alphabetic order by clicking on Indices and then on Parameter. A useful utility (compparm) that reports on differences between parameter files is documented in the

SPSD/M Tools User's Guide.

Parameter Files

The parameters in a parameter file are ordered and formatted in a fixed way. Parameter files that do not follow these formatting conventions will be re-formatted to conform, if a new parameter file is output by SPSM.

A typical section of a parameter file might appear as follows:

```
###
## 2.3.2 Government Transfers
###


###
## 2.3.2.1 Unemployment Insurance
###

UIFLAG              1        # UI/EI Activation flag
                             # Source: Permanent Program

UIERNMAX       710.00        # Maximum insurable earnings
                             # Source: Employers' Guide to Payroll Deductions, 1992
                             # Update: Factor=UIR

UIMINPCT      0.20000        # Exemptible limit (percent of maximum insurable earnings)
                             # Source: Employers' Guide to Payroll Deductions, 1992

UIPF          0.03000        # UI contribution rate on earnings
                             # Source: Employers' Guide to Payroll Deductions, 1992

UIEIREF          0.00        # EI contribution refund cut-in
                             # Source: Not in effect
                             # Update: Factor=NONE
```

The features to note are:

1. Parameters are arranged in numbered subject matter sections. Each such section is preceded by a block comment that labels the section. Each line of the block comment starts with `##'. The number in the block comment identifies the section of the *SPSD/M Parameter Guide* in which the parameter is discussed. This block comment is automatically generated by SPSM when the parameter file is output.

2. Parameter names start in column 1, and are always given in upper case.

3. Parameter values follow the name, separated by one or more spaces. The SPSM will align these values into columns when reformatting.

4. Each parameter has an identifying label, which is found on the same line as the parameter itself. The label is introduced by the `#' symbol. SPSM generates parameter labels automatically when the parameter file is output.

User-supplied comment lines are introduced by a single `#' character in column 1, followed by text. User-supplied comments must immediately precede the parameter to which they refer. If used in parameter include files (see section titled Read Facility in Parameter Editing Facility), the user-supplied comments will follow with the parameter itself and will be reproduced by SPSM in the output parameter file.

These conventions are designed to make a parameter file not just a source of information for a SPSM run, but also a legible form of documentation on the run.

Control Parameter Files

Each execution of the SPSM makes use of a single control parameter file. This file, which has the file extension `.cpr`, specifies the names of input and output files to be used in the run, and provides values that control all the SPSM facilities described in subsequent sections of this guide.

Some entries in the control parameter file are not used to communicate information to SPSM. Instead these `informational' control parameters are used to indicate certain information about the run to the user. An example is the parameter SAMPLE, which shows the fraction of the SPSD that was actually read in the run.

Database Adjustment Parameter Files

Each execution of the SPSM makes use of a single database adjustment parameter file, which has the extension .apr. The primary function of this file is to specify parameters that `age' the data to some subsequent year by providing SPSD variable-specific growth factors. The .apr file also includes parameters that specify structural features of the year in question, such as region-specific unemployment rates.

Tax/Transfer Parameter Files

Tax/transfer parameter files completely specify the operation of the tax/transfer algorithms. A particular execution of the SPSM can require 0, 1, or 2 tax/transfer parameter files. The number of tax/transfer parameter files depends on the values given to the BASMETH and VARMETH control parameters.

Types of Parameters

Parameters supply many different kinds of information to the SPSM. There are, accordingly, several different kinds of parameters, each suited to a particular kind of information. String parameters supply textual information to the SPSM. Textual information can be as simple as a label describing a parameter file, or as complicated as a user-supplied cross-tabulation request. Scalar parameters supply single numbers, such as the amount of Family Allowance per child. Vector parameters supply a one-dimensional set of numbers. An example is the set of income break points used to define the columns of built-in Table 2. Look-up Table parameters provide information on piece-wise linear functions used in the tax/transfer system. An example is the federal tax table. Array parameters provide two-dimensional sets of numbers. An example is the set of provincial effective sales tax rates by expenditure category and province. Each of these types of parameters is discussed in the following sections.

Strings

String parameters are used to provide textual (as opposed to numeric) information to the SPSM. Each string parameter has a maximum admissible length. If the maximum length is exceeded, an appropriate error message is generated and the parameter value is truncated.

If the string of textual information is short, it is supplied on the same line as the corresponding parameter name. The following example is extracted from the supplied parameter file `\spsd\ba92t.cpr`.

```
INPSPD       $SPSD/v60y92t.spd    # Name of SPSD file (in)
```

In this example, the INPSPD parameter is specifying the exact location of the main SPSD binary file, /spsd/v60y92t.spd. The portion of the line starting with the `#' character is a comment that is automatically generated by the SPSM to make the control parameter file more comprehensible. The user should not attempt to insert their own comment on the same line as the parameter value, because the SPSM removes such comments and replaces them with parameter labels when the parameter file is written. Please see the section titled Parameter Comments for a discussion of user-supplied comments in parameter files.

The string parameter can be broken up over several lines if it is too long, or if the user want a cleaner presentation. This is illustrated by the following example.

```
XTSPEC                          # X-tab specification
IN: {ex5,
     ex7:S=3,
     ex8:S=3}
     * cl0+;
IN: cl1+ * {ex5, ex5/ex7:L="Average Benefits", ex7:S=3, ex8:S=3};
IN: cl2+ * {ex5, ex7:S=3, ex8:S=3, scfrecs };
```

In this example, the XTSPEC parameter, which is used to generate user-defined cross tabulations, has been set to a string with seven lines. The first line, whose contents are given on the same line as the parameter name, is empty. The following SPSM supplied label, introduced by the `#' character, is not considered part of the parameter value. XTSPEC is continued onto the next six lines, the first four are part of the same table request and the user align the variables to ease its future use in editing the file and improve the readability of the file. The last two lines show request on the same line. The last request can finish be either a semi-column or not. It is suggested to always use the semi-column in the last table request.

Scalars

Scalar parameters are used to provide single numeric values to the SPSM. They are given on the same line as the parameter value, as indicated in the following example.

```
STDFA          418.56        # Standard federal family allowance per child
```

In this example, taken from the tax/transfer parameter file `\spsd\ba92.mpr`, the amount of Family Allowance per child has been set to the amount indicated. Scalar values, like all numeric parameters, are re-formatted when the parameter file is written.

Vectors

Vector parameters are used to provide an ordered set of numeric values. The number of values is indicated on the same line as the parameter name. The actual values of the parameter are indicated one per line on the immediately following lines. The following example illustrates these points.

```
DISTP         13                 # Breakpoints for histogram plot
         1
         5
        10
        20
        30
        40
        50
        60
        70
        80
        90
        95
        99
```

In this example (taken from the control parameter file `\spsd\ba88t.cpr`) the vector parameter DISTP is assigned a set of 13 values, ranging from 1 to 99. This parameter specifies the percentile break points that define the horizontal axis of the histogram plot produced by the distributional analysis facility (see the section titled Distributional Analysis Facility).

Look-up Tables

Look-up table parameters are used to provide information defining piecewise linear functions. Such a function is indicated in the SPSM by a set of ordered 3-tuples. Each 3-tuple corresponds to a particular `corner' or turning point in the function. The number of 3-tuples is indicated on the same line as the parameter name, while the 3-tuples follow one per line on succeeding lines.

- The first number in each 3-tuple gives the X value for the function.
- The second number gives the corresponding Y value.
- The third number gives the slope of the line segment that starts at (X, Y).

There is some redundancy in this information. The Y values (except the first) can be derived from the slopes. Alternatively, the slopes can be derived from the X and Y values. These two forms of redundancy correspond to two types of look-up parameters, called `Point-Slope' and `X-Y' look-up table parameters. Both types have are implemented in SPSM, because particular look-up tables are more naturally represented in one form or the other

Consider the following example of a `Point-Slope' look-up table (taken from the file `\spsd\ba88.mpr`):

```
FTX           3                   # Federal tax table
         0        0    0.170
     27500   (4675)    0.260
     55000  (11825)    0.290
```

As the comment indicates, this is the federal tax table for 1988. The table has 3 distinct tax brackets, and the function is specified by supplying the X values and the corresponding tax rate (i.e., slope) for each bracket. The starting Y value (0 in this example) also must be indicated. The rest of the Y values are computed from the X values and the slopes, and are indicated inside brackets to emphasize their derived character. These derived Y values are computed by the SPSM and output when the parameter file is written. The user must supply a value for all three columns of a Point-Slope lookup parameter, but the Y values need not be accurate (except the first one), since they will be correctly computed by the SPSM when the parameter is read.

The following is an example of a `X-Y' look-up table.

```
SPAT          3                   # SPA take-up rate by benefit level
         0    0.855   (0.0000)
       577    0.870   (0.0000)
      4401    1.000   (0.0000)
```

The SPAT parameter gives the proportion of eligible persons who partake of the program, as a function of the level of benefit. For example, 87% of persons eligible for SPA will participate in the program if their benefit amount is $577. The third column gives the slope of the corresponding piece-wise linear function. The slope values are too small to be accurately represented in this example. In any case, the slope values are computed accurately by SPSM and are presented in the parameter file for informational purposes only.

Arrays

Arrays are two-dimensional sets of numbers. An example is the parameter PTF found in the data base adjustment parameter file `\spsd\ba92.apr`. Its value in that file is reproduced below.

```
PTF        9            # Low income cut-off
   16186  13883  13787  12829  11186
   20233  17354  17234  16036  13982
   25163  21583  21433  19943  17390
   30460  26126  25945  24142  21050
   34049  29205  29002  26986  23531
   37638  32284  32059  29830  26012
   41227  35363  35116  32674  28493
   41227  35363  35116  32674  28493
   41227  35363  35116  32674  28493
```

The indexes used to determine a value in the table always start at zero The PTF parameter is a two dimensional array giving a user-supplied "income threshold" for families: the row are the number of persons minus 1 – this set the index to zero for one person family - and the column the urban area (hdurb).

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

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 are on the following page.

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.

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. Several examples of its use can be found in the *SPSD/M Introduction and Overview*.

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.

Parameter Comments

Parameter files have three different kinds of comments, which have been designed to make the files be a self-documenting record of a SPSM run. Block comments, introduced by the string `##' as the first characters in a line, are automatically generated by SPSM and serve to organize the parameter file into sections. Parameter name comments are introduced by a `#' character found on the same line as the parameter name itself. These comments are also generated automatically, and help remind the reader what the sometimes-obscure parameter names stand for.

The final type of comment is introduced by a single `#' character as the first character in a line. These comments are supplied by the user and follow the parameter they refer to. They are intended to be used to document the source or reason for a particular parameter value. These user-supplied parameters follow along with their associated parameters when used in parameter include files (See the section titled Read Facility). Because of this the resulting output parameter file will retain any user-supplied comments associated with the parameter values. The tax/transfer parameter files supplied with the SPSD/M include parameters that document the source and growth method for each parameter.

User-supplied comments must be entered directly into parameter files using a text editor. The parameter editing facility does not allow any comments to be entered. It will, however, indicate that a parameter value was changed by inserting the comment

```
# Note: The following parameter was modified interactively.
```

before any database adjustment or tax/transfer parameter which was modified.

Environment Variables

Environment variables can be used to generalize the names of various input files used by SPSM. For example, the `.cpr` files in this release contain entries such as the following:

```
INPSPD    $SPSD/v60y92.spd
```

If the environment variable SPSD has been set (using the DOS SET command or in WINDOWS-SYSTEM/environment) to d:/spsd, then the preceding is equivalent to

```
INPSPD    d:/spsd/v60y92.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. As a special case, an $SPSD entry such as that given in the above example will be expanded to /spsd. This ensures that the versions of `.cpr` files in this release will function even if the SPSD environment variable has not been set. Users are advised to set the SPSD environment variable to an appropriate value.

If a (DOS or WINDOWS) environment variable named SAMPLEREQ is created, its value will be used in place of the SAMPLEREQ parameter in the `.cpr` input file. This may be useful when performing small sample test runs using controlling DOS .bat files. Instead of changing the input `.cpr` files to change the SAMPLEREQ parameter value, a single change to the (DOS or WINDOWS) environment variable SAMPLEREQ will result in the analysis being performed on the specified sub-sample. In WINDOWS it may be simpler to specify the parameter interactively or in a .cpi file.

Variables

This section contains reference information on SPSD/M variables. A variable contains information on a particular household, individual, or family in SPSD/M. This is distinct from a parameter, which generally contains data used to specify the tax/transfer system used in a simulation. Definitive information on each of the SPSD/M variables can be found in the *SPSD/M Variable Guide*which can be access in alphabetic order by clicking on Indices and then on Variable.

It is possible to use SPSD/M without referring to variables at all. For example, the parameters of a tax/transfer simulation could be input to SPSM, and standard built-in reports used to analyze results. However, many more powerful facilities of SPSD/M require the use of variables. These facilities, which are described in more detail elsewhere in this manual, include record selection, cross-tabulation, SAS output, case reporting, and distributional analysis. There is also a facility that allows the user to create new variables based on the values of existing variables.

Variables have several characteristics that must be understood for the discussions in the following sections to be comprehensible. These characteristics are described in the following sections.

Class versus Analysis

Variables in the SPSD/M can be grouped into two broad categories. Variables whose values denote membership in distinct categories are termed `class variables' in this documentation. They are also referred to as categorical or classificatory variables. An example of a class variable is idsex, which records the sex of an individual. Variables whose values hold numeric values on which arithmetic operations can be meaningfully performed are termed analysis variables. An example of an analysis variable is idiemp, which records the employment income of an individual.

The distinction between class and analysis variables is an important one. Certain of the SPSM facilities require that either a class variable or an analysis variable be specified in certain contexts. Class variables can be converted to analysis variables, or analysis variables to class variables, using the user-specified variable facilities.

One variable, age, is often required to be either a class or an analysis variable. Accordingly, two variables for age have been defined, idage for the class version and idnage for the analysis version. idage might be used to generate a report on the distribution by age of a certain group of people, while idnage might be used to generate a report on the mean age of persons in various groups.

Family Level

The SPSD is organized hierarchically, in the sense that individuals retain their family context. This means that the SPSD can be considered to be a file of individuals, a file of families (variously defined), or a file of households. Each SPSD/M variable is defined at a natural level in this family hierarchy. For example, the variable hdprov (province of residence) is defined at the level of household, whereas idiemp (employment income) is defined at the level of individual. The various SPSM reporting facilities allow the user to specify the family level of analysis desired for a particular output function. However, the user is not restricted to using variables whose natural level is the same as that specified. A number of rules serve to interpret the meaning of such `cross-level' requests.

If the natural level of the variable is `higher' than the level specified by the user, the value of the variable is the value found at that higher level. For example, a reference to hdprov when specifying the level of analysis as `individual' simply refers to the province associated with the household containing the individual.

If the natural level of the variable is `lower' than the level specified by the user, two cases arise, depending on whether the variable is an analysis variable or a class variable. If the variable is an analysis variable, then the value of the variable is the sum of the `lower' level values contained in the family unit at the specified level. For example, a reference to idiemp when specifying the level of analysis as `household' refers to the sum of employment income of all persons in the household. If the variable is a class variable, then the value of the variable is the value associated with the first contained unit (called the reference family or individual) at the variable's natural level. For example, a reference to idage when specifying the level of analysis as `household' refers to the age of the first person in the household.

Using class variables in this way clearly requires knowledge of the order in which individuals and families are arranged. Individuals are arranged within families with the eldest spouse first, followed by the other spouse (if present), followed by children in order of increasing age. Families are arranged within households with the `primary' family coming first. Boarders, for example, form `secondary' families containing only a single individual.

An added complication arises when selection has been activated using the SELFLAG, SELUNIT, and SELSPEC control parameters. Selection ultimately occurs at the level of the individual, so the following remarks apply to variables whose natural level is `individual'. If such a variable is referenced when specifying a higher level of analysis, only selected individuals are processed. For example, if the selection facility has been set up in such a way as to select only persons whose age falls between 18 and 55, then a reference to idiemp when specifying the level of analysis as `household' refers to the sum of employment income of all persons aged 18 to 55 in the household. Similarly, a reference to idage when specifying the level of analysis as `household' refers to the age of the first person in the household whose age is between 18 and 55.

Database versus Modeled

SPSD/M variables can be divided into two groups based upon whether they are read (or directly derived) from the database, or whether they result from a modeling process. The first two letters of the variable's name will indicate whether the variable is modeled or not. As indicated in the section titled Variable Naming Conventions below, the prefixes im, ub, and ct indicate modeled variables. All other prefixes indicate database variables.

Variant versus Base

The SPSM allows access to two distinct sets of modeled variables, termed `base' values and `variant' values. If the user is performing a single simulation, variant values for all variables are defined. If the user is performing two simulations simultaneously, then values for all base variables are defined. If the user is reading in base values from a previously produced results file, then base values for the saved variables will be defined, and all other base variables will be set to the value 0.

To refer to a base variable, simply prefix the name of the corresponding variable with an underscore symbol. For example, immicons refers to variant consumable income, while _immicons refers to base consumable income.

Variable Naming Conventions

Unlike parameters, which are always represented in upper case, SPSD/M variables are always given in lower case. With only a few special exceptions, variables follow a naming convention in which the first two letters of the name (the prefix) indicate the basic family level of the variable, and whether it is read from the database or produced by the model. The remaining letters of the name (the stem) describe the variable itself. Unemployment Insurance claim variables have a numeric digit inserted between the prefix and the stem, indicating which claim the variable refers to. A table of valid prefixes and their meanings is given below.

hh	Household structure data
hd	Household characteristics
ef	Economic family characteristics
cf	Census family characteristics
nf	Nuclear family characteristics
id	Individual data, from database
im	Individual data, from model
uc	UI claim data, from database
ub	UI claim data, from model
fx	Expenditure pattern data, from database
ct	Commodity tax data, from model

Expressions

The capabilities of SPSM expressions, which are used in a number of control parameters, have been considerably extended. Instead of describing the changes, the syntax of SPSM expressions is presented below in toto.

Expressions are used in a number of SPSM facilities. Specifically, expressions can be found in the parameters SELSPEC, MARSPEC, TPSPEC, UVAR, and XTSPEC. An SPSM expression is a syntactically valid sequence of constants, variables, operators, and functions that evaluate to a floating-point quantity. The syntax is similar to that used in the C programming language. As in C, comments (delimited by /* and */) can be inserted as desired, and the order of operations can be changed by using the bracket characters (and). Also as in C, a value of zero stands for the logical `false' value, while any non-zero value represents the logical value `true'. Logical operators return the value 1 to represent `true'.

Constants

Constants are decimal numbers that can contain an optional fractional part. Exponential notation is not recognized. A number of constants that are commonly used in SPSM can be entered symbolically using the upper-case synonyms shown in the following table:

Mnemonic	Value	Meaning
NFLD	0	Province codes used in the hdprov variable.
PEI	1	
NS	2	

NB	3	
QUE	4	
ONT	5	
MAN	6	
SASK	7	
ALTA	8	
BC	9	
MALE	0	Codes used in the idsex and similar variables.
FEMALE	1	
HEAD	0	Codes used in the idcfrh variable.
SPOUSE	1	
CHILD	2	

Variables

Variables consist of an optional underscore (_), followed by a lower case alphabetic character, followed optionally by additional lower case letters and numeric digits. Variables must be either SPSD database variables, SPSM modeled variables, or user variables created by the User Variable Facility. Variables are described in more detail in the *SPSD/M Variable Guide* or can be access by clicking on Indices and then on Variable.

Operators

Operators take one or two arguments, and use the following precedence hierarchy:

Operators	Description
IN: SP: NF: CF: EF: HH:	family level specifiers
- ! @	unary minus, logical negation, variant-base difference
**	exponentiation
* /	multiplication, division
+ -	addition, subtraction
< <= > >= LT LE GT GE	numeric comparison
== != EQ NE	equality, inequality
&& AND	logical conjunction
|| OR	logical disjunction

Family Level Specifiers

Expressions are evaluated at the level of the individual. Family level specifier operators allow referencing of variables pertaining to other family members. Family level specifiers immediately precede a variable or expression, referencing and cumulating the value of the variable or expression over the family unit containing the individual.

Specifier	Meaning
IN:	current individual
SP:	spouse of current individual
NF:	nuclear family containing current individual
CF:	census family containing current individual
EF:	economic family containing current individual
HH:	household containing current individual

A number of examples follow:

SP:idiemp	value of employment income of spouse
SP:(idage>70)	is 1 if spouse is over age 70 and 0 otherwise
CF:(idisa>0)	number of persons in census family receiving social assistance
CF:immicons/HH:immicons	proportion of household consumable income associated with census family containing current individual
CF:(idage>=3 && idage<=5)	number of persons in census family between the ages of 3 and 5 inclusive

Family level specifiers involving user variables require careful attention. The analyst must ensure that the user variable has been computed for all persons in the household before the family level specifier operator is applied either to it or to an expression containing it.

Variant-Base Difference Operator

The variant-base difference operator computes the difference between the variant and base values of a modeled variable. For example, the expression @immicons is synonymous with immicons-_immicons and gives the change in consumable income from base to variant.

Arithmetic Operators

The conventional arithmetic operators +, -, *, / follow conventional precedence rules and require no explanation. The symbol ** stands for exponentiation and has a high level of precedence.

Logical and Comparison Operators

The logical operators operate in a conventional manner. SPSM allows synonyms for certain of the operators, as indicated in the above table. As mentioned, any non-zero value has the logical value `true'. All of the logical and comparison operators use 1 to denote a `true' value.

Functions

Functions are specified using the expected notation, and are listed in the following table.

Function	Description
abs(a)	absolute value
nneg(a)	non-negative: equivalent to max(0,a)
min(a,b)	minimum of two values
max(a,b)	maximum of two values
ln(a)	natural logarithm
exp(a)	exponentiation
split(a,b1,b2,...)	determine which range value falls in
trunc(a)	integer part of value

The split function returns 0 if a<=b1, 1 if b1<a<=b2, etc.

Statements

The user variable facility makes use of statements, which are similar to C-language statements. The syntax of these statements is described in this section. The main purpose of statements is to assign expression values to user variables.

Assignment Statements

Assignment is accomplished using the = symbol. SPSM is deliberately more restrictive than C in the syntax of this operator, thus preventing certain kinds of errors that are very easy to make in that language. The syntax is:

```
<user variable> = <expression> ;
```

The variable being assigned cannot be a database or modeled variable. If the user variable already has a value (created either by an earlier statement or through the Reference Value Facility), the value will be replaced. A special form allows assignment to the spouse of the current individual:

```
SP:<user variable> = <expression> ;
```

Definitional Statements

A user variable has a number of characteristics in addition to the value it holds for each individual. It can have a label, and be either a classificatory or analysis type variable. If its type is analysis, it can have a fractional part, printed to a specified level of precision. If its type is classificatory, it has a fixed number of allowed levels, and each level can have an associated label used for printing. By default a user variable is an analysis variable with no fractional part (except if generated by the split function), but these characteristics can be changed using the following statements.

label statement

As its name suggests, the label statement is used to associate a textual label with a user variable. The syntax is as follows:

```
label(a) = "textual label" ;
```

In this example, the user variable a is given the indicated label. This label will be used by the various SPSM output facilities.

levels statement

The levels statement gives the number of levels for a user variable and provides a label for each level. It implicitly sets the user variable type to classificatory.

```
levels(a) =  "level #0", "level #1", "level #3" ;
```

This example sets a to a classificatory variable with three levels labeled as indicated. As with all SPSM classificatory variables, the first level has the value 0, the second has the value 1, etc.

type statement

The type statement forces the type of the user variable to either classificatory or analysis. It is generally not needed because user variables get implicitly assigned a generally appropriate type.

```
type(a) = analysis ;
type(b) = class ;
```

The preceding examples set the type of the user variable a to analysis, and b to classificatory.

precision statement

The precision statement indicates that the user variable has a fractional part, and indicates the number of digits to display to the right of the decimal point when printing values for the variable. Most SPSM variables, being dollar quantities, have no fractional part. This allows them to be stored efficiently in SPSM result files. The precision must be set to a non-zero value to preserve fractional parts in result files. This will store the variable as a single precision floating point quantity, with a total of about 6.5 digits of accuracy.

```
precision(a) = 3 ;
```

The precision of the analysis variable a has been set to 3. If a is printed, 3 digits to the right of the decimal point will be displayed.

Assignment from split function

If a variable is assigned the result of the split function, a number of implicit declarations about the variable occur. Specifically, the variable will be a classificatory variable with the appropriate number of levels, and default level labels will be generated for the variable if possible.

```
b = split(idiemp, 1000, 2000, 3000) ;
```

The preceding example provides attributes to b equivalent the following statements:

```
label(b) = "Employment income Group" ;
```

```
levels(b) = "Min-1000", "1001-2000", "2001-3000", "3001-Max" ;
```

If the first argument of split had been an expression, a label for b would not have been generated. If any of the subsequent arguments of split had been expressions, the level labels would not have been generated.

Flow Control Statements

There are three statements that are used to affect the flow of control of statement execution.

Statement group

A set of statements can be made to act as a group by surrounding the group with curly braces ({ and }). This allows groups of statements to be conditionally executed using the if and else statements.

```
{ a=1; b=2; }
```

This example groups the two assignments into a single statement.

if statement

The if statement allows conditional execution of a statement (or statement group)

```
if (idcfrh==0 && idspoflg==0 && cfnkids>0 ) benefit = 1000;
```

This example assigns heads of single parent families a benefit of 1000 dollars.

if else statement

The if else statement allows one of two statements to be executed depending on whether a condition is true or false.

```
if (imigist == 0) benefit = 500 ;
else              benefit = 100 ;
```

This example assigns a benefit of 500 dollars to persons without any GIS top-up and 100 dollars to all others. The if else statement can be used repeatedly to specify an action depending on one of a number of conditions, as in the following example:

```
if (idcfrh==0) {
  if      (cfnkids==0) benefit = 0;
  else if (cfnkids==1) benefit = 100;
  else if (cfnkids==2) benefit = 300;
  else if (cfnkids==3) benefit = 600;
  else                 benefit = 600 + 400 * (cfnkids-3);
}
```

This example assigns a benefit to the head of a census family depending on the number of kids in the family. The benefit per child increases with the parity of the child.

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 up to eight characters, followed by a dot, followed by a three character extension.

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/ba88t.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, OUTVARMRS, OUTASC, OUTSAS, 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 occur.

After these six dialogue steps are complete, SPSM will perform the run. After about 1 minute of computation, an estimate of the final completion time will be issued. 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.

Descriptive Parameters

The CPRDESC control parameter allows the user to provide a short general description of the purpose of a particular SPSM execution.

The LICENSEE control parameter is informational in nature. It always contains the name of the person or organization legally permitted to use the particular copy of the SPSD/M.

The AUTHOR control parameter allows the user to document who produced the specific SPSM run.

The control parameter ALGDESC is informational in nature. Its value is created by SPSM and included in the control parameter file for documentation purposes. It contains information about the standard and alternate algorithms available in the version of SPSM being executed. If the user follows the procedures given in the *SPSD/M Programmer's Guide* when using `glass box' mode, appropriate descriptions of tax/transfer algorithms will appear in ALGDESC.

Miscellaneous Control Features

Break Processing

A break processing feature has been provided to allow greater control of the model execution. The processing of the break key is as follows:

Setting the BRKFLAG Control Parameter

If the BRKFLAG control parameter is set to `0,` SPSM will immediately exit if the break key CTRL-C combination is pressed.

To enable break key interception, set BRKFLAG to 1. This is the default value found in the `.cpr` files. BRKFLAG should generally be left at 1. Its main use is to enable break if large amounts of DEBUG output are being generated through glass box use.

If BRKFLAG processing has been activated, the action taken by SPSM when the break key CTRL-C combination is pressed varies:

> If SPSM is engaged in user dialogue, the message "Abort SPSM?" will appear. Type "Y" to abort, or any other key to continue. This extra prompt avoids an unexpected exit (losing editing changes) if CTRL-C was pressed by mistake.
>
> It splits a multi-line parameter instead of CTRL-X.
>
> If SPSM is in the simulation phase, a message indicating the current state of the simulation will be printed and the user asked if the run should be terminated, as before. Output tables will reflect the sample obtained at the time of the break. If the run has been truncated in this way, SPSM will return a non-zero error code. You can use the error level facility in .bat files to detect when SPSM has exited in less than

perfect condition, and take appropriate action (such as terminating the .bat file). See your operating system documentation for more details.

If SPSM is in the reporting phase (i.e. writing out the table file) then SPSM will terminate immediately and the file will be truncated.

If SPSM is being run in batch mode, it terminates the run.

Audio "Beep"

The BEEPFLAG parameter, if activated by setting its value to 1, will produce a beeping sound when SPSM execution is complete. Users may find this facility convenient if operating on multiple machines, or if busy on another tasks.

BEEPFLAG: Beep on completion parameter

Histogram Generation

ETAFLAG is normally activated and cause SPSM to generate a horizontal histogram that indicates an estimate of what percentage of the requested simulation has been completed. Note that the estimate will not be particularly accurate when selecting small samples.

ETAFLAG: Activate fraction completed display parameter

Rounding Control

ROUNDFLAG is normally activated, so those consumable and disposable incomes are rounded to the nearest dollar before being used for reporting or output purposes. Because SPSM results files always contain rounded data, this action ensures that base/variant comparisons that use disposable or consumable income will be identical whether BASMETH is 1 (read results file) or 2 (create base variables through simulation). When ROUNDFLAG is set to 0, this action is turned off. This action is useful in conjunction with the turning point facility, because otherwise the rounding of consumable income produces a large number of spurious turning points in the household tax function.

Database Files

The information in the SPSD is distributed over three distinct kinds of files. Different versions of each file type are found in the installation kit. This section describes the contents of the different type of SPSD files and the corresponding control parameters.

Household/Individual File

The INPSPD control parameter specifies the name of the main SPSD input file. Files of this type have an extension of .spd, and contain information on household and family structure, demography, and income.

Expenditure File

The INPFXV control parameter specifies the name of the SPSD file that contains household expenditure data. Because not all users may be interested in this data (whose primary function is to allow the simulation of commodity taxes) this file does not necessarily need to be read. Setting the FXVFLAG control parameter to 0 disables reading of the expenditure data file.

Weight File

The INPWGT control parameter specifies the name of the SPSD file containing household weights. A number of weight files have been supplied with the SPSD/M. These files have been constructed to reproduce the estimated or projected population of Canada by age, sex, and province for a number of years.

Control of Weighting

This WGTFLAG control parameter determines whether or not the database sample used will be weighted. It is usually left at a value of 1, in which case weights are applied. If given a value of 0 (deactivated), the weight file specified in the parameter INPWGT (if any) will not be read, and all weights will instead be set to the value 1.0. This facility can be used to produce unweighted tabulations of SPSD, and is also useful if `hypothetical households' generated using the bldspd60 utility (see the *SPSD/M Tool User's Guide*) are being used instead of the supplied SPSD.

Reference Value Facility

Values of modeled variables produced in an SPSM run can be accessed in a subsequent run by using SPSM results files to create base modeled values, as described in the section Variant and Base (using OUTMRSFLAG, OUTMRSVARS, and OUTVARMRS to create the results file and BASMETH and INPBASMRS to read the results file in a subsequent run). The Reference Value Facility allows a different form of access to variables created in a previous run. The main difference is that the variables produced by the Reference Value Facility can be either user variables or database variables, rather than base modeled variables.

Three control parameters are used by the Reference Value Facility. The facility is activated by setting REFFLAG to 1. INPREF must be set to the name of a previously produced results (.mrs) file. The REFVARS parameter lists the names of the variables that the user wishes to read from the results file, and can optionally be used to re-name these variables as well.

The different kinds of operations that the Reference Value Facility can perform are described in detail in the following sections.

Creating user variables from a previous run

Perhaps the most common use of the Reference Value Facility is to access user variables produced in a previous run. To do this, the user must first create a results file containing one or more user variables by setting control parameters in a manner similar to the following example.

```
UVARFLAG       1
UVAR           \
    etr = (CF:immtot - CF:immicons) / CF:immtot; \
    label(etr) = "Effective tax rate"; \
    precision(etr) = 3;
OUTMRSFLAG     1
OUTMRSVARS     immtot immicons etr
OUTVARMRS      refvals.mrs
```

In this example, a user variable named etr is created. This variable contains the effective tax rate on Census Family income resulting from a particular SPSM run. This variable, as well as the modeled total and consumable income, is written to a results file named refvals.mrs.

To access the etr variable in a subsequent run, the following control parameters would be used:

```
REFFLAG       1
INPREF        refvals.mrs
REFVARS       etr
```

This causes the etr variable to be retrieved from the refvals.mrs file. The etr variable can then be used by any of the SPSM reporting or output facilities. The etr variable could be renamed to refetr by specifying instead:

```
REFFLAG       1
INPREF        refvals.mrs
REFVARS       refetr=etr
```

This would allow the effective tax rate for the second run to be created and named etr (using the UVAR parameter) without conflict. In the following example, the values of the modeled variables immtot and immicons in the refvals.mrs file are converted into user variables as well:

```
REFFLAG       1
INPREF        refvals.mrs
REFVARS       refetr=etr reftot=immtot refcons=immicons
```

Modeled variables must be renamed to user variables if they are mentioned in REFVARS, as illustrated in the previous example. It should also be noted that user variables created using the Reference Value Facility can be overwritten by the User Variable Facility.

Values of variables from any number of SPSM runs can be combined into a single results file by reading in variables using the Reference Value Facility, and then writing out these variables, together with new ones from the current run, to the OUTVARMRS results file. When using this technique, note that the OUTVARMRS file cannot be named identically to either INPBASMRS or INPREF.

Modifying database variables

The reference value facility can be used to replace the values of SPSD variables. This is illustrated by the following simplified hypothetical example, in which a user attempts to model a certain kind of behavioural response to a decrease in UI income. In the first run, base variables are assumed to exist, and variant values come from a scenario in which UI benefits have been affected by parameter changes. Assume the user wishes to perform a subsequent run in which persons respond to a shortfall of UI benefits by increasing their employment income. To do this, the user would produce a result file from the first run using the following parameters:

```
UVARFLAG      1
UVAR          \
  newemp = idiemp + nneg(_imiuib - imiuib); \
  label(newemp) = "Employment income (after response)";
OUTMRSFLAG    1
OUTVARMRS     run1.mrs
OUTMSRVARS    newemp
```

The newemp variable contains the employment income after behavioural response (it has been assumed that an increase in UI benefits will not decrease employment income). In a subsequent run, the database value of employment income can be replaced by the employment income after behavioural response by using the following parameters:

```
REFFLAG       1
INPREF        run1.mrs
REFVARS       idiemp=newemp
```

Note that database adjustment occurs before the Reference Value Facility replaces any database variables. In other words, replaced database variables are not grown after being read from the INPREF file.

The ideas in this example can be combined with the Text File Output Facility and the database build utilities to create modified SPSD database files. For example, a new user variable could be created based on SPSD/M variables using the User Variable Facility and output to a results file. This variable could be read in a subsequent run and assigned to the extra database variable idext0 using REFVARS. The `/spsd/bldspd60.cpi` and `/spsd/bldspd60.api` files could be used to create an ASCII dump of the SPSD (including the new value in idext0) which would then be input to the bldspd60 utility to create a new .spd file which permanently incorporates idext0.

Database Adjustment

Database adjustment is a distinct phase of calculations in SPSM. It is a process through which SPSD variables can be modified to reflect changes that have occurred between the base year of the SPSD data and the year of interest for analysis purposes. The user may wish to modify SPSD variables for other purposes as well. The process of database adjustment is controlled by giving appropriate values to the parameters contained in the database adjustment parameter file (given in the control parameter INPAPR).

There are two distinct types of parameters in the database adjustment parameter file. The first type is structural parameters, which attempt to represent qualitative changes to specific variables. The second type are growth factors, which are used to scale up income variables to account for inflation and/or economic growth or to adjust for under-reporting of expenditure items. Please see the *SPSD/M Parameter Guide*for more information on database adjustment parameters.

Imputation Control

In order to correct for known under reporting problems on the SCF a series of variables have been added. The variables apply to Unemployment Insurance Benefits, Social Assistance Benefits, CPP/QPP benefits, and interest income. For each of these income items two possible variables are available:

1. No Conversion	Use original SCF reported Values
2. Rank Conversion:	Use Values Generated through rank conversion procedure

Please see the *SPSD/M Database Creation Guide*for more details on the actual conversion procedures. The user has control over the selection of variable S through three database adjustment parameters (.apr). The parameters are listed below along with their default settings.

```
IMPUIBOPT          2            # Imputation method, UI
                                # Source: Use rank conversion

IMPSAOPT           2            # Imputation method, SA
                                # Source: Use rank conversion

IMPINTOPT          2            # Imputation method,Interest
                                # Source: Activate conversion
```

Variant and Base

In the terminology used in SPSM, `variant' refers to the set of tax/transfer parameters, algorithms, and variables that corresponds to one of two possible simultaneous model calculations. The other possible set of calculations is described using the term `base'. The results of primary interest to the user will normally be variant values. Base values are normally used only for comparative purposes. This section describes the control parameters used to specify how base and variant values are produced.

Variant Control

The control parameter VARMETH provides information about the algorithm used to produce variant values for variables. Its values and their meanings are given in the table below.

VARMETH	Interpretation
0	No variant variable values.
1	Not used.
2	Variant variables produced using standard algorithm and tax/transfer parameter file INPVARMPR.
3	Variant variables produced using alternate algorithm and tax/transfer parameter file INPVARMPR.

The parameters used to generate variant values are given in a file whose name is provided in the control parameter INPVARMPR. This name may be a fully specified file name, such as `c:\spsd\ba92.mpr`, or it can consist just of the filename and extension, e.g. `var1.mpr`. In the latter case, the file is assumed to exist in the current directory. In the course of SPSM's interaction with the user, it is possible to change values of variant parameters. If such changes are made, SPSM will write out a parameter file containing the changes to the file whose name is specified in the control parameter OUTVARMPR. This allows one file (INPVARMPR) to be used as the starting point for variant parameter values, with another file (OUTVARMPR) containing the actual values that were used to produce variant variable values.

The VARALG control parameter is informational in nature. It gives the overall name of the algorithm (either standard or alternate as specified in VARMETH).

It is possible to preserve the variant variable results for use in future SPSM executions. This can result in substantial timesaving if the same results are used repeatedly in subsequent SPSM executions. Three control parameters must be specified to generate a results file. OUTMRSFLAG must be set to the value 1 to activate the facility for saving variant variables. OUTVARMRS must be set to the name of the file to contain the results (this file should have an extension of .mrs). Finally, the OUTMRSVARS control parameter must contain a list of variables to be output. The variable names in OUTMRSVARS are separated by spaces. User variables can be listed in OUTMRSVARS and will be output to the results file. These variables can then be retrieved in a subsequent run using the Reference Value Facility.

As with all string parameters, long values can be continued onto subsequent lines (See section titled Strings). It should be noted that each of the variables in OUTMRSVARS will have its value rounded to the nearest integer before being written to the results file. Note also that if selection has been activated (see section titled Selection Facility), unselected households will have zero values for all variables in the results file. The section titled Selection Facility describes how a results file can be used to supply base variable values in a subsequent SPSM execution.

The bldmrs60.exe utility can construct an SPSM results file from an ASCII file. This utility and a related new utility, spsdinfo, are described in the *SPSD/M Tools User's Guide*.

Base Control

The BASMETH control parameter determines which algorithm is used to generate base variable values. Note that, as distinct from VARMETH, the value 1 allows the user to use a previously produced results file to generate base variable values. This process is described in more detail below. Valid values for BASMETH and their meanings are given in the table below.

BASMETH	Interpretation
0	No BASE variable values.
1	Base variables come from existing results file INPBASMRS.
2	Base variables produced using standard algorithm and tax/transfer parameter file INPBASMPR.
3	Base variables produced using alternate algorithm and tax/transfer parameter file INPBASMPR.

The BASALG parameter, like the VARALG parameter, is informational in nature. It contains a string describing the overall algorithm used to produce base variable values.

If BASMETH has been set to 1 then a value for INPBASMRS must be specified. INPBASMRS gives the name of the file containing values to be used for base variables. The informational control parameter INPMRSVARS will contain a list of the variables read from INPBASMRS. The base values of any other variables will have the value zero.

If BASMETH has been set to 2 or 3 then a value for INPBASMPR must be specified. INPBASMPR gives the name of the file containing the tax/transfer parameters to be used to generate base variable values. SPSM does not provide a facility to modify these parameters interactively. If necessary, a text editor can be used for this purpose.

Sub-sampling Facility

The SPSM can take a considerable amount of time to compute results for the entire SPSD database, particularly if computationally intensive facilities have been activated. In order to quickly assess the validity of a run, or to explore the broad effects of parameter changes, SPSM incorporates a sub-sampling facility.

Sub-sampling can be controlled in two ways. In the first method, the user sets the control parameter SAMPLEREQ to the sub-sample desired. SPSM will then read and process the SPSD until the desired sub-sample has been attained. In the second method, the user presses CTRL-C to interrupt the run after some fraction of SPSD has been processed. In either case, the sub-sample actually obtained is recorded in the SAMPLE informational control parameter, and output tables are scaled by the reciprocal of this value.

Reading of the SPSD always proceeds sequentially, but a relatively unbiased sub-sample is nevertheless obtained because households in SPSD are by and large randomly ordered. The ordering is not completely random, however. In order to improve the accuracy of results obtained when using sub-sampling, SPSD has been arranged as a running stratified sample. The strata used are province and household income, and the sub-sampling stratification samples occur for SAMPLE values of 0.05, 0.10, and 0.25. Samples of these sizes have been arranged to have representative provincial and household income distributions. The demonstration version of SPSD is identical to the 5% stratified sample.

The sub-sampling facility operates independently of the selection facility (see section titled Selection Facility). This means that for a given value of SAMPLE, the same households are read and processed, irrespective of selection. If the user wishes to explicitly sample a particular group of interest, a random number stream can be activated and the corresponding random number can be used in the SELSPEC expression (see section titled Random Number Facility for more information on SPSM random numbers).

Selection Facility

The SPSM selection facility provides a means to restrict the set of individuals and families that are processed by the SPSM output and reporting facilities. Selection satisfies two distinct needs. Firstly, it allows the analyst to focus attention on a particular sub-group of interest. Secondly, it can allow SPSM to execute faster by restricting the number of households processed to those of interest.

The SELFLAG control parameter must be set to 1 to activate the selection facility. When set to 0, the facility is deactivated, but the other parameters that control the facility (SELSPEC and SELUNIT) retain their values. Thus by changing the value of SELFLAG, the user can temporarily turn off a complicated selection specification, and later re-activate it easily.

The parameter SELSPEC is a string containing an expression (see section titled Expressions for a description of SPSM expressions) which is evaluated for each individual in the household. If the result of the evaluation is non-zero, the individual is considered to be selected. If an individual is selected, then everyone in the same family unit (as indicated by SELUNIT) is also selected, irrespective of the value of SELSPEC for those other individuals. In other words, if one or more persons in the family unit (as indicated by SELUNIT) are selected, the entire family unit is selected. Note that if SELUNIT is 0 then each individual is selected based on the value of SELSPEC for that individual. In this case no propagation to other family members occurs.

The values of SELUNIT and their meanings are given in the following table. SELUNIT controls the type of units that are to be selected, but does not influence the level of analysis used to calculate SELSPEC for each individual. The level of analysis of variables in the SELSPEC expression is always the level of the individual person. Note, however, that individual variables in the SELSPEC expression can be overridden to a higher level of analysis by using appropriate family level prefixes. For example, the variable idiemp in SELSPEC refers to the employment income of the individual, whereas CF: idiemp refers to the total employment income of the Census Family in which the individual is found.

SELUNIT	Level of selection propagation
0	Individual (no propagation)
1	Nuclear Family
2	Census Family
3	Economic Family
4	Household

Table 1: Interpretation of SELUNIT values

The combination of selection propagation using SELUNIT, general expressions using SELSPEC, and level of analysis prefixes within SELSPEC create a powerful and general selection facility. The following examples illustrate how these features operate.

Eg#	SELUNIT	SELSPEC	Explanation
1	0	hdprov == 0	Individuals in Newfoundland.
2	4	hdprov == 0	Households with individuals in Newfoundland (more efficient than previous example).
3	3	idage >= 65	Economic Families containing one or more elderly persons.
4	3	idage >= 65 && idiemp > 0	Economic Families containing elderly with employment income.
5	0	idiemp > 10000	Individuals with over $10,000 in employment income.
6	2	idiemp > 10000	Census Families containing one or more individuals with over $10,000 in employment income.
7	2	CF:idiemp > 10000	Census Families with over $10,000 in employment income.
8	0	imiuib > 0 && (EF:idiemp / efnpers) > 10000	Individuals receiving UI in Economic Families whose employment income per person exceeds $10,000.
9	2	(immdisp - _immdisp) > 1000	Census Families whose disposable income increased by $1,000 or more between the base and variant.

Table 2: Selection Facility Examples

The selection facility operates essentially at the level of the individual. The end result of selection is that individuals within a household are marked as selected or not. However, the SPSM output and reporting facilities operate at user-specified family levels of analysis. The following rule indicates whether or not a family unit is considered to be an observation for reporting purposes. If any individual within a family unit is marked as selected, the family unit is considered to be an observation for reporting. Conversely, if no individual within a family unit is marked as selected, the family unit is not reported.

The user must be careful when SELUNIT is less than the level of analysis specified in one of the SPSM output or reporting facilities. In such a case the family units being reported can be partial units. For example, if SASUNIT is set to 2, SASVARS is set to idiemp, and the selection facility parameters are as indicated in example 5 in the above table, then each record in the resulting SAS file will correspond to a Census Family and will contain the employment income of all members of the Census Family whose employment income exceeded $10,000.

To compute taxes and transfers correctly, SPSM always simulates entire households. Because of this, selection has no effect on the values of any variables, modeled or database, at the individual level. If, however, an examination of the database variables in the SELSPEC expression indicates that no individual in the household could possibly be selected, (irrespective of the values of any modeled variables in SELSPEC), then SPSM skips immediately to the next household. This can result in a considerable decrease in SPSM execution time. It is nevertheless necessary to compute any activated random number streams to assure that their values will be reproducible from run to run irrespective of selection.

Marginal Tax Rate Facility

The marginal tax rate is defined as the proportion of an extra dollar of income that is taxed. It is a useful concept because it measures the extent to which incentives to obtain additional incomes are reduced by the tax/transfer system. SPSM provides a facility to compute marginal tax rates. When MARFLAG is set to 1, the facility is activated and the tax/transfer system is applied twice to each household: once to the original incomes and once to the incremented incomes, and the resulting change in consumable income is noted.

The definition of marginal tax rate given above is not totally complete. To calculate a marginal tax rate, one must in addition specify the source of income being incremented, the amount of the increment to income, and which individuals are to receive the increment. The change in consumable income at the individual level can then be aggregated to produce marginal tax rates at different family levels of analysis. Each of these issues is discussed in turn below.

Marginal tax rates vary by source of income because the tax/transfer system treats income differently by source. For example, an additional dollar of dividend income is taxed differently than an additional dollar of employment income. The MARVAR control parameter gives the name of the income database variable to be incremented to compute marginal tax rates. Continuing the example above, if MARVAR contains the value idiemp, employment income is incremented, whereas if it contains the value ididiv, dividend income is incremented.

If the amount of increment to income is made small, then an accurate measurement of the local slope of the tax/transfer function (equivalent to the marginal tax rate) can be made. This is not always desirable, however. The tax/transfer function may change rapidly over small ranges of income. For example, at the income at which the Guaranteed Income Supplement is reduced to zero (the break-even income), there is an instantaneous change in the marginal tax rate of 50%. Also, it is not clear that individuals have the ability (or the inclination) to change their incomes by small amounts in response to local fluctuations in the marginal tax rate. The parameter MARAMT controls the amount of the increment to income used to compute marginal tax rates.

When computing marginal tax rates for families, it is necessary to specify not only the source and amount of income to be incremented, but also which individuals have their income incremented. For example, when evaluating the marginal tax rate on employment income it would not be desirable to increment the employment income of non-employable individuals (e.g. children and elderly). The MARSPEC parameter identifies which individuals are to receive the increment. If the value of the MARSPEC expression is non-zero for an individual, then the individual receives an increment of MARAMT for the purposes of marginal tax rate calculation. In the above example, setting MARSPEC to idage>17 && idage<65 would restrict the incrementing operation to the desired subset of individuals.

Marginal tax rates differ depending on the family level of analysis. Consider a married couple, where one person has zero income. If $500 is given to the individual with no income, that person's marginal tax rate would be zero. The consumable income of the person's spouse would decrease however, due to a reduction in the married tax credit/deduction. Hence the marginal tax rate of the spouse would compute to an infinite value, since the spouse's taxes have increased, even though he/she received no additional income. Considered as a family, however, the marginal tax rate would evaluate to a reasonable value.

To allow the computation of marginal tax rates at different family levels of analysis, SPSM assigns values to the individual level variables immaramt and immartax when the marginal tax rate facility is activated. immaramt records the amount of income increment received by the individual. Its value is equal to MARAMT if MARSPEC evaluated to a non-zero value, otherwise its value is zero. immartax records the difference between immaramt and the change in the individual's consumable income as a result of income incrementation. It represents the amount of `tax' collected from the individual as a result of income incrementation. Note that setting the tax/transfer parameter CTFLAG to 0 deactivates the calculation of commodity taxes, making the marginal tax rate calculations occur on disposable rather than consumable income.

Using immartax and immaramt, the user can calculate marginal tax rates at various family levels of analysis through the user-defined variable facility (see the section titled User-Defined Variable Facility). For example, the expression immartax/immaramt will calculate the marginal tax rate.

User-defined Variable Facility

The SPSM has two distinct modes of use, termed `black box' and `glass box'. The `glass box' mode provides the user with considerable flexibility to design new algorithms and create new variables, but it can be somewhat complicated to use, and requires some knowledge of programming.  The user-defined variable facility allows the user to create new reporting variables in `black box' mode. It allows the user to perform many analyses that would otherwise require programming changes to SPSM. Below is a description of the User-defined Variable Facility.

User Variable Facility

The User Variable Facility allows the creation of new user-defined variables, whose values are created through SPSM statements. User variables are defined at the individual level, although expressions can refer to constructs at higher levels of analysis. The UVARFLAG parameter must be set to 1 to activate the facility. The UVAR parameter contains a list of SPSM statements that create and assign the user variables. Up to 75 new user variables can be created. Note that assignments in UVAR can replace the values of identically named user variables created by the Reference Value Facility.

In the following example, the user wishes to create a new weight file in which the population of Newfoundland has been increased by 3% over its base value. Consider the following set of control parameters:

```
VARMETH       0
SEED          1
         42
UVARFLAG      1
UVAR
    if (hdprov==NFLD) {
      fltwgt = 1.03 * hdwgthh;
      intwgt = trunc(fltwgt);
      if (idrand0 < (fltwgt-intwgt)) intwgt = intwgt + 1;
    }
    else intwgt = hdwgthh;
ASCFLAG       1
ASCUNIT       4
ASCVARS       intwgt
ASCSTYLE      4
OUTASC        newwgts.prn
```

VARMETH has been set to zero to avoid unnecessary calculation of modeled variables. One of the random number streams has been activated, with an arbitrary initial seed value of 42. A number of user variables are created in UVAR and one of them (intwgt) is output to a text file for subsequent input to the bldwgt60 utility (described in the *SPSM Tools User's Guide*).

The statements in UVAR are interpreted as follows. First of all, intwgt is set to hdwgthh if the province is not Newfoundland. If the province is Newfoundland, hdwgthh is scaled up by 3%, creating the variable fltwgt. This variable is not suitable for output as is, since it contains a fractional part, whereas SPSD weights must always be integers. The fractional part of fltwgt is therefore discarded to produce intwgt. The next line adds one to intwgt probabilistically depending on the size of the fractional part: the closer the fractional part is to 1, the likelier it is that intwgt will be incremented.

UVAR consists of a list of statements. Each statement is evaluated in turn for each household member before the next statement is evaluated. This means that examples such as the following will work as expected:

```
UVAR
```

```
    income = idiemp + idisenf + idisefm;
    incrat = income / CF:income;
```

In this example, incrat is the individual's share of Census Family income. This works because the first statement is evaluated for all individuals before the second statement is evaluated. There is a computational cost associated with evaluating statements this way, because SPSM must pass through all household members for each statement in UVAR. This cost can be avoided by grouping statements in UVAR as shown in the following example:

```
UVAR
   {
     income  = idiemp + idisenf + idisefm ;
     tax     = idftax ;
     taxrate = tax / income ;
   }
   rate = taxrate / (CF:tax / CF:income);
```

In this example, UVAR contains two statements, the first one being a compound statement. income, tax, and rate can be computed without reference to other family members, so they have been grouped for efficiency into a compound statement. The calculation of rate cannot be grouped with the preceding statements because it requires that tax and income be computed for all family members beforehand. rate is the ratio of the tax rate for an individual to the overall tax rate for the individual's family.

Text Output Facility

The SPSM text output facility creates a character file containing micro-data information. There are two classes of uses to which one can put such a file. One can either print or observe it directly, or one can use it as input into some other computer package, such as a spreadsheet or database.

The facility is activated by setting ASCFLAG to a value from 1, and OUTASC to the file name to be produced. As with the other parameters containing output file names, OUTASC will be generated automatically if not specified, based on the control parameter file name (See the section titled SPSM Control).

The variables whose values are to be displayed are given in ASCVARS, which may be continued onto subsequent lines using the trailing `\' convention described in the section titled Strings. Each case that is output corresponds to the level of analysis given in ASCUNIT, as shown in the table below.

ASCUNIT	Case output family level
0	Individual
1	Nuclear Family
2	Census Family
3	Economic Family
4	Household

Table 3: Interpretation of ASCUNIT values

The number of digits of precision in the output file is controlled by the parameter ASCEXTPRC (Number of digits of extra precision). If ASCEXTPRC is left at zero, the text file output facility will output only the integer part of the variables. When ASCEXTPRC is set to a positive number, it adds the required number of decimals to numbers (not to integer variables). This extra precision is useful to accurately compute marginal tax rates when using the turning point facility. The user should be careful by grouping integer and numbers variables in ASCVARS because it may have a serious visual impact on the output file.

A number of styles of output are supported. A particular style of output is requested by setting an appropriate value to the ASCSTYLE parameter. Four different output styles are supported, corresponding to ASCSTYLE values of 1, 2, 3, 4, or 5. The format of the output for each style is more easily illustrated than described verbally. The appearance of the output for ASCUNIT set to 0 (cases are individuals) and ASCVARS set to hdprov idage idsex idiemp is shown below for each style.

An ASCSTYLE value of 1 produces a report designed to be easily human-readable. One household is output per page, and one variable is output per line. Both the variable's name and label are printed, and values for each unit are shown in aligned columns. In this example units are individuals, because ASCUNIT was set to 0. Variables that exist at the household level are shown only in the first column, since their values are known to be identical for all units in the household. The selection facility (see the section titled Selection Facility) is generally used in conjunction with this style, since otherwise very large files could result.

As a special case applying to an ASCSTYLE value of 1, if a dash character ('-') is found in ASCVARS, a separator line is generated in the report. The file detsum`.cpi` in the `/spsd` directory illustrates this feature, and contains values for the text output facility parameters that create a useful set of variables for general use. It can be activated easily by using the read function of the parameter editing facility (see the section titled Read Facility).

```
hdseqhh   Household sequence number ..........      61
idefseq   Economic family sub-sequence number        0       0       0
idcfseq   Census family sub-sequence number ..       0       0       0
hdprov    Province ...........................       6
idage     Age ................................      28      24       0
idsex     Sex ................................       0       1       1
idcfrh    Relationship to census family head ..      0       1       2
immmkt    Market income ......................   27953   17350       0
immtran   All transfer income ................     194     371       0
<page break>
hdseqhh   Household sequence number ..........      62
idefseq   Economic family sub-sequence number        0       0       0
idcfseq   Census family sub-sequence number ...      0       0       0
hdprov    Province ..........................        6
idage     Age ...............................       53      50      18
idsex     Sex ...............................        0       1       0
idcfrh    Relationship to census family head .       0       1       2
immmkt    Market income  .....................   36457    3750    4274
immtran   All transfer income ...............        0    1222       0
```

An ASCSTYLE value of 2 produces a report designed to be read using a spreadsheet import function. The layout is very similar to that for an ASCSTYLE value of 1, but variable labels and all superfluous spaces have been eliminated, and the printer page break character has been replaced by an empty literal string.

```
""
"hdseqhh" 61
"idefseq" 0 0 0
"idcfseq" 0 0 0
"hdprov" 6
"idage" 28 24 0
"idsex" 0 1 1
"idcfrh" 0 1 2
"immmkt" 27953 17350 0
"immtran" 194 371 0
""
"hdseqhh" 62
"idefseq" 0 0 0
"idcfseq" 0 0 0
"hdprov" 6
"idage" 53 50 18
"idsex" 0 1 0
"idcfrh" 0 1 2
"immmkt" 36457 3750 4274
"immtran" 0 1222 0
```

An ASCSTYLE value of 3 produces a report designed to be read using a spreadsheet or a database system. Each unit (in this example each individual) is recorded on a single line, with a single space between each variable value. The first line of the file contains a list of the variable names in the order in which they are written in each line.

```
"hdseqhh" "idefseq" "idcfseq" "hdprov" "idage" "idsex" "idcfrh" "immmkt"
"immtran"
61 0 0 6 28 0 0 27953 194
61 0 0 6 24 1 1 17350 371
61 0 0 6 0 1 2 0 0
62 0 0 6 53 0 0 36457 0
62 0 0 6 50 1 1 3750 1222
62 0 0 6 18 0 2 4274 0
```

An ASCSTYLE value of 4 produces a report identical to that for an ASCSTYLE value of 3, except that the first line (which might be excessively long or inappropriate for some purposes) is eliminated.

```
61 0 0 6 28 0 0 27953 194
61 0 0 6 24 1 1 17350 371
61 0 0 6 0 1 2 0 0
62 0 0 6 53 0 0 36457 0
62 0 0 6 50 1 1 3750 1222
62 0 0 6 18 0 2 4274 0
```

An ASCSTYLE value of 5 produces a report designed to be converted into a compressed format, which can be read by the SPSM. The output generated using this value is utilized by the utilities bldspd60, bldfxv60, and bldwgt60. Please refer to the *SPSD/M Tool User's Guide* for more information.

Below is an example of ASCSTYLE=5 for one household (not all variables values are shown).

```
000001 47 217 2 1 2 1 3 4 1
0 0 0 0 0 0 0 0 42 0 3 11 13 99 1 2 3 0 52 1 0 0 0 0 0 0 0 0 0 1 0 0 0 ...
```

ASCSTYLE=5 produces a fixed format which contains all requested variables, is blank delimited, and contains all records per case beginning with a household record which is followed by individual records.

SAS Output Facility

SAS Institute produces a microcomputer version of their mainframe Statistical Analysis System with substantially identical functionality. SAS has extensive facilities for analysis, reporting, and manipulation of files of microdata. In order to take easy advantage of these facilities, SPSM incorporates an interface to the SAS PC system. This interface produces self-documenting binary files in the same format that SAS itself uses to store data. As a result, data can be made accessible to SAS without the awkwardness of dealing with character file layouts, appropriate variable definitions and labels, etc. Another benefit is that disk requirements and execution time are greatly reduced, since SAS would normally first have to read in a large character file and then convert it into a large binary file before any operations could be performed on the data.

Many operations that can be performed with SAS, such as record selection, creation of new variables, cross tabulation, and distributional analysis can be performed using built-in SPSM facilities. These facilities operate many times faster than their SAS equivalents and don't require large amounts of disk space. Therefore as a general rule, the built-in SPSM facilities should be used in preference to a SAS solution wherever possible. Only if the user needs to perform a function that SPSM cannot handle should SAS be used. Examples of applications that require SAS include: regression analysis, scatter plots, and microdata analysis on three or more model runs simultaneously.

Setting SASFLAG to 1 activates the SAS output facility. SPSD/M always set OUTSAS to “spsmtemp.ssd”. The user can then change "spsmtemp.ssd" to some other name if desired by using the PROC DATASETS procedure in SAS. Each record of the resulting SAS file corresponds to the family level of analysis given by SASUNIT, as shown in the table below. The variable to be output are given in the ASCVARS string parameter, which may be continued onto subsequent lines if necessary using the `\' convention described in the section titled Strings. The SASTITLE string parameter can be used to provide a built-in title to the resulting SAS file.

SASUNIT	SAS output family level
0	Individual
1	Nuclear Family
2	Census Family
3	Economic Family
4	Household

Table 4: Interpretation of SASUNIT values

SPSM creates a special variable, hdwgthhs, designed to be used with the SAS output facility when the sub-sampling facility (see the section titled Sub-sampling Facility) is also active. The value of hdwgthhs is the same as the household weight hdwgthh, except that it has been scaled up by the reciprocal of SAMPLEREQ, the sample size requested by the user. This allows SAS tabulations on sub-samples to have correct overall weighted totals. Please note that hdwgthhs is represented with limited precision on the SAS file, so that tabulations will not agree precisely with tables produced by SPSM. Results will be identical if SAMPLEREQ is set to 1.0, however. Note also that if the user terminates an SPSM run prematurely by pressing CTRL-BRK, hdwgthhs will not correctly reflect the sample actually obtained.

There is a restriction on the use of the OUTSAS parameter. Because SAS files contain an encrypted header structure, SPSM must use the header of an identically-named existing SAS file to create a new SAS file named OUTSAS. The structure of such a file (e.g. the variables or number of records it contains) is irrelevant - all that is needed is a file produced by SAS with the correct name. Such a file can be easily created in SAS using statements such as the following:

```
LIBNAME CD '.';
DATA CD.MYRUN;
RUN;
```

Alternatively, SPSM does know how to create a file named spsmtemp.ssd, and will use this name if a pre-existing SAS template file named OUTSAS does not exist (in this case SPSM will change the file name in OUTSAS to spsmtemp.ssd). The user can then change this name to a more appropriate name in SAS by using statements such as the following:

```
LIBNAME CD '.';
PROC DATASETS LIBRARY=CD;
  CHANGE SPSMTEMP=MYRUN;
RUN;
```

SAS files do not contain all of the information necessary for their use. What is missing is the information on the strings used to display levels of classification variables (e.g. the association of the numeric codes 0-9 with strings giving province names for the variable hdprov). In SAS this information is given by associating a named SAS format with each class variable, and providing a definition of this format in a SAS format library. A SAS format library is in turn produced by providing instructions to the SAS PROC FORMAT procedure. When SPSM produces a SAS file that contains class variables, it also produces an associated file of SAS statements that will create the needed SAS format library. The file that contains these statements is the same as OUTSAS, but with the extension .sfm. The following example illustrates these points.

Assume that a SAS file named sasexam1.ssd was already created in the current directory, and that the following control parameters were included in an SPSM run:

```
SAMPLEREQ     0.049919527         # Size of sample requested
SASFLAG             1             # SAS output facility activation flag
OUTSAS        sasexam1.ssd        # Name of SAS results file (out)
SASUNIT             0             # SAS output family level
SASVARS                           # Variables selected for SAS output
          hdwgthhs hdurb idind idiemp
SASTITLE  Example 1               # SAS file label
```

Then SPSM would overwrite the file sasexam1.ssd with a file containing the requested variables, and in addition produce a file named sasexam1.sfm whose contents would be as follows:

```
PROC FORMAT;

VALUE $URB
    '0' = '>500,000'
    '1' = '100,000-499,999'
    '2' = '30,000-99,999'
    '3' = '<30,000'
    '4' = 'Rural'
   ;
VALUE $IND
    '0' = 'Never Worked'
    '1' = 'Agriculture'
    '2' = 'Other Primary'
    '3' = 'Manufacturing, Non-durables'
    '4' = 'Manufacturing, Durables'
    '5' = 'Construction'
    '6' = 'Transportation & Communication'
    '7' = 'Wholesale Trade'
    '8' = 'Retail Trade'
    '9' = 'Finance, Insce., Real Estate'
    ':' = 'Education & Related'
    ';' = 'Health, Welfare, Religious'
    '<' = 'Recreation, Accommodation, Food'
    '=' = 'Business & Misc. Services'
    '>' = 'Public Administration'
    '?' = 'Worked >5 Years Ago'
   ;
RUN;
```

A typical SAS job to process these files and produce a sample tabulation might be the following:

```
OPTIONS PS=66 LS=100;
LIBNAME cd '.';

%INCLUDE 'sasexam1.sfm';

PROC TABULATE DATA=cd.sasexam1;
  FREQ hdwgthhs;
  VAR idiemp;
  CLASS hdurb idind;
  TABLE idind , hdurb * idiemp * MEAN /RTS=32;
RUN;
```

The log output of the SAS job looks like this (header comments in the source code have been removed):

```
NOTE: Copyright(c) 1985,86,87 SAS Institute Inc., Cary, NC 27512-8000,
U.S.A.
NOTE: SAS (r) Proprietary Software Release 6.03
      Licensed to STATISTICS CANADA, Site 11250001.

NOTE: AUTOEXEC processing completed.
```

```
   10
   11    OPTIONS PS=66 LS=100;
   12    LIBNAME cd '.';
   13
   14    %INCLUDE 'sasexam1.sfm';
NOTE: Format $URB has been output.
NOTE: Format $IND has been output.
NOTE: The PROCEDURE FORMAT used 4.00 seconds.
   43
   44    PROC TABULATE DATA=cd.sasexam1;
   45      FREQ hdwgthhs;
   46      VAR idiemp;
   47      CLASS hdurb idind;
   48      TABLE idind , hdurb * idiemp * MEAN /RTS=32;
   49    RUN;
NOTE: The PROCEDURE TABULATE used 28.00 seconds.
NOTE: SAS Institute Inc., SAS Circle, PO Box 8000, Cary, NC 27512-8000
```

The list output of the SAS job (the result of the PROC TABULATE) looks like this:

```
SAS   2:38 Thursday, November 3, 1988                1
```

	Size of urban area				
	>500,000 Earnings from Employment	**100,000-499,999 Earnings from Employment**	**30,000-99,999 Earnings from Employment**	**<30,000 Earnings from Employment**	**Rural Earnings from Employment**
	MEAN	MEAN	MEAN	MEAN	MEAN
Industry					
Never Worked	6.76	0.00	0.00	0.13	0.00
Agriculture	2698.67	2817.82	10818.05	2061.59	5553.86
Other Primary	31381.33	15286.83	23136.33	34467.59	12669.33
Manufacturing, Non-durables	15810.92	19891.68	14169.33	11600.21	15971.44
Manufacturing, Durables	19260.93	19184.97	18454.21	12534/72	16083.21
Construction	18343.82	6477.05	12288.02	9963.10	13942/21
Transportation & Communication	22447.44	19853.87	19917.40	18425.47	15658.44
Wholesale Trade	18825.63	18380.04	17831.44	20956.16	11500.33
Retail Trade	10105.57	8703.40	10764.16	6511.38	7975.06
Finance, Insce., Real Estate	16439.42	18389.81	22450.45	9194.94	18690.63
Education & Related	17711.69	21707.24	14946.91	13942.98	18353.11
Health, Welfare, Religious	16600.30	20511.94	15668.01	14343.19	12162.36
Recreation, Accommodation, Food	7338.29	6572.39	3373.93	8054.35	3443.01
Business & Misc. Services	11549.08	9165.25	8003.49	9348.16	4496.60
Public Administration	20048.38	14224.60	21265.92	14206.35	14602.06
Worked > 5 Years Ago	162.66	47.73	233.73	215.25	1026.99

Built-in Tabulation Facility

SPSM can generate a number of pre-defined tables that contain general information on the results of a simulation. The tables contain information on the demographic, income, tax, and transfer characteristics of family units.

They can be activated simply by turning on a flag parameter. In addition, auxiliary parameters control features of some of the tables. Because they are so easy to activate, and are designed to provide an overall picture of the tax/transfer system, they are often used as the initial output for an analysis. These tables are also used in the spreadsheet interface facility, described in the *SPSD/M Tools Users Guide*, which extends the utility of the built-in tables by allowing a wider range of run-comparison measures.

All of the built-in tables have an identical set of rows, most of which correspond to variables defined in the <u>*SPSD/M Variable Guide*</u>. Five different column formats are available, corresponding to table numbers 0, 1, 2, 3, and 4. Each of these tables has a separately-selectable auxiliary table, which contains the underlying count of non-zero observations for each cell of the original table. These auxiliary tables, (numbered 0A, 1A, 2A, 3A, and 4A) can be used to derive means or take-up proportions. The following table describes the features of each of the five table types.

Table	Column Dimension	Controlling Parameters
0	Canada total (one column)	T0FLAG, T0AFLAG to activate.
1	Provinces, total	T1FLAG, T1AFLAG to activate.
2	Income groups, total	T2FLAG, T2AFLAG to activate. INCVAR to specify income variable. INCGP to specify income groups.
3	Family type (6), total	T3FLAG, T3AFLAG to activate.
4	Low income ratio groups	PVRAT to specify low income ratio groups. INCVAR to specify income variable. PTF (structural parameter) to specify low income thresholds by family size and urbanization.

There are three other parameters that affect all activated built-in tables.

- OUTTBL contains the name of the file which will contain the generated tables. If OUTTBL is not specified, a file name based on the name of the control parameter file, but with an extension of .tbl, will be used.
- TABUNIT defines the family level of analysis to be used for all activated built-in tables, as indicated in the table below. If a TABUNIT value of 0 (individual level) is specified for tables 3 or 4, individuals are classified based on the type of their containing nuclear family.
- TABDELTA defines the threshold unit consumable income used to define the winner or loser rows of the tables.

TABUNIT	Built-in tables level of analysis
0	Individual
1	Nuclear Family
2	Census Family
3	Economic Family
4	Household

Table 5: Interpretation of TABUNIT values

The following chart uses built-in table 0 (activated by setting T0FLAG to 1) to document the meaning of each row of the built-in tables. Please refer to the <u>*SPSD/M Variable Guide*</u>for a description of the meaning of each variable.

```
Table 0: Results for Census Families

+'---------------------------------------  + -----------------+'
|  Variable  (x1,000,000)                |         TOTAL  |
+'---------------------------------------  + -----------------+'
|Family Units (x1000)                    |          11921.4|              units
```

```
|Persons      (x1000)          |         27433.5|            persons
|SCF Survey Records            |         45371.6|            scfrecs
|SPSD Records                  |         79764.0|            spsdrecs
|Income (Base)                 |        359305.8|            _immicons
|Income (Variant)              |        359305.8|            immicons
|Change                        |             0.0|            (immicons-_immicons)
|Number of Gainers (x1000)     |             0.0|
|Number of Losers  (x1000)     |             0.0|
|No Change         (x1000)     |         11921.4|
|Gainer's Gain                 |             0.0|
|Loser's Loss                  |             0.0|
|Total Income                  |        520266.3|             immtot
|  Market Income               |        437703.4|             immmkt
|    Wages and Salaries        |        341723.5|             idiemp
|    Self-Employment Income    |         27159.4|             imiself
|    Investment Income         |         41430.8|             imminv
|    Other Income              |         27395.1|             immoth
|  Transfer Income             |         82554.7|             immtran
|Total Tax                     |        160955.0|             immtax
|Net Transfers                 |        -78401.8|             imnettr
|Disposable Income             |        408230.1|             immdisp
|Consumable Income             |        359305.8|             immicons
|  Federal Taxes               |        100153.4|             imftax
|    Federal Income Tax        |         59603.5|             imtxf
|    UIC Contributions         |          8173.4|             imuic
|    UI Benefit Recovery       |           131.7|             imuibr
|    CPP/QPP Contributions     |          6068.5|             imcqppc
|    Other Recoveries          |           644.4|             imothrep
|    Federal Commodity Taxes   |         25532.0|             imtxfc
|  Federal Transfers           |         72064.5|             imftran
|    New programs              |             0.0|             imiotg
|    Federal Family Allow.     |          2855.3|             imffa
|    CTC / Child Benefits      |          2287.3|             imctc
|    OAS                       |         14354.2|             imioas
|    GIS                       |          4318.9|             imigis
|    SPA                       |           565.3|             imispa
|    UI Benefits               |         18789.6|             imiuib
|    CAP (Federal Portion)     |          5014.2|             imfsa
|    CPP/QPP Income            |         16493.3|             imicqp
|    Other Federal Transfers   |          4697.3|             imfoth
|    Sales Tax Credit          |          2688.0|             imfstc
|    Que. Tax Abatement ref.   |             0.7|             imqtar
|  Federal Net Balance         |         28089.5|             imfedbal
|  Provincial Taxes            |         60802.9|             imptax
|    Provincial Income Tax     |         37415.1|             imtxp
|    Provincial Commodity Tax  |         23388.2|             imtxpc
|  Provincial Transfers        |         10490.9|             imptran
|    Family Programs           |           519.8|             impfp
|    Elderly Programs          |           611.7|             imigist
|    CAP (Provincial Portion)  |          7853.3|             impsa
|    Tax Credits & oth Trans.  |          1506.2|             imptc
|  Provincial Net Balance      |         50312.3|             imprvbal
```

```
+'--------------------------------------  + -----------------+'
```

X-tab Facility

The design of the SPSM recognizes that the built-in tables will not always meet the analyst's needs. Thus, it also provides a powerful facility that allows a user to request the specific tables desired. A separate auxiliary guide, the SPSD/M X-tab Facility, presents detailed documentation on the X-tab facility in a comprehensive format that is also suited to those learning the X-tab facility. This section, in contrast, presents the key information in a more concise format.

Specification of User-defined Tabulations

The user-defined tabulations are activated by XTFLAG set to 1. The analyst specifies the desired tabulations via the XTSPEC element of the control parameter file. XTSPEC is one of the SPSM's string parameters. After the parameter name, XTSPEC, the parameter content consists of tabulation requests separated by semicolons. The following example contains two tabulation requests and would generate two tables.

```
XTSPEC
EF: hdprov * {units};
CF:cftype+*{immtot/units};
```

The first table would display the numbers of economic families by province. The second would tabulate the average total income per census family for several different categories of census family and also for all census families taken together.

The last line of the request now accepts semi-column. It is highly suggested to use it.

Components of a Tabulation Request

An individual tabulation request consists of multiple components. It typically begins with the specification of the relevant unit of analysis, i.e. individual (IN:), nuclear family (NF:), census family (CF:), economic family (EF:) or household (HH:). If the tabulation request does not include a unit specification, a default unit of individual (IN:) applies. The remainder of the tabulation request consists of one or more "levels" separated by asterisks. One of the levels, the tabulation level, indicates the item or items to be tabulated. Any remaining levels are classification levels. They specify the particular pattern of categorization desired in the table. Thus, in the first of the two tabulations requests shown above, the level of analysis is the economic family (EF:). Further, the quantity being tabulated is the number of such families (units), and the analyst wants to see these counts of families broken out by province (hdprov).

The Tabulation Level of a Tabulation Request

In any given tabulation request the SPSM identifies the unique tabulation level by its enclosure in braces, the { and } characters. A tabulation level may specify multiple items for tabulation in the same table. If multiple items are present in the tabulation level, the analyst must separate them by commas. The following specification would tabulate total income

XTSPEC

```
CF:{immtot,
    units,
    immtot/units}
    *hdtenur+;
```

accruing to census families, the numbers of such families, and the average income per census family, all broken out by tenure categories (rented, owned with mortgage, etc.). The request can also be put on two lines, like the following, but we strongly suggest to use the previous form in order to easily keep track of the variables and their output formats.

XTSPEC

```
CF:{immtot,units,immtot/units}*hdtenur+;
```

Items Suitable for Tabulation

Naturally, there are some restrictions on just what the X-tab facility can tabulate, but the design of the SPSM makes it quite flexible. For starters, the analyst can tabulate any analysis variable available in the SPSD or calculated in the (SPSM) model. Complete lists of these variables appear in the *SPSD/M Variable Guide*. Further, as illustrated in the preceding examples, the analyst can define "on the fly" the desired tabulation expressions, constructing them from any of the preceding types of variables and appropriate mathematical operators. The section titled Expressions of this guide contains a full discussion of the SPSM's use of expressions.

Expressions Involve Functions of Aggregated Variables

When the X-tab facility tabulates an expression, it begins by creating an aggregate "sum" value for each variable in the expression. Only then, using the summed values, does it execute the operations (addition, multiplication, division, etc) in the expression. This is typically the analyst's precise intention. That is, an analyst tabulating {immtot/persons} wants to divide a sum of incomes by the relevant count of total persons. This result is exactly what the X-tab facility would provide.

Similarly, the analyst tabulating {immtot/persons} does not typically wish to sum, across units, the per capita income of those units, the result that would obtain if the X-tab facility evaluated the expression for each record instead of using the aggregate values. If the analyst really needs intra-record calculations prior to tabulation, then he or she would use a two step approach. First, an ex variable, defined at the appropriate level of analysis, would execute the intra-record calculations. Second, that ex variable would appear in the tabulation request.

Imposing Qualifiers on Items Being Tabulated

In the absence of explicit instructions from the analyst, the X-tab facility makes informed choices as to labels, scaling, number of decimal places, etc. However, it also provides a mechanism for the analyst to exercise, when desirable, explicit control over these features. Specifically, the analyst can impose one or more qualifiers on any variable or expression being tabulated. The four available qualifiers, S, P, L, and M are discussed in turn below.

The S qualifier controls the scaling of the tabulated variable. Thus, for example, a qualifier of S=6 would yield a table entry denominated in millions. The X-tab facility would reflect this selection by including the string "(M)" in its labeling for the variable. The analyst should restrict S qualifiers to the range [-6, 9].

The P qualifier controls the number of decimal places displayed for a table entry. For example, a qualifier of P=2 would yield table entries that included decimal points followed by two digits. A qualifier of P=0 would yield table entries without decimal points. The analyst should restrict the value of P to the domain [0, 8].

The L qualifier permits the analyst to override the default labeling for table entries. The default labeling is already fairly sophisticated -- the X-tab facility uses a variable label when it is available; in its absence the X-tab facility makes the label out of the name of the tabulated variable or the text of the tabulation expression. The text of the label supplied in an L qualifier replaces the X-tab facility's default label. Thus, for example, an analyst tabulating {scfrecs } might specify a label of "SCF Records" for improved readability. Just as the user would expect, the X-tab facility adds to the label all appropriate "suffixes," e.g. "(%)" for entries with a scaling factor of -2.

The M qualifier permits the analyst to request that a normalization be carried out along one of the table's margins. The most common application of this qualifier is the calculation of percentage distributions; indeed, use of the M marginal sets the default-scaling factor to -2. As an example, consider the tabulation request:

```
CF: {units:M=hdprov P=1} * cftype+ * hdprov+;
```

The request would yield a table showing the percentage distribution, across provinces, of numbers of census families broken out by census family type. The M=hdprov qualifier indicates that the normalization will occur over the province dimension. The qualifier P=1 ensures that the table entries will display results down to tenths of a percent.

Syntax for Qualifiers

The analyst indicates the presence of qualifiers by a colon after the tabulated variable or expression. The qualifiers then follow the colon and precede the next tabulation variable or the brace at the end of the tabulation level. A qualifier takes the form Q=C, where Q represents the qualifier character (S,P,L or M), and C represents the qualifier content (a number, label or variable name depending on the type of qualifier used). If there are multiple qualifiers, they are separated from each other by spaces. Within an individual qualifier, however, there should be no spaces on either side of the "equals" character (=). Note that each of the tabulation variables or expressions in the tabulation level of a table request may have its own set of qualifiers. The example tabulation request

```
EF: {units: S=6 P=1,
     units :L="EF Distr." M=hdprov }
    *hdprov+ * hdtenur+;
```

shows some of these possibilities. The resulting table would display the numbers of economic families by province and tenure, as well as their distribution across provinces.

Classificatory Levels

A tabulation request typically includes one or more classificatory levels. A classificatory level consists of the name of a classificatory variable, i.e. an "integer" variable that classes the relevant unit into mutually exclusive and exhaustive categories. These levels are separated from the tabulation level, and each other, by asterisks. Functionally, a classificatory level specifies a dimension along which the X-tab facility is to break out the item(s) in the tabulation level. Thus, in the illustrative tabulation request

```
IN: {scfrecs: L="Records" P=0 S=0} * idsex * idmarst+;
```

the analyst is counting numbers of SCF records for individuals across sex and marital status.

Classificatory variables typically come from two sources: database classificatory variables or model classificatory variables in the control file. "Glass box" users can also define their own classificatory variables.

Creating an "All" Category Via the "+" Suffix

Used by itself as a classificatory level, a classificatory variable will generate a set of mutually exclusive categories in the table. However, the analyst will often wish to have supplementary "sum across categories" entries in the table. The + suffix provides this capacity. When the analyst includes a + suffix on a classificatory level's variable name, the X-tab facility generates a new "All" category equal to the sum across the variable's categories. The analyst can use the + suffix independently across classificatory levels in a tabulation request.

Thus, our previous example of

```
IN: {scfrecs: L="Records" P=0 S=0} * idsex * idmarst+;
```

will create an "All" category for the marital status dimension, but not the sex dimension.

Ordering of Levels and Table Appearance

The analyst's ordering of the tabulation and classificatory levels in tabulation request controls the appearance of the resulting table. Generally speaking, the further to the right a level is, the more frequently its categories cycle. Multiple tabulation items in a tabulation level are effectively treated as categories. Thus, the rightmost level in a request controls the column categories of the table. The next-to-last level controls the row categories, the "next-to-next-to-last" level controls the segment categories, etc. This convention gives the analyst good control over the appearance of the resulting table.

Continuing with the earlier example

```
IN: {scfrecs: L="Records" P=0 S=0} * idsex * idmarst+
```

we see that the resulting table will have marital status categories in its columns, and sex as the control variables for its segments.

An exception to the general rule arises when there is only a single item in the tabulation level. For this special case, the X-tab facility processes the tabulation request as if the tabulation level were the first in the request. This treatment prevents multi-segment tables that have only a single row or single column in its segments.

Creating a Multi-line Table Specification

The XTSPEC parameter is one of the string parameters in an SPSM control parameter file. SPSM expects a request to be terminated by a semi-column. The request can be spread on as many lines as required to have an easily readable request. The important is to put carriage return at the end of each line of the request. The following example shows a request of multiple variables output.

```
IN: imfiler+  *
  {imitot:S=0,
    imdedfn:S=0,
    imdedft:S=0,
    imitax:S=0,
    imbft:S=0,
    imfsur:S=0,
    imtxf:S=0,
    imqtaa:S=0,
    imtxp:S=0,
    units:S=0} * hdprov+;
```

Related Control Parameters

Of all the SPSM parameters controlling the production of tables, XTSPEC is the real workhorse. However, three other mandatory parameters have a direct effect on the generation of the tables.

The XTFLAG parameter is a flag that tells the SPSM whether or not to produce the tables specified by XTSPEC. The tables are produced only if XTFLAG is set to 1.

The XTLINES parameter tells the SPSM how many lines can be fit on a page. It helps the X-tab facility to avoid both splitting a given segment across multiple pages and wasting paper by beginning each new table or segment on a new page. XTLINES must take on a value in the range 0 to 32767. A typical value is the 66 lines that will fit on an eleven inch page at the standard six lines per inch.

The XTCOLS parameter affects the tradeoff between having multiple lines in column headings and having wider tables. XTCOLS must take on a value in the range 80 to 32767. The general interpretation of the parameter's value is that of a desired maximum number of print columns for the output file. However, the specific algorithm by which XTCOLS affects the printed output is quite complex and cannot be conveniently summarized with accuracy. A typical value for XTCOLS is 132; this value corresponds to the maximum number of print positions available on many printers.

Level of Analysis Issues

The SPSD/M and its X-tab facility operate in a hierarchical context. Although five levels of unit of analysis are supported (IN, NF, CF, EF and HH), the underlying data exist only at the household and individual levels. Variables for intermediate levels must thus be defined by propagating household variables downward or rolling up variables defined at the individual level. Floating-point variables are fairly straightforward in this regard because they can be aggregated up from the individual level. For example, disposable income for an economic family is simply the sum of the disposable incomes of its members. However, using expenditure variables defined at the household level in a table of nuclear families requires some care in interpretation.

Classificatory variables, crucial for cross tabulations, are still more challenging. Although propagation downward, e.g. province of residence for a census family, presents no problems, performing roll-ups can be tricky. For example, idsex is, mechanically, acceptable as a classificatory variable for a table defined at the census family level. What, though, does it represent? The analyst must be aware that the census family will be classified based on the sex of one representative individual member. Correspondingly, he or she should be aware of the rules by which the SPSM identifies such individuals.

In summary, even though the SPSM provides reasonable defaults for the typical things that an analyst will do, roll-up is a potentially complicated issue. The section titled Family Level of this User's Guide provides the authoritative discussion of roll-up issues throughout the SPSD/M.

Tabulations Depend on Selection Parameters

Because table entries depend on aggregates of relevant (weighted or unweighted) cases or variables in the SPSD, they necessarily depend on the criteria used to select those cases to be processed. In particular, the entries in a table, and the interpretation of those entries may depend critically on the selection criteria used for an SPSM run. Three control parameters are relevant. (1) The SELFLAG parameter determines whether any selection will take place. (2) The SELUNIT parameter controls the type of unit that is selected. (3) The SELSPEC parameter defines the condition that cases must pass to be included in the analysis; it includes the potential for specifying criteria that depend simultaneously on multiple levels of unit of analysis.

Problems of interpretation are especially likely to arise when the SELUNIT level refers to a subset of the level associated with a tabulation request. The difficulties arise because one might only be including subsets of a family in the resulting tables. The section titled Selection Facility of this User's Guide includes a more complete discussion of selection criteria and their impacts on SPSM outputs.

Distributional Analysis Facility

The primary function of the SPSM distributional analysis facility is to allow the analyst to gauge visually the statistical properties of an SPSD/M variable. A frequency histogram, using percentile cut-points provided by the analyst, and incorporating tail truncation if desired, is generated from a sample of observations of the variable. Because the observations are subject to selection (see the section titled Selection Facility) and the variable can be defined through the user-defined variable facility (see the section titled User-defined Variable Facility), the distributional analysis facility can be a powerful exploratory tool.

An independent secondary function is to record extreme values of the requested variable for all observations, and identify the households in which these extremes occur. Using the resulting household identification numbers, the selection facility, and the case output facility (see the section titled Text Output Facility), the analyst can explore the properties of these households and understand what it is about them that results in an extreme value for the variable in question.

The distributional analysis facility is activated by setting the DISTFLAG parameter to 1. The variable to be analyzed is given in the DISTVAR string parameter, and observations are made at the DISTUNIT level of analysis, as indicated in the following table.

DISTUNIT	Distributional analysis family level
0	Individual
1	Nuclear Family
2	Census Family
3	Economic Family
4	Household

Table 6: Interpretation of DISTUNIT values

To generate the frequency histogram, up to DISTSAMP observations will be made, and observations with value zero are included if the DISTZERO flag is set to 1. These observations, and their associated household weights, are kept in memory and sorted after SPSM has completed processing all households. Increasing DISTSAMP thus increases memory requirements. Only DOS system is subject to a maximum. Other OS (UNIX, WINDOWS) can use the full sample.

The DISTP vector parameter contains the percentile cut-points to use in generating the frequency histogram. The first and last element of DISTP are interpreted as values to use to truncate the distribution below and above, respectively. If these values are set to 0 and 100, then the entire sample will be used to generate the histogram. It is often useful to have some tail truncation, however, since the inclusion of extreme values reduces the detail that can be shown for the bulk of the observations.

Two parameters remain to be described. DISTPWID is the number of print positions used to produce the histogram. A value of 79 is suitable for display on the screen of your PC, but a larger value may be given if the output is destined for a printer with a wider capacity. DISTPHGT gives the number of lines used to produce the histogram. A value of 20 is suitable for screen display.

The following example illustrates the above points. The distributional analysis facility is used, at the Census family level, to analyze the results. The relevant control parameters are as follows:

```
DISTFLAG            1
DISTUNIT            2
DISTVAR           ex0
DISTSAMP         4000
DISTZERO            1
DISTP         13
        1
        5
       10
       20
       30
       40
       50
       60
       70
       80
       90
       95
       99
DISTPWID           70
DISTPHGT           17
```

The output resulting from the distributional analysis facility appears as follows:

```
Distribution report: Average Tax Rate for Census Families

   Total observations =   3278
   Zero observations  =    798
   The following statistics are based on all 3278 observations.

   Descriptive Statistics:

   Sum of weights          =     10053381
   Weighted Sum            =   1088932.76
   Weighted Sum of Squares =    202166.59
   Weighted Mean            =         0.11

   Extreme Values (with associated household numbers):

      Minima  hdseqhh                Maxima  hdseqhh

       -0.00     2111                  0.57     2354
        0.00        1                  0.57     1853
        0.00        2                  0.57     2113
        0.00        3                  0.57     2114
        0.00        4                  0.57     1034

   Selected Quantiles:

   Q1 =0.00   P1=0.00   P90=0.23   P20=0.00   P60=0.14
   Med=0.11   P2=0.00   P95=0.26   P30=0.01   P70=0.16
   Q3 =0.18   P5=0.00   P98=0.30   P40=0.08   P80=0.19
             P10=0.00   P99=0.33

   Histogram Plot:

   +--+
   |  |
   |  |
   |  |
   |  |
   |  |
   |  |
   |  |
   |  |
   |  |
   |  |
   |  |
   |  |
   |  |                         +----+
   |  |            +------------+    +-----+
   |  |            |                       +-------+
   |  +------------+                               +-------------------+
   +--+------------+------+-----+----+-----+-------+-------+-----------+
   P1 P30          P40    P50   P60  P70   P80     P90     P95
P99
   0.00            0.08   0.11  0.14 0.16  0.19    0.23    0.26
.33
```

The first part of the report is based on all records (subject to selection if SELFLAG has been turned on). It includes a section on descriptive statistics on the requested variable, followed by a section that gives the five smallest and five largest observations, along with the sequence numbers (variable hdseqhh) of the households in which the extreme value was observed.

The second part of the report is based on the first DISTSAMP observations. It contains estimates of selected quantiles of the distribution, and the frequency histogram. In interpreting the frequency histogram, note that area corresponds to the (weighted) number of observations. The percentiles requested in DISTP are printed immediately below the histogram, while the next line gives the values of the variable corresponding to the percentiles. If there is insufficient space, some of the labels are suppressed.

Turning Point Facility

Introduction

The SPSM Turning Point Facility provides a means to analyze the points at which an individual household's marginal tax rate changes as its income increases. The marginal tax rate is the proportion of tax paid on one additional dollar of income. The turning points represent changes in the slope of the entire tax/transfer function (the marginal tax rate). The facility computes the various income levels where an individual household, given its characteristics, would experience changes in its marginal tax rate and then outputs information on the current value of a number of variables at each of these levels of income.

The level of income where turning points will typically occur in an individual's tax/transfer function includes the income level where CPP contributions are first deducted, the income level where UIC contributions are first deducted, the level of income where federal income tax becomes payable, etc. For example, the actual 1988 tax/transfer function of a single resident of Nova Scotia, who worked full time for 52 weeks in 1988 and whose only source of income was from salaried employment, is presented in the following diagram**. Note that the chart was produced using other software using results from the SPSM.**

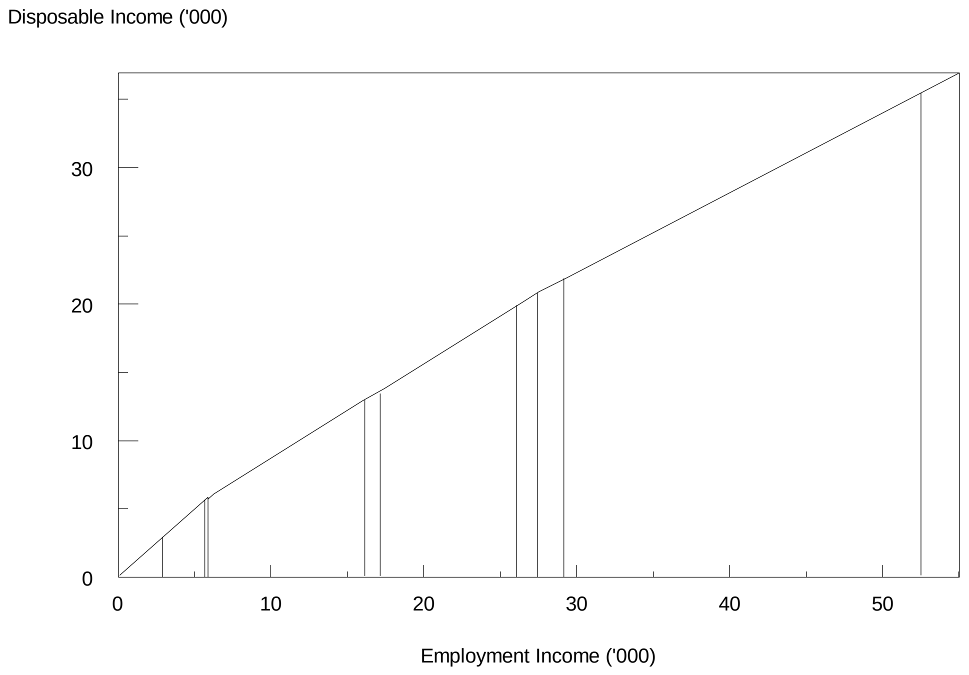

Figure 1. Tax Function for Household Head, Taxation Year 1988

The turning point facility identified the level of income at each point in this tax/transfer function where the slope of the tax function changed, i.e. the marginal tax rate changed. Each turning point is marked with a vertical line on the diagram above. From the output of the turning point facility for this individual, the marginal tax rates can be computed and are presented in the following diagram. The reason for each turning point is recorded in the notes following the diagram:

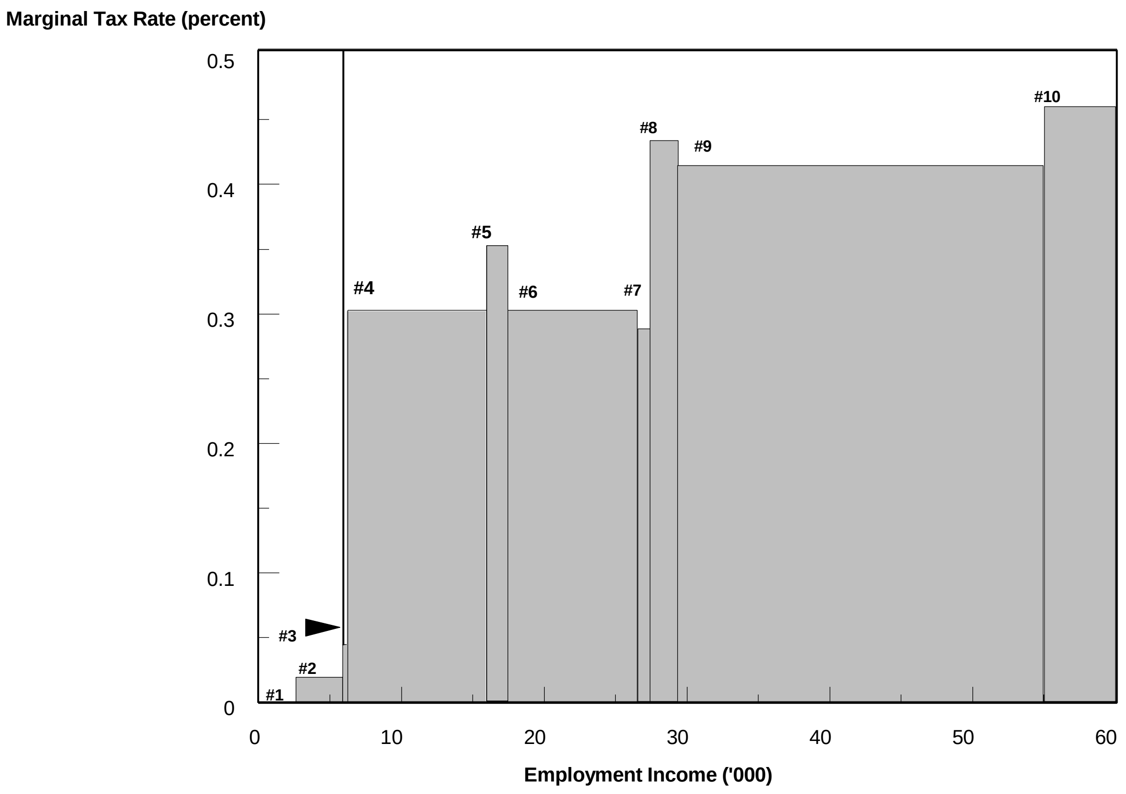

Figure 2. Turning Points For Household Head, Taxation Year 1988

[1] The individual receives income from employment

[2] Income level where CPP contributions are paid.

[3] Income level where UIC contributions are paid.1

[4] The individual pays Basic Federal Income Tax and Federal Surtax at this income level.

[5] The Federal Sales Tax Credit begins reduction at this income level.

[6] Income level where the Federal Sales Tax Credit becomes zero.

[7] Income level where CPP contributions reach their maximum tax credit value.

[8] Individual reaches the second federal income tax bracket.

[9] Income level where UIC contributions reach their maximum tax credit value.

[10] Individual reaches the third federal income tax bracket.

Note that the chart was produced with software using SPSD/M output. The next section explains how to access the SPSD/M turning point facility. Following this discussion, a worked example is presented which describes the control parameter file that produced the data represented in the above diagrams. An example of the text output of the turning point facility is also presented.

Turning Point Facility Operation

The SPSM turning point facility processes one household at a time. The facility identifies the various income levels, to the penny, at which the household would experience changes in its marginal tax rate, due to changes in the applicability of taxes or transfers, given each household member's original income tax return information as recorded in the database. [The turning point facility is not an optimizing model; therefore, it acts to maximize the level of household disposable income for each given income only. However, the SPSM model will optimize the values of several of the tax/transfer parameters; for example, child care expenses]. To activate the turning point facility the TPFLAG control parameter must be set to 1.

A household is selected for turning point modification through the selection facility (see the section Sub-sampling Facility). The parameter SELSPEC should contain an expression which defines the type of household the analyst wishes to examine, for example the household's province or number of children, etc. The parameter SELMAX, which defines the maximum number of households selected, must be set to one to select only one household for modification by the turning point facility. It is possible to select more than one household; however, the greater the number of households selected the longer the processing time will be. The first household examined by the SPSM which contains at least one member that meets the selection criteria will be chosen from the SPSD database. If no household within the database meets all of the selection criteria then no household will be chosen for turning points modification. Because the turning point facility computes turning points at the household level only, the parameter SELUNIT must be set to 4, the household level. If a lower level of analysis is chosen, the analyst may observe output containing "unexplainable" turning points, which result from changes in the tax/transfer parameters of other members of the household which have not been selected for output.

The analyst also must specify which household members' incomes will be incremented by the turning point facility. Household members not selected for the income increment will retain their original income values at each turning point. This selection is made with the parameter TPSPEC, a string containing an expression (see Section Variables for a description of SPSM expressions) which identifies whose income, within the chosen household, will be modified by the turning point facility. For example, the analyst could set TPSPEC to the expression idcfrh==0 which would cause only the income of the census family head to be modified by the turning point facility. When this parameter is set to 1, all household members' incomes will be incremented. The analyst is cautioned that the number of turning points produced will increase substantially as the number of members with modified income increases.

The analyst must also identify which income source, or sources, will be incremented by the turning point facility. The TPVARS parameter specifies the selected income source or sources. The facility can modify any database analysis variable (i.e. containing the prefix id). If more than one variable is specified then each must be entered as a blank separated list of variables. Typically, the analyst will want to modify only a single source of income, such as idiemp , employment income, but his/her decision will depend on the income range selected and the results desired.

The income range, over which turning points will be calculated, must also be specified. The analyst must set a lower and an upper limit for the income range, expressed as dollar values or expressed in relation to household's income, depending on the value of TPMETH. For example, the analyst can choose an income range between $0.00 and $80,000 of household income, or an income range between the current household income value and ten times the current household income value. To specify dollar value modification, the parameter TPMETH is set to 1. Scaling factor modification is activated when the parameter TPMETH is set to 2.

When dollar value modification is selected, the value of the control parameters TPLL, (turning points lower limit), and TPUL , (turning points upper limit), is set to a particular dollar value. For example, if TPLL is set to 0 and TPUL is set to 80000, then turning points which occur in the range of 0 to 80000 dollars of income will be calculated. Selecting an upper limit that is greater than 80,000 dollars, for example, will not produce more turning points if there are no further changes in the marginal tax rate at a higher income value. When scaling factor modification is selected the value of TPUL becomes a scaling factor, for example setting TPUL to 10 would produce turning points up to 10 times the original household income value. In this case if the value of TPLL is set to 0 then scaling will begin from 0 dollars of income, if TPLL is set to 1 then scaling will begin at the original level of income, and if TPLL is set to 2 then scaling will begin at twice the original level of income. Scaling factor modification makes it possible to modify the incomes of more than one household member and observe the changes in their taxes and transfers as their incomes increase by a certain percentage. Remember to adjust the parameters TPLL and TPUL when changing from dollar value modification to scaling factor modification.

If more than one household member is identified for turning points modification and TPMETH=2 is selected then the relationship between the incomes of the selected household members is preserved throughout the analysis. If, for example, the income of the first selected member is twice as large as the income of the only other selected member then the first member will receive two thirds of every cent of income increment. Similarly, if more than one income source is selected for incrementation and TPMETH 2 is selected then the relationship between the two income sources is preserved. If the first income source is twice as large as the second income source then the first source will receive two thirds of every income increment. If TPMETH 1 is chosen then both selected household members and/or both selected income sources increase by exactly the same quantity at each turning point. Therefore, TPMETH 2 is the more appropriate method for estimating the turning points of more than one household member or more than one income source.

The form and content of the output of the turning point facility must also be specified. Results can be output using the text output facility (see section 12 of the *SPSD/M User's*

Guide) or the analyst may export results using the SAS output facility for further analysis (*SPSD/M User's Guide* Section 13). **A concise text output in a readable form is available to the user with the inclusion of `/spsd/detsum.cpi` in the control parameter file**. To increase the precision of the text output, the control parameter ASCEXTPRC could be set to 2 or more. Accurate computation of marginal tax rates can require this additional precision.

Note that before the turning point estimation is run, the control parameter ROUNDFLAG must be set to 0. If the number rounding facility is not deactivated then many superfluous turning points will result. The analyst should also set the tax/transfer parameter CTFLAG to 0. If the commodity tax calculation facility remains activated then commodity taxes will be calculated based on a constant pattern of expenditures, regardless of the level of income. Therefore, the only effect of commodity taxes the model can produce is an increase in the marginal tax rate by a constant amount, at all income levels.

The effective marginal tax rate displayed in the previous diagrams was computed by the following formula:

$$MTR_{i(i+1)} = \frac{immicons_{i+1} - immicons_{i}}{idiemp_{i+1} - idiemp_{i}}$$

This formula represents the marginal tax rate in the interval between turning points i and i+1. The denominator of this function contains the variable specified by TPVARS.

Example

In this example simulation, the SPSM turning points facility will be activated to modify the 1988 employment income of a single individual, who has the following specific characteristics:

Characteristics	**SPSM expression**
• person is the sole member of his/her household	hhnin==1
• person has some income	immtot>0
• person's only source of income is from employment	immtot==idiemp
• person is a full-time employee	idlyfp==1
• person worked at least 50 weeks during 1988	idlyww>=50
• person is a resident of Nova Scotia	hdprov==2

The relevant section of the control parameter file (`.cpr`)which selects an individual who meets the above specifications is presented below.

```
###
## 2.1.7 Record selection facility
###

SELFLAG       1   # Selection facility activation flag
SELUNIT       4   # Selection facility family level
SELSPEC       \   # Selection specification
hdprov==2 && hhnin==1 && idiemp>0 && idlyfp=1 && idlyww>=50 \
&& immtot==idiemp
SELMAX        1   # Selection facility maximum # of households


###
## 2.1.9 Turning Point facility
###

TPFLAG        1   # Turning point facility activation flag
TPSPEC        1   # Expression identifying individuals to Change
TPVARS   idiemp   # Variables to modify
TPMETH        1   # Method for modifying variables
TPLL       0.00   # Lower limit for modified variables
TPUL   60000.00   # Upper limit for modified variables
```

SELFLAG turns on the selection facility. SELUNIT is set to the household level. SELSPEC identifies the relevant households within the database. SELMAX = 1 indicates that turning points will be calculated for only the first household which matches the SELSPEC. TPFLAG activates the turning point facility. TPSPEC = 1 indicates that all household members incomes will be incremented; however, SELSPEC has restricted the number of household members to one. Dollar Value modification is chosen, TPMETH = 1; therefore, turning points will be calculated over the income range 0 to 60000 dollars of employment income. The control parameter ROUNDFLAG is also set to zero.

The text output facility is set for appropriate output. The settings used here are found in the file `\spsd\detsum.cpi`. Lastly the commodity tax model is turned off by setting CTFLAG = 0 in the appropriate `.mpr` file.

Users will notice that for the calculation of the MTR in the current example, total income is equal to employment income due to our selection criteria. Also, consumable income is equal to disposable income because the commodity tax model has been turned off.

The following is a listing of the text output facility output produced for the run. This edited list shows all relevant, non-zero variables for this individuals taxable position. The list is a subset of the list produced by the output facility settings contained in the file `\spsd\detsum.cpi`.

```
---------------------------------------------------------------------------
1     hdseqhh   Household sequence number .......  217  217  217  217  217
2     hdprov    Province ................  .....    2    2    2    2    2
3     idage     Age ....................... ....   42   42   42   42   42
4     idsex     Sex ....................... ...    0    0    0    0    0
5     idmarst   Marital status  ................    3    3    3    3    3
6     idlyww    Weeks worked ......................52   52   52   52   52
---------------------------------------------------------------------------
7     idiemp    Earnings from employment ........    0 2600 5876 5876 6267
8     imitot    Total income ....................    0 2600 5876 5876 6267
---------------------------------------------------------------------------
9     imcqppc   CPP/QPP contributions ...........    0    0   66   66   73
10    imuic     UIC contributions .................  0    0    0  138  147
11    idothdn   Other deductions from total income .13   13   13   13   13
12    imdedft   Deductions from total income ....   13   13   13   13   13
13    iminet    Net income .....................     0 2587 5863 5863 6254
---------------------------------------------------------------------------
14    idchara   Charitable donations and gifts ..   34   34   34   34   34
15    imitax    Taxable income ..................    0 2587 5863 5863 6254
---------------------------------------------------------------------------
16    imfedtax  Federal tax before tax credits ..    0  440  997  997 1063
17    imbtc     Basic personal tax credit ...... 1020 1020 1020 1020 1020
18    imchartc  Charitable donations tax credit .    0    6    6    6    6
19    imcppctc  CPP contributions tax credit ....    0    0   11   11   12
20    imuictc   UIC contributions tax credit ....    0    0    0   23   25
21    imtaxcr   Total tax credits .............. 1020 1026 1037 1060 1063
22    imatxcrt  Total tax credits applied  ......    0  440  997  997 1063
23    imfstc    Federal sales tax credit ........   70   70   70   70   70
---------------------------------------------------------------------------
24    immemp    All employment income ...........    0 2600 5876 5876 6267
25    immmkt    Market income ...................    0 2600 5876 5876 6267
26    imftran   Federal transfer income .........   70   70   70   70   70
27    immtran   All transfer income ............   70   70   70   70   70
28    immtot    Total income ....................   70 2670 5946 5946 6337
29    immtax    All taxes .......................    0    0   66  204  221
30    imptax    Provincial taxes ................    0    0    0    0    0
31    imftax    Federal taxes ...................    0    0   66  204  221
32    immdisp   Disposable income ...............   70 2670 5880 5742 6117
33    imtxfc    Federal commodity taxes .........    0    0    0    0    0
34    immicons  Consumable income ................ 70 2670 5880 5742 6117
---------------------------------------------------------------------------
```

Because TPLL was set to zero, the initial level of employment income is zero (column 1, row 7). The first turning point, employment income of $2,600 (column 2, row 7), occurs because a one cent increase in income beyond $2,600 results in the deduction of CPP contributions (column 2, row 9). The third level of employment income, $5,876, identifies the threshold level of income beyond which UIC contributions are deducted. A change in employment income of less than a cent more than this income level, represented as also $5,876 (column 4, row 7) results in UIC contributions of $138 deducted (column 4, row 10). The fifth level of employment income, $6,267, identifies the point where the individual's federal income tax payable equals his total non-refundable tax credits (column 5, rows 21 and 22). After this point, he pays federal income tax. The remaining turning points in this individual's tax/transfer function are not presented here but were discussed earlier.

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/ba88t#temp#N#N#N#N
```

would invoke SPSM using the control parameter file `/spsd/ba88t.cpr` and create output files `temp.cpr`, `temp.tbl`, etc. The N's indicate `No' responses to the normal questions that SPSM asks the user. In the more complicated example

```
spsm temp##Y#SAMPLEREQ#.001#read#/spsm/example/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 `/spsm/example/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.

Note that the MS-DOS output re-direction facility can be used to save the dialogue for later perusal. For example, the line

```
spsm /spsd/ba88t#temp#N#N#N#N > temp.log
```

is similar to the example given above, except that the lines that SPSM would normally write to the screen are instead written to the file `temp.log`.

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/ba88t
temp
N
N
N
N
```

or a single line

```
/spsd/ba88t#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.

Miscellaneous Facilities

Parameter difference report

It is often desirable to know the exact tax/transfer parameter differences between the base and variant. If the PRDFFLAG parameter is set to 1, and if both base and variant results are activated (i.e. both BASMETH and VARMETH are non-zero) then a report displaying parameter differences between base and variant is produced on the file specified by OUTTBL. This is the same file that contains any requested built-in or user-specified tables. The user has no control over the format of this report. If additional control is required, a more sophisticated parameter difference report capability is provided in the compparm stand-alone utility, which is described in more detail in the *SPSD/M Tools User's Guide*. The PRDFFLAG parameter functions even when working with base results files, (i.e. if BASMETH=1), since SPSM results files contain a copy of the tax/transfer parameters used to produce them.

Random number facility

It is typically the case that a tax/transfer program, even though it may be targeted at a specific population, does not succeed in reaching that population. This `take-up' phenomenon can be modeled using pseudo-random numbers, if a take-up probability is given. The SPSM provides up to 20 independent random number streams, which are controlled by the SEED vector parameter. The number of elements of SEED is the number of independent random numbers generated. Each element of SEED provides an integer used to start each random number generator. These elements are usually set to different values, in order to generate independent random numbers.

The random numbers are generated using a standard integral-congruential algorithm, are stored in the individual level variables idrand0 through idrand19 and take on values from 0.0 to 1.0 inclusive, with uniform probability. The random number facility has been implemented to operate reproducibly, independent of any record selection. In other words, for a given value of SEED, the random numbers idrand0 through idrand19 will have identical values for a specific individual in the database, independent of any selection that may have been requested.

Low-income line analysis

The SPSD/M can produce a standard table showing the distribution of many variables by a grouping of the ratio of a household's income to a specified threshold. The following parameters control the use of this standard table.

`PTF`: Low income cut-off [size, urban] NOT [size,type]

PTF allows the use of Statistics Canada's Low Income Cut-offs in SPSM. It is indexed by family size and size of urban area.

`efpovthr` Economic family poverty threshold

efpovthr simply contains the appropriate value (i.e. corresponding to the number of persons and degree of urbanization) for the current economic family, as given by the PTF array parameter.

`impovinc` Income for poverty measurement

This variable contains the value of income used for low-income line analysis. To avoid double counting, this value contains economic family income, but only for the first person in the economic family. An expression of the form EF:impovinc < efpovthr would identify individuals in families below the low income threshold.

The family income for poverty analysis purposes is total income less the federal child tax credit and sales tax credit. Note that this is the current Statistics Canada income definition used for Low Income Cut-offs.

Note that because the SPSD attempts to correct certain under-reporting problems present in its data sources, counts may differ from those published elsewhere. It is possible to turn off many of these imputations if desired (see the section Imputation Control).

SCF Replication

NOTE: SCF replication is not possible in this version but may be available in the next version.

A number of variables allow the user to replicate the population that underlies Statistics Canada publications based on the Survey of Consumer Finances (SCF). The variables *idefpub, idcfpub,* and idinpub identify, respectively, which individuals are in the Economic Family, Census Family and Individual publication universes. Certain variables that are normally simulated in SPSM are instead reported on the SCF. To allow replication of corresponding totals, new variables have been added to SPSD. These variables are *idscfoas, idscffa, idscfctc, idscfftc, idscfptc,* and *idscfuib.*

In the course of SPSD creation, records from the SCF are cloned and sometimes replaced.

Original information has been retained in this release, but requires the use of a specially created weight file that removes cloned households and re-instates replaced records. A number of imputations must also be un-done to replicate the SCF. These operations are illustrated in the include files /spsd/scfYY.cpi (where YY represents the database year) and /spsd/scfYY.api, which, if included in an SPSM run based on /spsd/baYY.cpr, will reproduce the SCF universe.

Due to confidentiality requirements, a small number of households with outlying characteristics had their province information randomized. As a result only Canada totals accurately reflect SCF values.

Appendix-A An Example Control Parameter File

This appendix contains a listing of the parameter file `/spsd/ba92.cpr`. Only a few of the SPSM facilities have been activated in this file.

```
###
## 2.1 Model Control Parameters
##     $Id: BA92.CPR 6.13 1997/10/23 16:47:20 gribble Exp gribble $
###


###
## 2.1.1 General information
###

CPRDESC      Tax/transfer:1992 Population:1992 Incomes:1992 (1992$)
LICENSEE     Statistics Canada    # SPSD/M licensee
AUTHOR                            # Name of person doing simulation
OUTCPR       ba92.cpr             # Name of control parameter file (out)
ALGDESC                           # Names of standard and alternate algorithms
 +---------+------------------------------+------------------------------+
 |Algorithm|          Standard            |           Alternate          |
 +---------+------------------------------+------------------------------+
 |call     | $Revision: 6.13 $     Oct 22/97 | $Revision: 6.13 $     Oct 22/97 |
 |drv      | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |ui       | $Revision: 6.13 $    Oct 22/97 | None                 Oct 22/97 |
 |fa       | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |oas      | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |dem      | Stub routine         Oct 22/97 | None                 Oct 22/97 |
 |txinet   | $Revision: 6.13 $    Oct 22/97 | None                 Oct 22/97 |
 |txccea   | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |gis      | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txitax   | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txhstr   | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txcalc   | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txhhexp  | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txprov   | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txnfld   | $Revision: 6.13 $    Oct 22/97 | None                 Oct 22/97 |
 |txpei    | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txns     | $Revision: 6.13 $     Oct 23/97 | None                 Oct 22/97 |
 |txnb     | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txque    | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txqinet  | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txqccea  | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txqitax  | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txqhstr  | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txqcalc  | Untitled             Oct 22/97 | None                 Oct 22/97 |
 |txont    | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txman    | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txsask   | $Revision: 6.13 $     Oct 23/97 | None                 Oct 22/97 |
 |txalta   | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txbc     | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |gist     | $Revision: 6.13 $    Oct 22/97 | None                 Oct 22/97 |
 |sa       | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txctc    | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |txfstc   | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |gai      | Stub routine         Oct 22/97 | None                 Oct 22/97 |
 |memo1    | $Revision: 6.13 $    Oct 22/97 | None                 Oct 22/97 |
 |ctmod    | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |memo2    | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 |cceopt   | $Revision: 6.13 $    Oct 22/97 | None                 Oct 22/97 |
 |mpc      | $Revision: 6.13 $     Oct 22/97 | None                 Oct 22/97 |
 +---------+-------------------------------+-------------------------------+
BRKFLAG             1             # Activate interception of `Break' key
BEEPFLAG            0             # Beep on completion
CLOSEFLAG           0             # Close window on completion
UPDATEINT        1000             # Interval between screen updates (hhlds)
ETAFLAG             1             # Activate fraction completed display
ROUNDFLAG           1             # Round disposable and consumable income
LOGFLAG             0             # Produce a .log file for this run
OUTLOG       ba92.log             # Name of log file (out)

###
## 2.1.2 SPSD input files
###

INPSPD       $SPSD/v60y92.spd     # Name of SPSD file (in)
FXVFLAG             1             # Read FAMEX expenditure vector file
INPFXV       $SPSD/v60y92.fxv     # Name of FAMEX vector file (in)
WGTFLAG             1             # Read weight file
INPWGT       $SPSD/v60y92.wgt     # Name of weight file (in)
REFFLAG             0             # Read reference results file
INPREF                            # Name of reference results file (in)
REFVARS                           # Reference results file variables

###
## 2.1.3 Database adjustment
###

AGENAME      Standard adjustment  # Name of database adjustment algorithm
INPAPR       $SPSD/ba92_92.apr    # Name of database adjustment parameter file (in)
OUTAPR                            # Name of database adjustment parameter file (out)

###
## 2.1.4 Variant Information
###

VARALG       Version 6.0 : 84-97  # Name of variant algorithm
VARMETH             2             # Method of creating variant variables
VARDESC      Current values for 1992
INPVARMPR    $SPSD/ba92.mpr       # Name of variant tax/transfer parameter file (in)
OUTVARMPR                         # Name of variant tax/transfer parameter file (out)
OUTMRSFLAG          0             # Variant results file creation flag
OUTVARMRS                         # Name of variant results file (out)
OUTMRSVARS   immicons             # Variant results file variables

###
## 2.1.5 Base Information
###

BASALG                            # Name of base algorithm
BASMETH             0             # Method of creating base variables
BASDESC      No Base              # Description of base parameters
INPBASMPR    $SPSD/ba92.mpr       # Name of base tax/transfer parameter file (in)
INPBASMRS                         # Name of base results file (in)
INPMRSVARS                        # Base results file variables

###
## 2.1.6 Subsampling, random number seed
###

SAMPLEREQ     1.000000000         # Size of sample requested
```

```
SAMPLE        1.000000000         # Size of sample obtained
WGTTOT       10605993             # Sum of weights on tax file
SEED           3                  # Random number generator seeds
          0
          1
          2

###
## 2.1.7 Record selection facility
###

SELFLAG             0             # Selection facility activation flag
SELUNIT             0             # Selection facility family level
SELSPEC                           # Selection specification
SELMAX              0             # Selection facility maximum # of households

###
## 2.1.8 Marginal Tax rate facility
###

MARFLAG             0             # Marginal tax rate facility activation flag
MARAMT         100.00             # Amount to be added to variable for marginal calculation
MARVAR       idiemp               # Variable to add MARAMT to
MARSPEC      idiemp>=1000         # Expression identifying recipients

###
## 2.1.9 Turning Point facility
###

TPFLAG              0             # Turning point facility activation flag
TPSPEC       1                    # Expression identifying individuals to change
TPVARS       idiemp               # Variables to modify
TPMETH              1             # Method for modifying variables
TPLL             0.00             # Lower limit for modified variables
TPUL         100000.00            # Upper limit for modified variables

###
## 2.1.10 User-defined Variables
###

UVARFLAG            0             # Activate UVAR parameter for expressions
UVAR                              # User statements

###
## 2.1.12 Text output facility
###

ASCFLAG             0             # Text output facility activation flag
OUTASC                            # Name of text output file (out)
ASCUNIT             0             # Text output family level
ASCSTYLE            1             # Style of text output
ASCDELIM                          # Field delimiter
ASCEXTPRC           0             # Number of digits of extra precision
ASCVARS                           # Variables selected for text output

###
## 2.1.13 SAS output facility
###

SASFLAG             0             # SAS output facility activation flag
OUTSAS                            # Name of SAS output file (out)
SASUNIT             0             # SAS output family level
SASVARS                           # Variables selected for SAS output
SASTITLE                          # SAS file label

###
## 2.1.14 Reports
###

OUTTBL       ba92.tbl             # Name of report file (out)
```

```
###
## 2.1.15 Parameter reporting
###

PRDFFLAG            0             # Parameter difference report activation flag

###
## 2.1.16 Tabular reporting
###


###
## 2.1.16.1 Built-in tables
###

T0FLAG              1             # Canada totals table flag (Dollars)
T0AFLAG             0             # Canada totals table flag (Units)
T1FLAG              1             # Provincial totals table flag (Dollars)
T1AFLAG             0             # Provincial totals table flag (Units)
T2FLAG              0             # Income group totals table flag (Dollars)
T2AFLAG             0             # Income group totals table flag (Units)
T3FLAG              0             # Family type totals table flag (Dollars)
T3AFLAG             0             # Family type totals table flag (Units)
T4FLAG              0             # LICO ratio group totals table flag (Dollars)
T4AFLAG             0             # LICO ratio group totals table flag (Units)
TABUNIT             2             # Built-in tables family level
TABDELTA        10.00             # Built-in tables winner/loser threshold
INCVAR       _immicons            # Variable to use for table 2
INCGP          9                  # Income cutpoints for table 2
       5000
      10000
      15000
      20000
      25000
      30000
      35000
      40000
      50000
PVRAT          9                  # Family poverty ratio fractions for table 4
     0.50
     0.75
     1.00
     1.25
     1.50
     2.00
     2.50
     3.00
     4.00

###
## 2.1.16.2 User-specified Tabulation facility
###

XTFLAG              0             # X-tab facility activation flag
XTSPEC                            # X-tab specification
XTDBLFLAG           1             # X-tab double precision activation flag
XTCOLS            132             # X-tab desired print width
XTLINES            66             # X-tab desired lines per page

###
## 2.1.16.3 Distributional Analysis Facility
###

DISTFLAG            0             # Distribution facility activation flag
DISTUNIT            0             # Distribution facility family level
DISTVAR                           # Distribution facility variable
DISTSAMP         3000             # Distribution facility sample size
DISTZERO            1             # Distribution facility zero inclusion flag
DISTP         13                  # Breakpoints for histogram plot
         1
         5
```

```
        10
        20
        30
        40
        50
        60
        70
        80
        90
        95
        99
DISTPWID           70             # Width of histogram plot
DISTPHGT           17             # Height of histogram plot

###
## 2.1.17 User-defined control parameters (if any)
###
```