STREAMS

User's Guide

Version 2.5 for Windows

Jan-Eric Gustafsson Per Arne Stahl

Göteborg University, Sweden

MultivariateWare
P. O. Box 300
SE-405 30 Göteborg
SWEDEN
Tel: +46 31 773 24 20
Fax: +46 31 773 20 70
www.mwstreams.com

Suggested bibliographic citation:

Gustafsson, J.-E., & Stahl, P. A. (2000). *STREAMS User's Guide. Version 2.5 for Windows*. Mölndal, Sweden: MultivariateWare.

ISBN: 91-972871-3-X

Printed by Elanders Digitaltryck AB, Angered, Sweden

Preface to Version 2.5

The current manual describes STREAMS Version 2.5, which also supports the Arbuckle and Wothke (1999) Amos 4 program. STREAMS takes advantage of the programmability features of Amos 4 and generates and interprets Amos Basic code. The user thus specifies the model in the MB language as usual and needs not write any Amos Basic code. STREAMS also has an interface to Amos Graphics so one-group MB models are automatically drawn. The model may then be edited and estimated in Amos Graphics. After the model has been edited in Amos Graphics it may be translated back to the MB language.

STREAMS 2.5 also offers a new user interface to the data management and data preparation functions. It is hoped that these new features in STREAMS will be appreciated, and both positive and negative feedback is solicited.

Mölndal May 22, 2000

Jan-Eric Gustafsson

tel: +46 31 773 24 20
fax: +46 31 773 24 62
e-mail: Jan-Eric.Gustafsson@ped.gu.se.

Address:

MultivariateWare
P. O. Box 300
SE-405 30 Göteborg
SWEDEN
www.mwstreams.com

Preface to Version 2.1

The current manual describes STREAMS Version 2.1, which also supports the Muthén and Muthén (1998) Mplus program. This program offers a wide range of new and exciting methods and techniques, which should be of particular interest to the advanced SEM user. It is hoped that the inclusion of Mplus in the STREAMS modeling environment will allow wider and easier access to the functionality of Mplus.

Mölndal July 18, 1999

Jan-Eric Gustafsson

Preface to Version 2.0

Structural equation modeling (SEM) is an invaluable tool for analyzing data involving multiple observations on a set of individuals, and particularly so when the observations are not perfectly valid and reliable. However, anyone who has applied SEM techniques to an actual large-scale set of data is also likely to have experienced some frustration, because these techniques are complex and computationally cumbersome. Thus, for a model involving many variables and groups of cases model-specification is tedious and error-prone, even with a sophisticated path diagram interface. Sometimes the iterative solution of the non-linear equations implied by the model fails to converge, or converges only slowly. And when the researcher is applying a complex and specialized approach to deal with a particular class of problems, such as multilevel or growth curve problems, these problems are aggravated by the increased complexity of model specification, and the need for optimal start values for the iterations.

Such frustrations were experienced very strongly when I, along with Ingrid Munck, Ingvar Lundberg, Monica Rosén, Anna Lindbom, Kjell Härnqvist and others, embarked on a project aiming to reanalyze the IEA (the International Association for the Evaluation of Educational Achievement) Reading Literacy (RL) study (Elley, 1994), using the multilevel latent variable modeling approach developed by Muthén (1989, 1990, 1994). Having applied these techniques in a previous study (Härnqvist, Gustafsson, Muthén, & Nelson Goff, 1994) we knew how to specify the model, but we stumbled at the practical problems of getting the specification (which typically amounted to a couple of thousands of lines of LISREL code) right, and of getting good enough start values for the iterations.

To get around these problems I wrote, in 1994, a pre-processor program, which combined a simple school-level model and a simple student-level model into a two-level model, and constructed the LISREL code, complete with start values. With this tool we managed to get some meaningful work done (e. g., Lundberg & Rosén, 1995; Munck, 1995), but most users experienced the program as unfriendly, and some even as being hostile, so it did not really encourge use. The next step, however, was to develop a full-fledged language, called MB (Model Building language), for describing one- and two-level models in one or more populations. Through the heroic efforts of Per Arne Stahl, the system was also put into the Windows environment to improve user friendliness. In further steps of development support for other types of complex data (i. e., structurally missing data) were added, as well as data handling facilities, and often Per Arne was a chief contributor of ideas and code. This tool was used, with quite interesting results, in the final phases of the reanalysis of the IEA data (e. g., Gustafsson, 1997, in press; Rosén, 1997, in press).

In June 1995 STREAMS 1.0 was released, This was a fairly simple program, which only supported LISREL, and it had its problems and limitations, so it was fairly rapidly replaced by a sequence of new releases. About a year later STREAMS 1.5 was released, and by that time it was obvious that the program does fill a great need. In January 1997, STREAMS 1.7, which supported Amos, EQS and LISREL, and which included a fairly rich set of utility routines. was published, and was distributed internationally by ProG-

AMMA in The Netherlands. Now STREAMS 2.0, which is a 32-bit version with considerably extended capabilities (among others, a path diagram interface, and support for the Mx program) is being released, and I hope that it will prove to be even more useful than its predecessors.

The STREAMS system thus has been created in an attempt to improve the usefulness of structural equation modeling for the kinds of data and problems that are encountered in large-scale educational research. However, the system should be useful in many other areas of social and behavioral research as well. There is, however, room for further improvements and extensions of the system, and I hope to see a further rapid development through what has proven to be a main source of improvements, namely suggestions from users for how to augment the usefulness of the system.

Mölndal January 31, 1999

Jan-Eric Gustafsson
tel: +46 31 773 24 20
fax: +46 31 773 24 62
e-mail: Jan-Eric.Gustafsson@ped.gu.se.

Address:

MultivariateWare
P. O. Box 300
SE-405 30 Göteborg
SWEDEN
www.ped.gu.se/mw

Some Practical Information

The present manual describes STREAMS 2.5, which supports Amos, EQS, LISREL® 8, Mplus and Mx. Observe that STREAMS will not work unless Amos 3.5-4.0, EQS 4 or 5, LISREL® 8.03-8.30, Mplus 1.0, and/or Mx 1.44- is installed on the computer.

Amos, EQS, LISREL and STREAMS may all be ordered from:
ProGamma bv
P. O. Box 841
9700 AV Groningen
The Netherlands
Tel: +31 503 63 6900
Fax: +31 503 63 6687
www.gamma.rug.nl

The Mx program, which has been authored by Michael Neale at Virginia Commonwealth University, may be downloaded without cost from *http://griffin.vcu.edu/mx.*

Amos may be ordered from:
SmallWaters Corporation
1507 E. 53rd Street, #452
Chicago, IL 60615
Tel +1 773 667 8635
Fax +1 773 955 6252
www.smallwaters.com

EQS may be ordered from:
Multivariate Software, Inc.
4924 Balboa Blvd., # 368
Encino, CA 9136
Tel +1 818 906 0740
Fax +1 818 906 8205
www.mvsoft.com

LISREL may be ordered from:
Scientific Software International, Inc.
1525 East 53rd Street, Suite 530
Chicago, IL 60615-4530
Tel: +1 312 684 4920
Fax: +1 312 684 4979
www.ssicentral.com

Mplus may be ordered from:
Muthén & Muthén
11965 Venice Blvd., Suite 407
Los Angeles, CA 90066
Fax: (310) 391-8971
www.statmodel.com

1 Introduction

Structural equation modeling (SEM) is a versatile and powerful statistical tool, which has proven useful for analyzing a wide range of phenomena within many disciplines (see textbook presentations by, e. g., Bollen, 1989; Byrne, 1994; Hayduk, 1987; Hoyle, 1994; Jöreskog & Sörbom, 1993c; Loehlin, 1992; Maruyama, 1998; Schumacker & Lomax, 1996). SEM thus is a general statistical method, which includes many other methods as special cases (e. g., regression analysis, path analysis, factor analysis, simultaneous equations, ANOVA, MANOVA), but it also goes far beyond these. Above all, SEM gains its strength from the idea of latent variables.

Several excellent computer programs, each of which has its unique strengths, are available for specifying and estimating structural equation models. The power and flexibility of the approach, along with the differences among the programs may, however, make SEM seem inaccessible to novices, and it may be forbiddingly complex even for experienced users. STREAMS (Structural Equation Modeling Made Simple) has been developed with the purpose of making SEM easily available to a broad range of users. Both novices and experienced SEM users will find STREAMS useful, because the program offers:

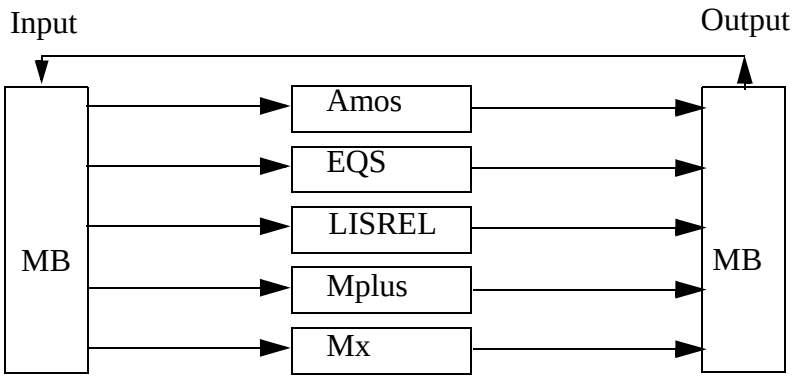
- *One language and one interface for five SEM Programs.* STREAMS offers a consistent interface to four SEM programs (Amos, EQS, LISREL, Mplus and Mx), with generation of model statements, control of program execution and post-processing of program output.
- *User-friendly tools for model specification.* Model specification may be done with a simple and powerful language and/or through path diagrams. The user interface is equipped with numerous tools designed to support specification, estimation, and interpretation of models.
- *Automatic data setup.* STREAMS takes care of the minute details of data specification and setup for all the estimation programs, relieving the user the burden of dealing with program specific rules and conventions for data management.

- *Models for complex data*. Easy specification of complex models for multiple groups of cases, multi-level data and incomplete data.
- *Efficiency.* In STREAMS start values are automatically copied from previously estimated models, which usually yields substantial improvements in program execution time. It is also trivial to switch between different estimation programs, so as to take advantage of their respective strengths.
- *Data management tools*. STREAMS performs preparation of data for modeling, including computation of matrices for two-level analysis and structurally missing data.
- *Project management*. Efficient functions for management of data and models are included in the system.

The present manual presents the basic procedures and functions of the system.

Basic Ideas of STREAMS

STREAMS is based on two fundamental ideas: the idea of representing models in a generic system, and the idea of assembling both models and data within a project. Schematically, the process of model representation and model estimation may be represented in the following way:



The model is specified in terms of a general meta-language for SEM (the Model Building language, MB), and for this process STREAMS offers a large set of tools and functions. When the model is to be estimated it is translated into the language of the selected estimation program. Often the model specification may be furnished with good starting values from one or more previously estimated models. When the model has been estimated, the output is translated back into the MB language, and the model parameters are stored for future use.

STREAMS thus relieves the user of the burden of having to remember syntactical details of one or more program specific modeling lagnuages, and of generating large amounts of complex, error-prone, instructions. However, what is often equally important is that STREAMS takes care of all the details of data specification (e. g., file management, variable labels, missing data codes, and so on). This can be done because different data sets (typically either in the form of covariance matrices or in the form of raw data files) are

stored in a data base, which also contains meta-information (e. g., variables, code labels, and number of cases) about the data sets. This allows STREAMS to prepare both the data and the description of the data in the ways expected by the different SEM programs.

The combination of powerful modeling functions with sophisticated data handling facilities makes STREAMS a powerful data analytic environment for both advanced and novice users.

Functions of STREAMS

STREAMS presents users of the SEM technique with an environment for specifying, estimating, and evaluating models:

- *Model specification* is done with the MB language and/or through construction of path diagrams with the Amos 4 AmosGraphics system. This Amos system (which requires Amos 4 to be installed) offers advanced editing and presentation tools. A model specified in terms of the MB language can be transformed into a path diagram, and a model drawn as a path diagram is automatically translated into MB statements. The user can thus move between the MB language and the path diagram representation. The MB language is simple and powerful, and particularly so for specifying models involving many variables and/or groups. The language also has extensions for specifying two-level models, and models for incomplete data. For model editing, the system also offers a large set of tools and functions, such as procedures for combining several models into one, and for imposing different kinds of constraints.
- *Model estimation* is done in a three-step process. In the first step (the pre-processor step) the MB language statements are translated into instructions for one of several different SEM estimation programs. The instructions are also typically supplied with start values from one or more previously estimated models. The current version of STREAMS (2.5) supports Amos (3.51, 3.6 and 4.0), EQS (4 and 5, along with an experimental interface to the not yet released version 6 of EQS), LISREL (8.03-8.30), Mplus 1.0 and Mx (1.44-). In the second step (the estimation step) the chosen estimation program is run, and in the third step (the post-processor step) the output from the estimation program is translated back into the MB language. These three steps are transpararent to the user.
- *Model evaluation* involves scrutiny of the results produced in the post-processor step, which may be presented both in the form of listings, as a path diagram, or as charts. Sometimes there is also reason to consult the output listing from the estimation program, which is also easily available. Model fit statistics are also available for scrutiny, as is diagnostic information about reasons for model misfit. Often this information results in a decision to respecify the model, in which case the process of model specification, estimation, and evaluation is repeated once again.

STREAMS also includes functions for preparing data for modeling, such as:

- Computation of covariance matrices from raw data for one or more groups of cases.
- Computation of matrices for two-level analysis, which is useful when the observational units (e. g., individuals) are clustered (e. g., into schools).

- Computation of matrices for analysis of systematically missing data, which is useful, for example, when the data to be analyzed has been generated by a matrix sampling design.
- Preparation of raw data for use in SEM programs which require raw data input.
- Preparation for analysis of matrices computed outside the STREAMS system.

The use of these functions, and other basic STREAMS functions, is documented and described in the following chapters of this guide.

Overview of the User's Guide

Chapters 2 and 3 introduce, in the form of a quick-start guide, the basic functions of the system, as well as the basic principles of the MB lanuguage. A user with some previous knowledge of structural equation modeling should be able to use STREAMS after having studied these chapters.

Chapters 4 to 6 present how to use STREAMS for specifying and estimating more complex models, such as multiple-group models (Chapter 4), missing-data models (Chapter 5), and two-level models (Chapter 6). These chapters treat specialized issues of rather great complexity, so novice users are adviced to skip these until the need arises. Chapter 7 discusses how to improve the possibility of obtaining estimates, and how the functions for copying start values may be used to avoid non-convergence in model estimation,

Chapters 8 to 11 present the tools available for data and project management. Chapter 8 describes the construction of the projects in STREAMS and how data should be prepared before starting a modeling project. Chapter 9 presents how to use the built-in function for computing covariance matrices, and Chapter 10 the functions for importing matrices. Chapter 11 describes the tool available for maintaining projects.

Chapters 12 and 13 are reference chapters which provide more complete presentations of different aspects of the system. Chapter 12 presents the MB language and Chapter 13 the interface to the different estimation programs.

Chapter 14, finally, describes how to install STREAMS, and how to connect STREAMS to the estimation programs.

Part 1

Specifying and Estimating Models

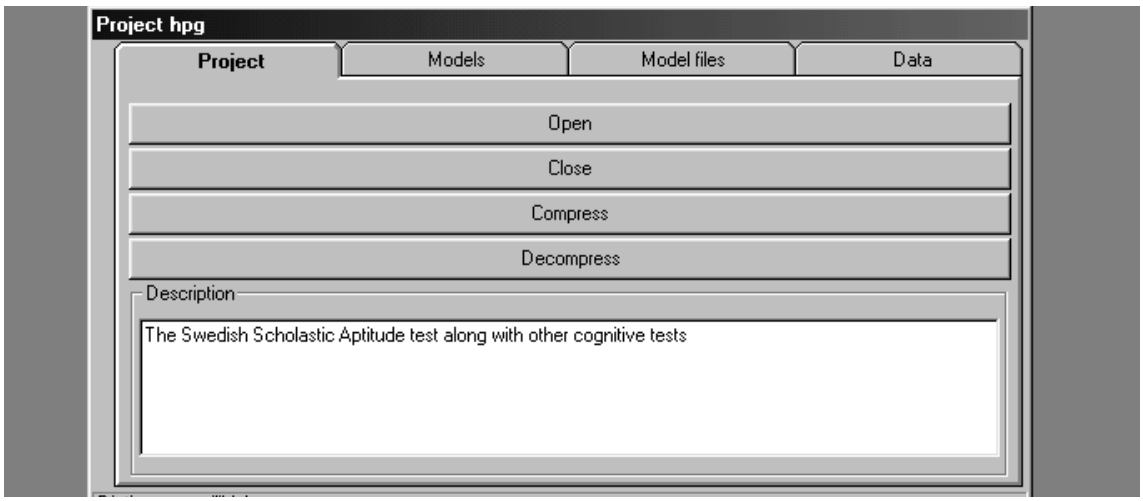
The first part of the User's Guide presents the how to specify and estimate structural equation models with STREAMS. Chapters 2 and 3 are intended to serve as a quick start guide for how to use the program. Chapters 4 to 6 deal with specification of specialized types of models, and novice users are adviced to skip these chapters. Chapter 7 is devoted to a short discussion of issues of efficiency in using structural equation modeling techniques. All chapters also present concrete examples of models.

2 Using Projects and Estimating Models

This chapter introduces one fundamental part of STREAMS, namely the *project*. It is shown how projects in STREAMS may be opened and inspected, and how models within an existing project are estimated. The basics of the MB language are also introduced.

The STREAMS Project

For most operations in STREAMS the *project* plays an essential role. A STREAMS project may be described as a collection of related datasets (raw data and/or matrices) and models. Within the project the datasets are organized into *project folders*. The project also stores information about the matrices and data files (e. g., number of cases, variable names, code labels, missing data codes, and so on) in a *project dictionary*. Most actions related to projects are done with the *Project* window, which is always presented when STREAMS is started and which is always available on the desktop:



The *Project* window has four tabs. One of these is the **Project** tab, which is used to open and close projects. The **Models** tab is used to open previously created models, and the **Model files** tab to manage the files that are associated with different models. The **Data** tab offers functions for managing data, such as computing matrices and importing data into the project.

Opening and Inspecting an Existing Project

Almost all STREAMS activities require an *active* or *open* project. If a project has been created previously (see Chapter 8) it must be opened to make the information in the project available. For purposes of illustration we will be using one of the projects (*hpg*) in which are supplied with the system. The project includes data from a study of 579 12th grade students who had taken the Swedish Scholastic Aptitude Test (SweSAT; "Hög-skoleprovet"), along with some other tests. The present version of SweSAT consists of 6 subtests which measure both verbal and non-verbal abilities, the capacity to make use of information, and general knowledge:

- *Vocabulary* (WORD), which measures understanding of words and concepts.
- *Data Sufficiency* (DS), which aims to measure numerical reasoning ability.
- *Reading Comprehension* (READ), which measures reading comprehension in a wide sense.
- *Diagrams, Tables and Maps* (DTM), which is a problem-solving test with information presented in tables, graphs, and/or maps.
- *General Information* (GI), which measures knowledge and information from many different areas.
- *English Reading Comprehension* (ERC), which measures reading comprehension in English

In addition, four standard psychometric tests were administered:

- *Synonyms* (SYNONY), which is a vocabulary test.
- *Visualization* (VIZUAL), which is a spatial visualization test.
- *Figural Reasoning* (FIGRES), which is a non-verbal reasoning test.
- *Number Series* (NUMSER), which is an inductive reasoning test with numerical content.

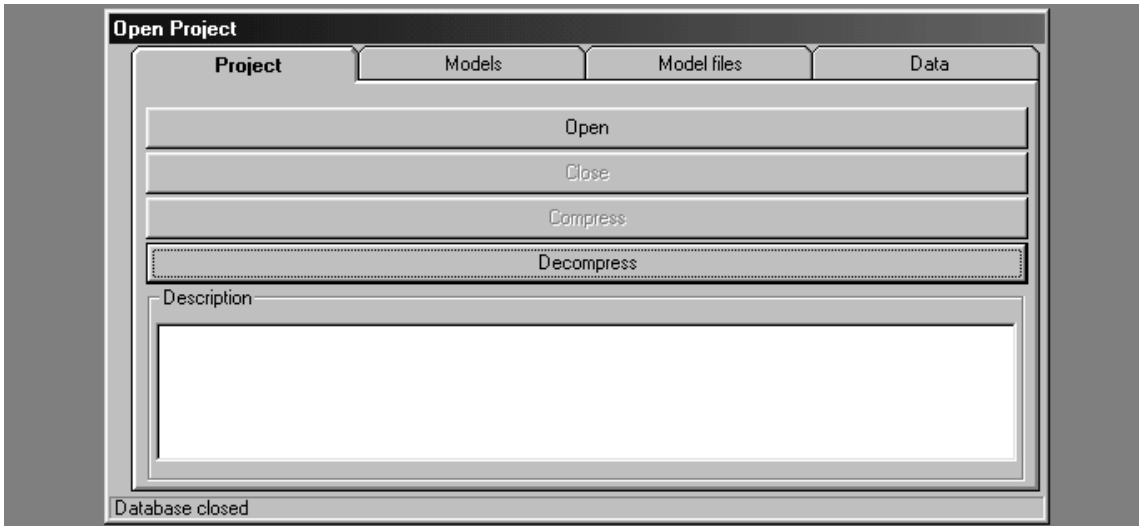
There is also information about student gender (GENDER), and program of study: the Humanistic (HUM), Science (SCI), Social (SOC), Economic (ECO;), or Technical (TEC) program. The variables indicating program belongingness are dummy variables, with the Economic program taken to be the reference group. The data set also includes a variable (MRK) which represents the mean grade awarded when leaving grade 12.

Decompressing the project

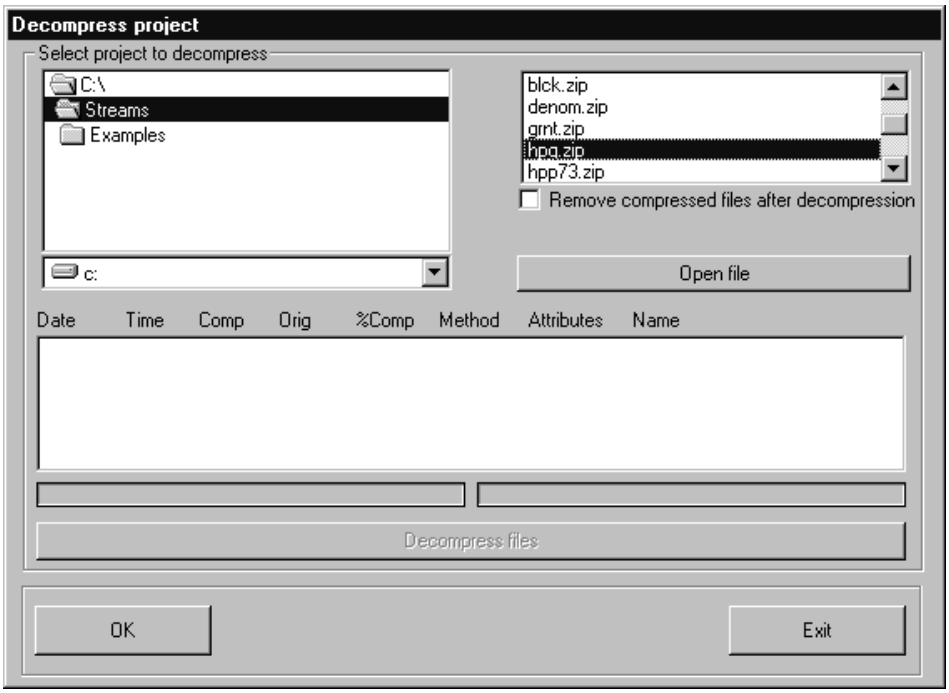
When STREAMS is installed all the examples projects discussed in this guide are copied, in compressed form, into the STREAMS installation directory (usually *C:\STREAMS*, but this may be changed during the installation). These compressed files have the name of the project (e. g., *hpg*), and *.zip* as suffix. The compression or "packing" of a project implies that all the files and directories associated with a project are stored in compact format in one file (an "archive"). The STREAMS archives are stored in zip 2.04g-format, and can

be opened with the WinZip program, but this program is not needed because STREAMS has its own routines for compressing and decompressing projects. Apply the following steps to use the STREAMS procedure to decompress the *hpg* project:

- Start the STREAMS program.
- If the *Project* window is not shown, make it appear through clicking the **F9** function key.
- Click the **Project** tab on the *Project* window:

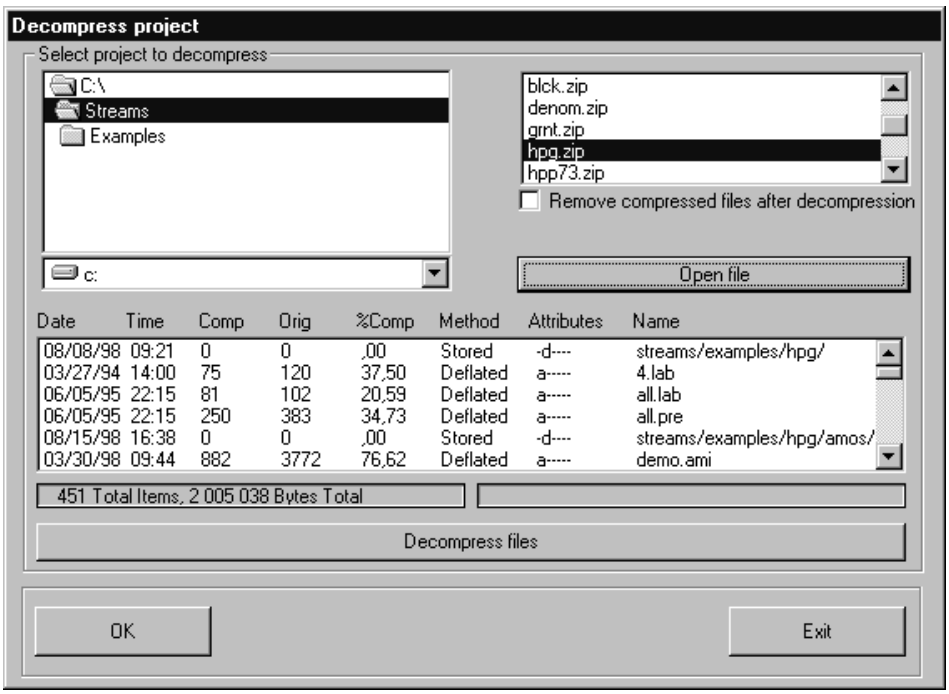


- Click the **Decompress** button, which produces the *Decompress project* form:



- Use the dialogue box on the left side to select the STREAMS installation directory (here *c:\STREAMS;* if the *hpg.zip* file is not in this directory it may be downloaded from *www.mwstreams.com*). This causes the compressed project files to be displayed

in the list-box to the right. Select the *hpg.zip* file and click the **Open file** button (or double-click the file name). This opens the archive and presents the contents in the large list-box:

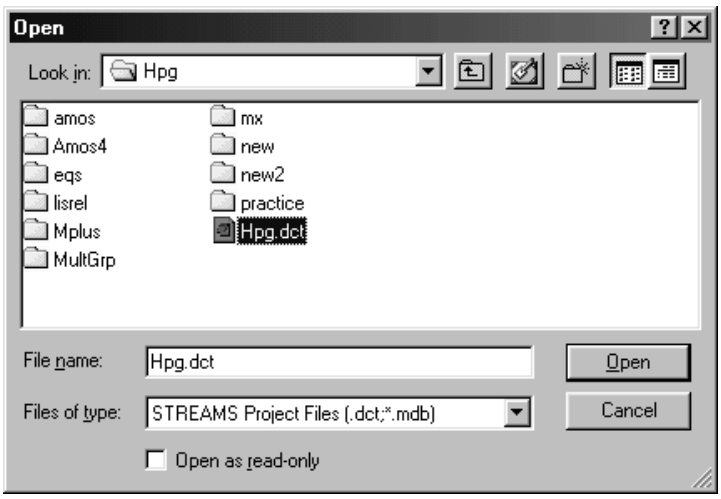


- To actually copy the files from the archive into the project the **Decompress files** button must be clicked. This restores the project, *using the pathname that the project had when it was compressed*. In this case the project will be restored to *C:\STREAMS\EXAMPLES\HPG*.

Opening the Project

After the *HPG* project has been decompressed it must be opened, which is done in the following way:

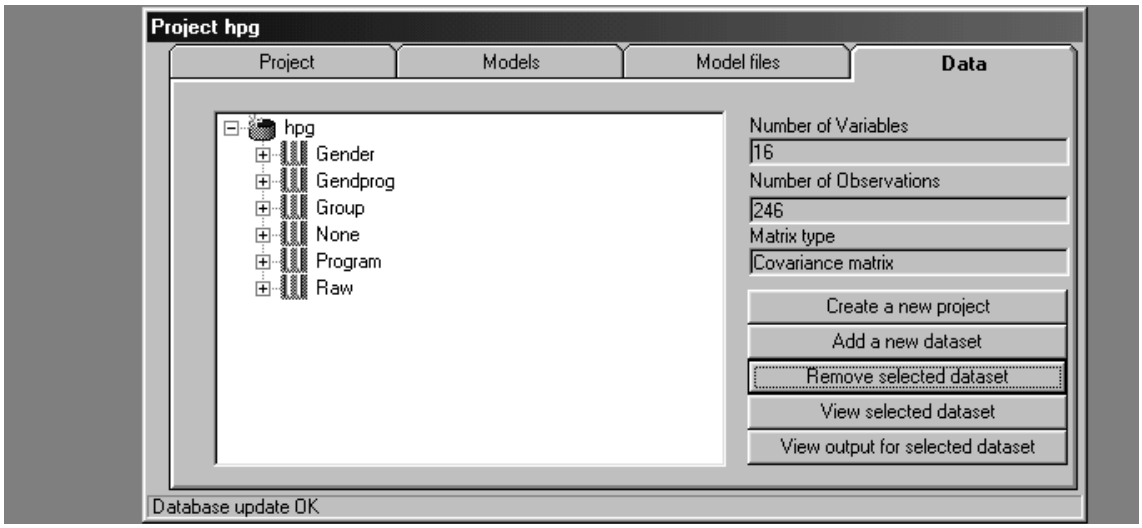
- Start the STREAMS program.
- If the *Project* window is not shown, make it appear through clicking the **F9** function key.
- Click the **Project** tab on the *Project* window.
- Click the **Open** button. This presents the standard open dialogue form. Use this form to locate either the file *hpg.dct* or the file *hpg.mdp* in the *STREAMS\EXAMPLES\HPG* directory. The *.dct* suffix is used for dictionary files created with STREAMS versions up to 2.1, while the *.mdp* suffix is used for dictionary files created with STREAMS 2.5-. When a *.dct* dictionary is opened with STREAMS 2.5 it is automatically converted into an *.mdp* dictionary. The original *.hpg* project is automatically compressed for backup purposes, but it should be observed that an *.mdp* dictionary cannot be converted back to a *.dct* dictionary.



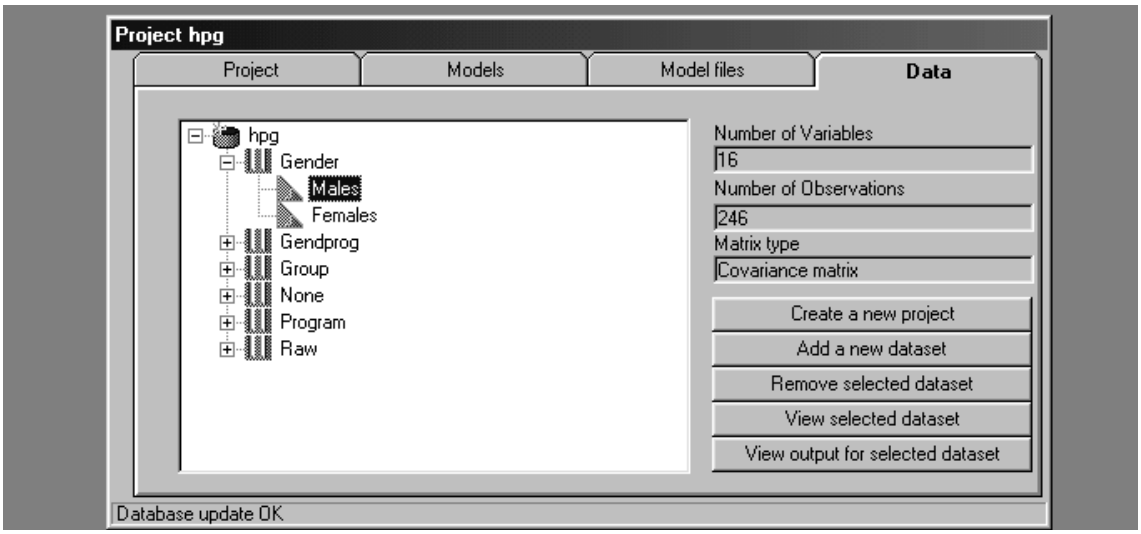
When the **Open** button is clicked the *hpg* project is opened.

Getting Information about the Project

More specific information about the data in the project is obtained if the **Data** tab of the *Project* window is clicked:



There are several *datasets* (e. g., covariance matrices, correlation matrices, or raw data-sets) in the project, which are subsumed under several *folders.* Folders are identified by the icon , which signifies an *open folder* to which additional datasets may be included, or by the icon which signifies a *closed folder* to which no new datasets may be added. This is a simple way of organizing datasets in a two-level hierarchical system. For example, when a dataset is split up in several disjoint subsets they may naturally be organized in a folder. The folder is assigned by the user when the matrix is computed or the data set is imported into the project. When the + sign next to the folder icon is clicked, the **Dataset labels** for the datasets subsumed under the folder are shown, next to an icon identifying the nature of the data set (e. g., covariance matrix: , raw data file:). The first folder is *Gender:*



When the + sign next to Gender is clicked it is seen that in this folder, there are two covariance matrices, one with the dataset label *Males* and the other with the dataset label *Females.*

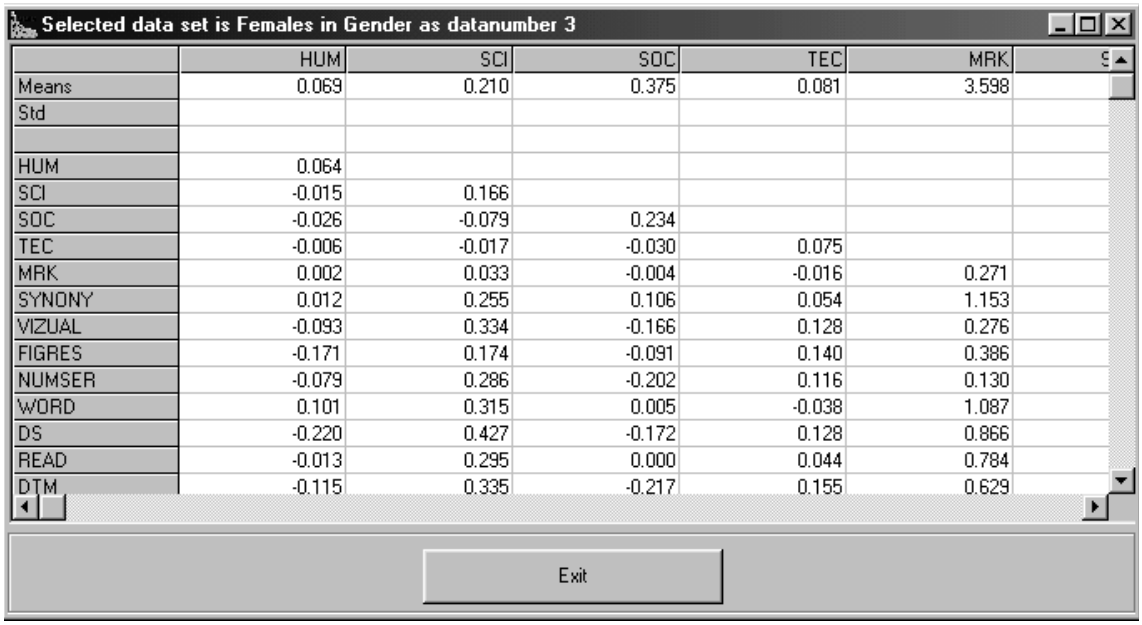
A particular set of data may be selected through clicking on the icon or the dataset label, which gives information about the number of observations and variables. The matrix for *Males* thus encompasses 16 variables and it is computed on 246 observations; for *Females* the number of observations is 333. In the *Program* folder there are 5 matrices (*Eco*, *Hum*, *Sci*, *Soc* and *Tec*) which represent different lines of study. The *Gendprog* folder includes 9 matrices, where the total sample has been divided by gender and program (there were too few males in the *Hum* program to compute a matrix; otherwise there would, of course, have been 10 covariance matrices).

This way of organizing the data is quite useful when dealing with multiple groups of cases. The folder and the dataset label is assigned when the matrix is computed or imported (see Chapters 8 to 10).

A STREAMS project may include several kinds of data. A dataset label may, thus, refer to, among other things, a:

- Covariance matrix and a mean vector.
- Polychoric correlation matrix.
- Covariance matrix with either a diagonal or a symmetric weight matrix.
- Polychoric correlation matrix with either a diagonal or a symmetric weight matrix.
- Raw data matrix.

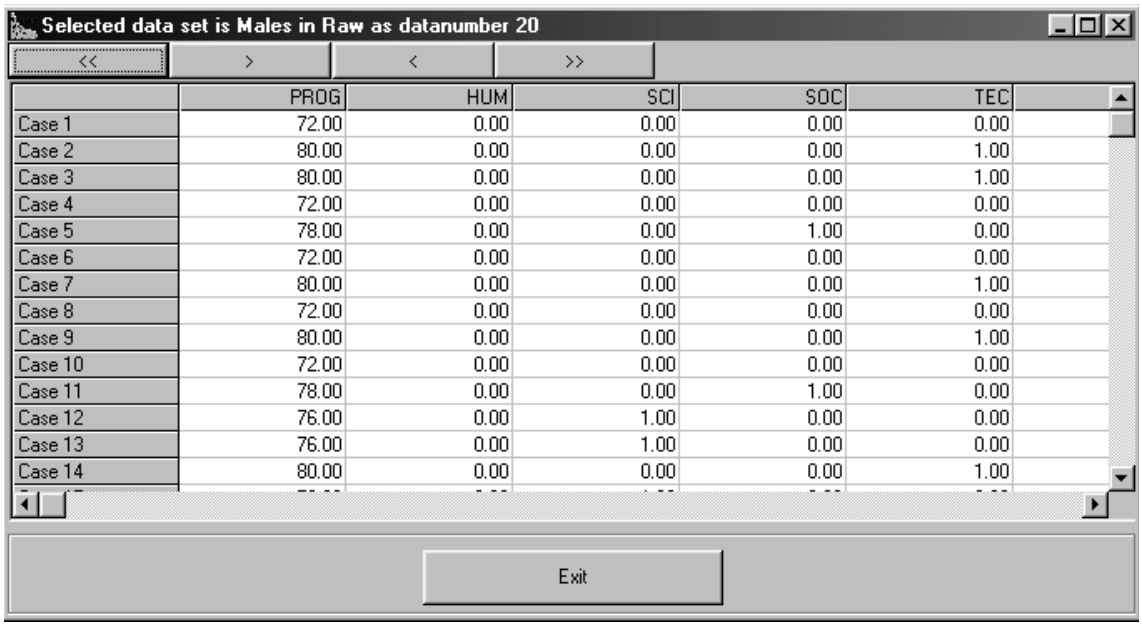
In the present case the **Matrix type** field says *Covariance matrix*, and if the **View selected dataset** button is clicked for, e. g., the *females*, or the icon for *females* is double-clicked, the matrix is displayed:



	HUM	SCI	SOC	TEC	MRK
Means	0.069	0.210	0.375	0.081	3.598
Std					
HUM	0.064				
SCI	-0.015	0.166			
SOC	-0.026	-0.079	0.234		
TEC	-0.006	-0.017	-0.030	0.075	
MRK	0.002	0.033	-0.004	-0.016	0.271
SYNONY	0.012	0.255	0.106	0.054	1.153
VIZUAL	-0.093	0.334	-0.166	0.128	0.276
FIGRES	-0.171	0.174	-0.091	0.140	0.386
NUMSER	-0.079	0.286	-0.202	0.116	0.130
WORD	0.101	0.315	0.005	-0.038	1.087
DS	-0.220	0.427	-0.172	0.128	0.866
READ	-0.013	0.295	0.000	0.044	0.784
DTM	-0.115	0.335	-0.217	0.155	0.629

When the **Exit** button is clicked, the form presenting the covariance matrix is closed and the *Project* window appears again.

Often the data in a project are stored in the form of a symmetric covariance matrix, but frequently it is also necessary to store raw data in the project. Such data are stored in a rectangular matrix with the observations (individuals) as rows, and the variables as columns. In the *Raw* folder there are three sets of data (*Males*, *Females* and *Tot*). If the *Males* dataset label is selected and the **View selected dataset** button is clicked, the raw data is displayed:



	PROG	HUM	SCI	SOC	TEC
Case 1	72.00	0.00	0.00	0.00	0.00
Case 2	80.00	0.00	0.00	0.00	1.00
Case 3	80.00	0.00	0.00	0.00	1.00
Case 4	72.00	0.00	0.00	0.00	0.00
Case 5	78.00	0.00	0.00	1.00	0.00
Case 6	72.00	0.00	0.00	0.00	0.00
Case 7	80.00	0.00	0.00	0.00	1.00
Case 8	72.00	0.00	0.00	0.00	0.00
Case 9	80.00	0.00	0.00	0.00	1.00
Case 10	72.00	0.00	0.00	0.00	0.00
Case 11	78.00	0.00	0.00	1.00	0.00
Case 12	76.00	0.00	1.00	0.00	0.00
Case 13	76.00	0.00	1.00	0.00	0.00
Case 14	80.00	0.00	0.00	0.00	1.00

The form offers different ways of scrolling. The horisontal scroll bar may be used to select

other variables, and the arrows in the upper-left corner of the form may be used to scroll among the observations. Clicking the **Exit** button returns control to the *Project* window again.

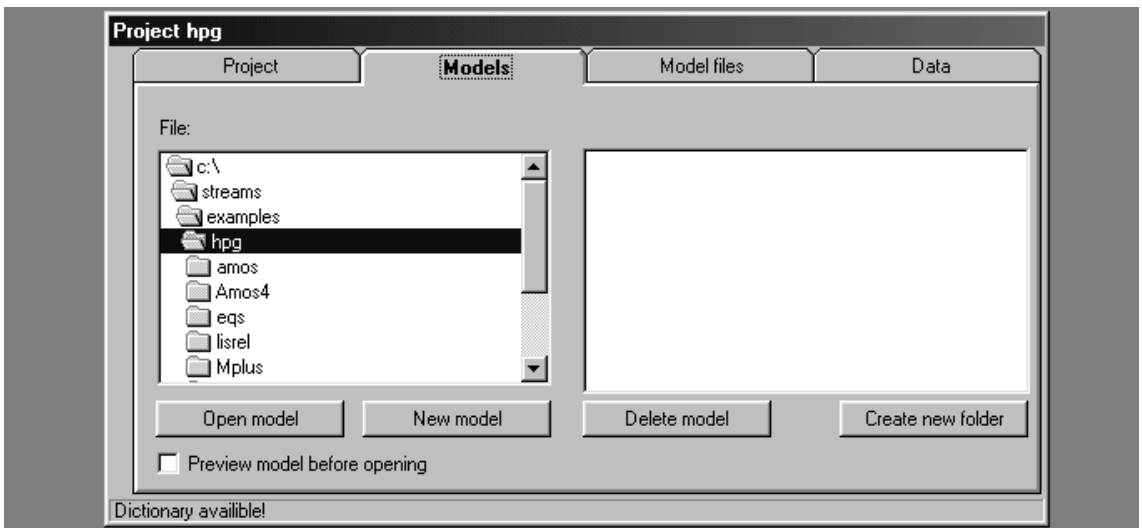
The **Data** tab offers several functions for adding data to the project, and for removing data, but we will not go inte these here (see instead Chapters 8-10). In addition to data, however, a project also includes models, and we will take a closer look at one of these.

Opening and Estimating a Previously Created Model

A STREAMS project not only includes data, but typically there also are models as well. These models are specified in terms of the MB (Model Building) language which is a part of the STREAMS system. We will first demonstrate how an existing model can be opened, inspected and estimated.

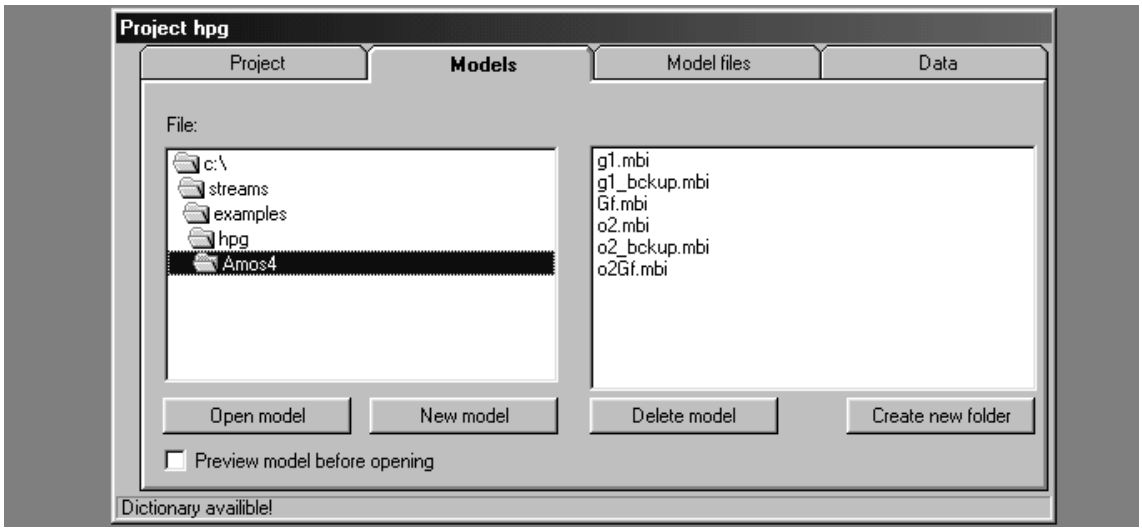
Opening the Model

If the *HPG* project has not already been opened, this should first be done, using the procedure described above. When the project has been opened, the **Models** tab on the *Project* window should be clicked:

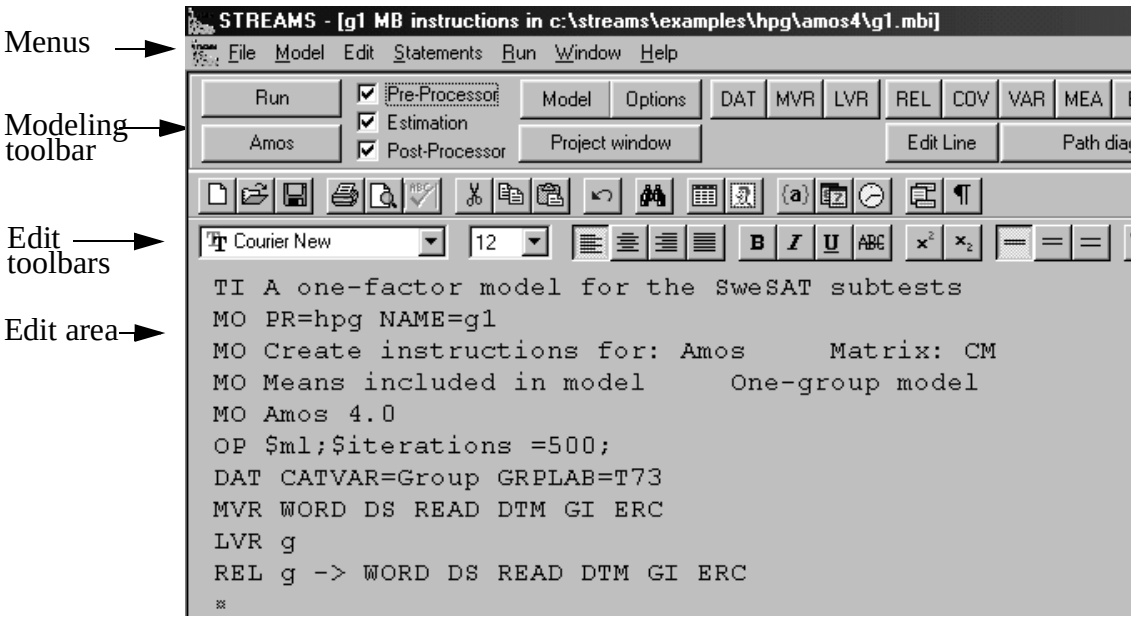


The *Project* window presents a list of directories, and there is also a list-box where a list of model files, which have the suffix *.mbi*, may be shown. Here the list is empty, because all the model files reside in subdirectories under the *HPG* directory, there being one directory for each estimation program and a few others as well. Any number of subdirectories may be created under the directory in which the project dictionary is stored.

For purposes of illustration the Amos 4 program will be used, so the Amos4 folder under the HPG folder should be double-clicked.



A particular model may be opened, either through selecting it from the list, and clicking the button labeled **Open model**, or through double clicking the model name. One of the models is *g1.mbi*, and when this model is opened a set of statements is displayed in the *Model Building* window:



The actual appearance of the screen may be somewhat different from what is shown here, because a choice may be made which editor tools are to be presented (this is done with the **Preferences** option under the **File** menu, but is of no importance here). The size of the *Model Building* window may be increased so that it fills the entire screen through clicking the Maximize button in the upper right corner of the window. This will make the *Project* window, and possibly also other open *Model Building* windows disappear. These other windows may, however, be brought in the foreground with the **Windows** menu. The *Project window* can also always be produced by clicking the **Project window** button on the modeling toolbar, or through using the **F9** function key.

At the top of the *Model Building* window there is a set of drop-down menus (e. g., **File**, **Model**, **Edit**, ...), which, among other things, include tools for opening, specifying, editing, estimating and inspecting models. Beneath is a set of buttons on the Modeling toolbar, with the **Run** button in the left-most position, and the **SCL** button in the right-most position. These buttons provide short-cuts to functions which are available in the menus as well, and most of these are also available through the function keys on the keyboard. The buttons on the Modeling toolbar are used for specifying and editing models. Beneath the Modeling toolbar are the Edit toolbars, which include standard text-editing tools, which also may be used for editing model statements. Below the toolbars is the edit area, where the model statements are displayed.

The model is defined by a series of statements, which are formulated in the MB language, which is presented in full in Chapter 12. Some statements concern basic aspects of the model, such as its name and description, which estimation program is to be run, which matrix is to be analyzed, and so on. The following three lines define the actual model through specifying the manifest and latent variables and their relations:

```
MVR WORD READ GI DS DTM ERC
LVR g
REL g -> WORD READ GI DS DTM ERC
```

With the MVR statement the six manifest variables in the model are declared, and the LVR statement declares the single latent variable (*g*) of the model. The *g1.mbi* model thus specifies a simple one-factor model.

The MB language includes four basic statements: REL, VAR, COV and MEA. Through applying these statements on four different kinds of variables (manifest, latent, and residuals in manifest and latent variables) almost any structural equation model can be specified with ease.

Except for the declaration statements MVR and LVR, the *g1.mbi* model is defined by the single REL statement. This statement says that the 6 manifest variables are influenced by (or are indicators of) the single latent variable *g*. By default the program also assigns a residual to each dependent variable, and assumes that each independent variable has a variance. Thus, to specify the model a single REL statement suffices.

The MB statements can be constructed in several different ways, such as through the text editor, or with a path diagram editor, or through clicking buttons and filling in forms, or through combinations of these methods. Chapter 3 describes the different functions available for specifying and editing models.

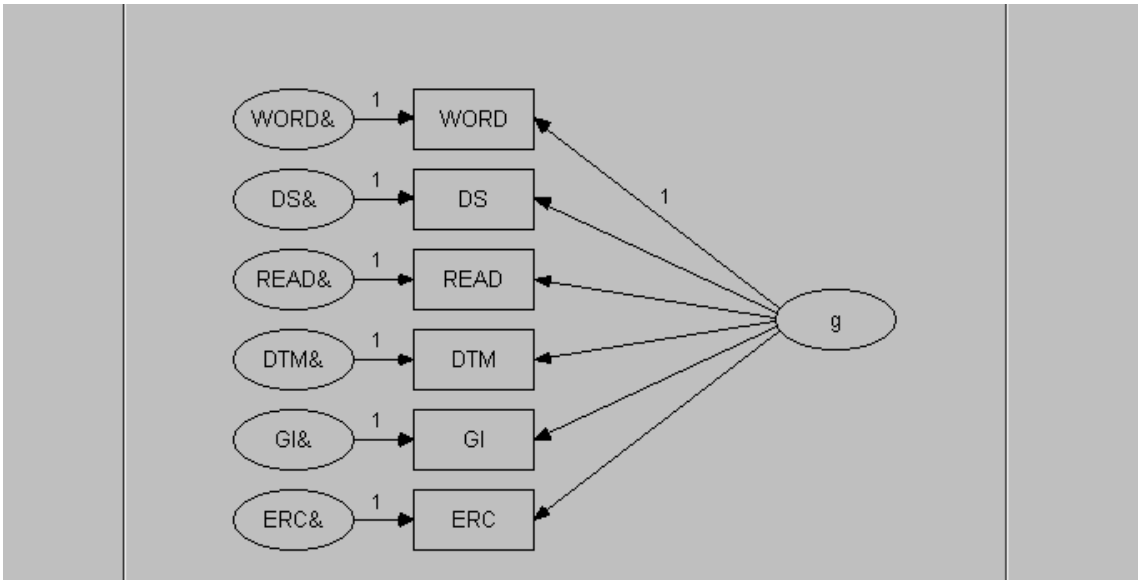
We will experiment with this model, and the very first thing to be done is to change the name of the model, so that the *g1.mbi* file remains unchanged. The change of name can be accomplished in several different ways, but the easiest method is to just move the cursor to the line

```
MO PR=hpg NAME=g1
```

and edit the old name (*g1*) into a new name. Another way to do this is to click the **Model** button on the modeling toolbar, and change the **Model name** field on the **Model Description** tab. Here the new name *g1n* has been chosen, but it is recommended that users who follow these steps select another name.

In the upper right part of the window is a button with the label **Path diagram**, which is enabled if Amos 4 is used as the estimation program. If this button is clicked, the model

is represented as a path diagram in Amos Graphics:



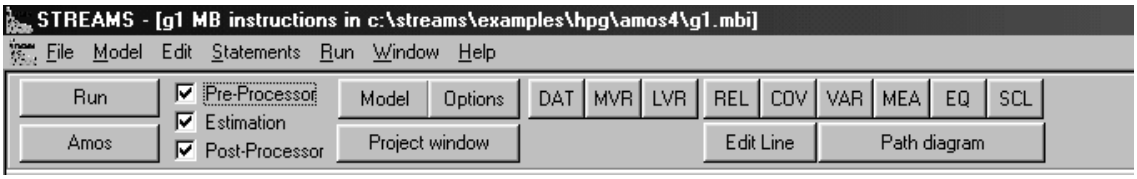
The diagram displays the manifest variables as squares, and the latent variable as a cricle. For each manifest variable, which here are dependent variables, there is a residual, which has the same label as the manifest variable, but with an ampersand added (e. g., WORD&). This is a general principle of the MB language (see Chapter 12).

The path diagram may be edited so that it attains a more aesthetically pleasing appearance, and it may also be edited in such a way that the model itself is changed. Manifest and latent variables may, for example, be added and deleted; relations and covariances may be added; and constraints of different kinds may be imposed. The methods for doing that are described in Chapters 3 and 12.

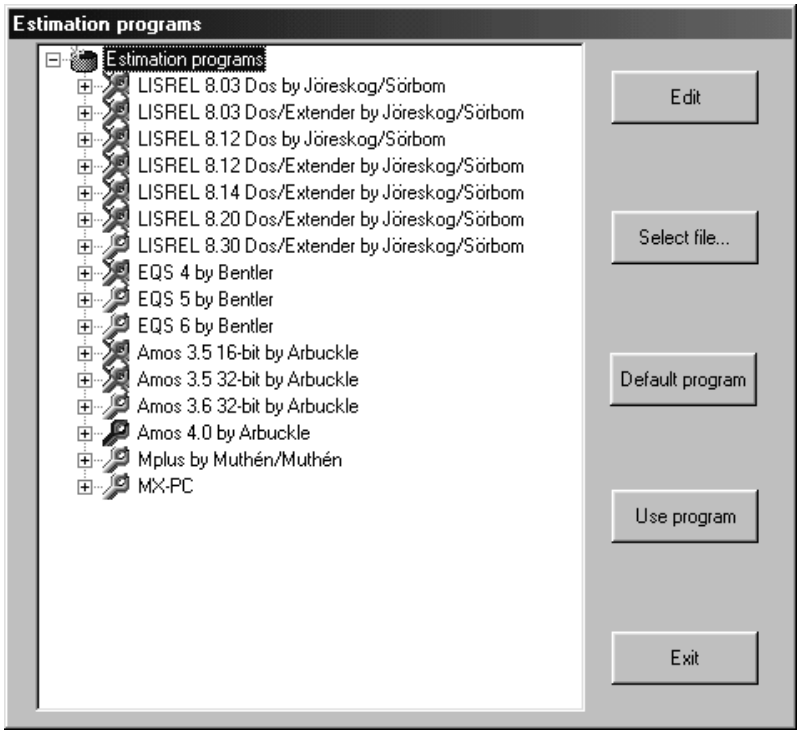
Let us assume, however, that we are satisfied with the model. We may now click the **Model Building** button in the upper right corner of the window to go into MB mode, and run (i. e., estimate) the model from there.

Estimating the Model

Let us go through the sequence of estimating the model. The estimation program selected is Amos, which is shown on the button beneath the **Run** button:



If other estimation programs (e. g., EQS, LISREL, Mplus or Mx) are installed they may be selected through clicking the Amos button. This produces a form which presents the available estimation programs:

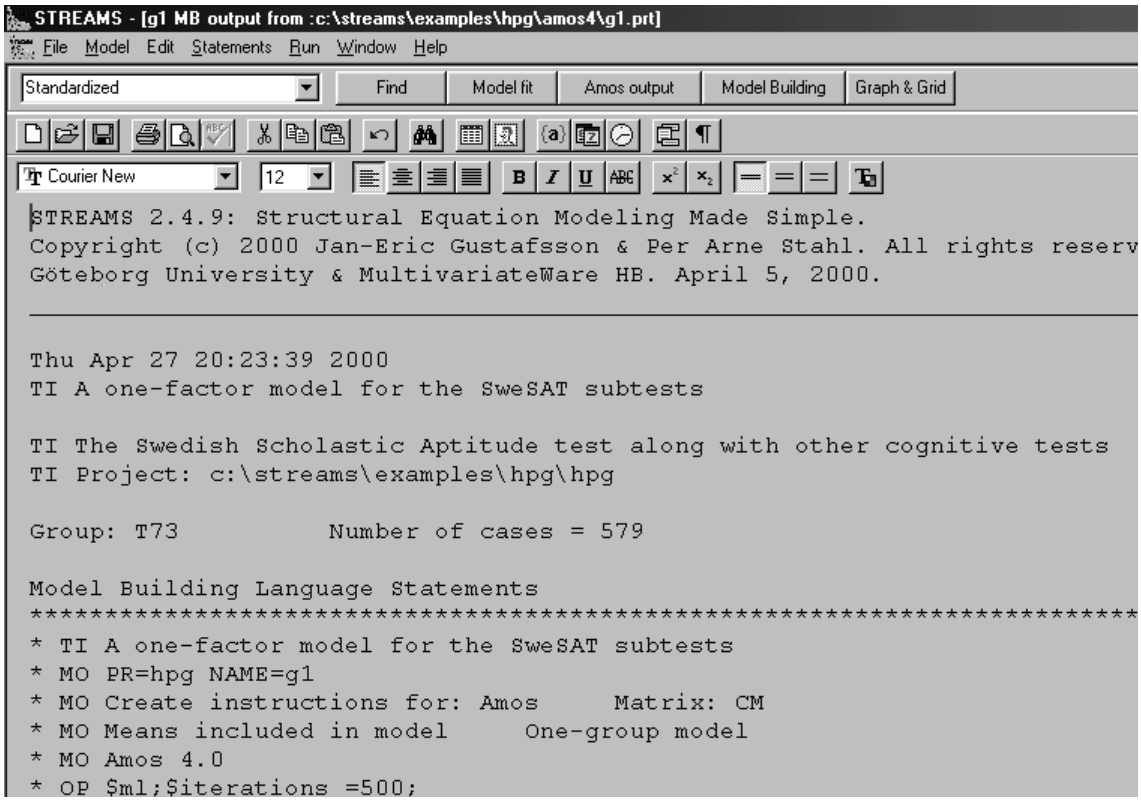


The currently selected estimation program is marked by a blue wrench, available programs by a grey wrench, the selected default program by a green wrench, and non-available programs by a crossed-over wrench (see Chapter 14 for information how to make STREAMS aware of installed estimation programs). To swith to another program or program version the program name is selected and the **Use program** button is clicked. Here, however, we will stick to the Amos 4 program.

When the **Run** button is clicked (or the **Run** item under the **Run** menu is selected) the pre-processor, which translates the MB instruction into instructions for the choosen estimation program, is started. This is shown by the message “Running pre-processor ...” which is displayed in the status line at the bottom of the window.

When the pre-processor has finished its work, the estimation program automatically starts. The estimation program runs in its own window, and the message “Running Estimation ...” is displayed in the status line.

When the estimation process is completed STREAMS starts the post-processor and the message “Running post-processor ...” is displayed. When this process is completed, the result is presented in an output file, which is shown in an editor window:



The text in the output file may be scrolled and browsed, and it may also be printed, using the **Print** function under the **File** menu. It is also possible to search for a specified string of text, either through selecting one of the pre-entered strings, or through writing a string in the field, and then clicking the **Find** button.

The post-processor output file (which has the suffix *.prt*) includes several sections of output: after the model statements have been listed, basic goodness-of-fit information is presented. Then follows a section presenting unstandardized parameter estimates, after that t-values are presented, and finally the standardized estimates are presented. In this case the following standardized estimates are obtained:

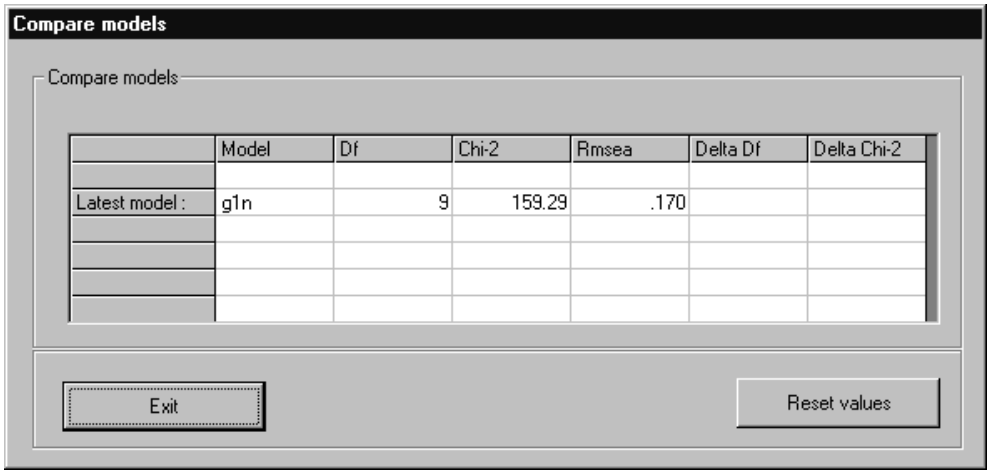
```
Standardized estimates:

WORD      =     +0.80*g          +0.60*WORD&
DS        =     +0.55*g          +0.84*DS&
READ      =     +0.74*g          +0.67*READ&
DTM       =     +0.52*g          +0.85*DTM&
GI        =     +0.67*g          +0.74*GI&
ERC       =     +0.80*g          +0.60*ERC&
```

As may be seen there are high standardized relations between the latent variable and all the manifest variables, but the relations do seem to be particularly high for the verbal subtests (i. e., WORD, READ, GI and ERC).

There also are several other ways to display the estimation results. On the toolbar there is a button labeled **Amos Output** (the label changes when another estimation program is selected) and when this button is clicked the program presents the listing file from the esti-

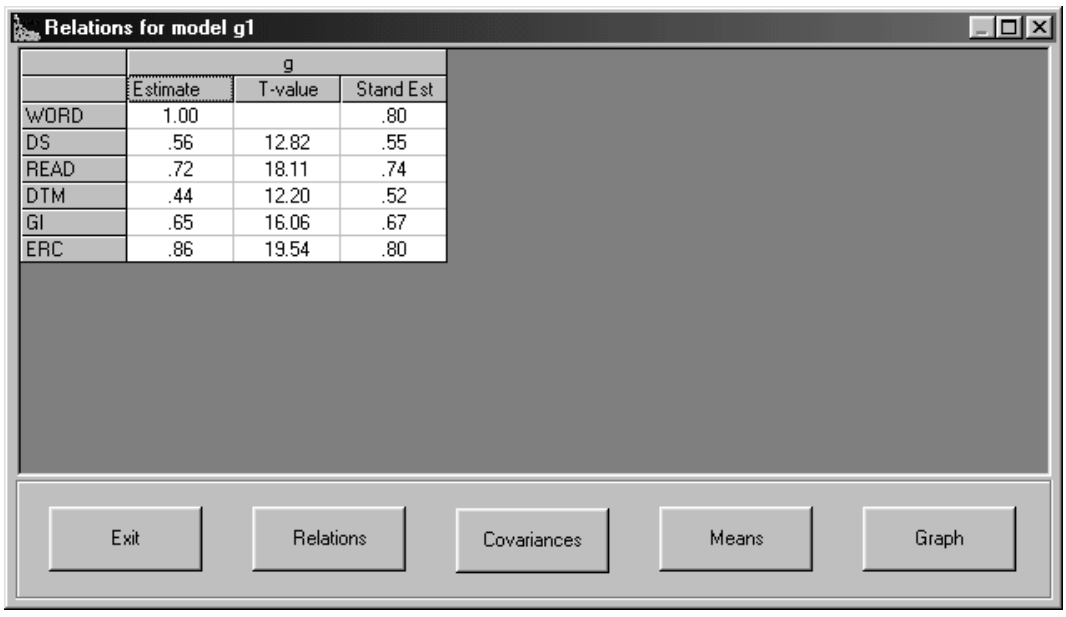
mation program. There is often a need to consult this file, for example to inspect the complete set of modification indices.

There is also a button labeled **Model fit,** which may be clicked to obtain results from the goodness-of-fit tests:

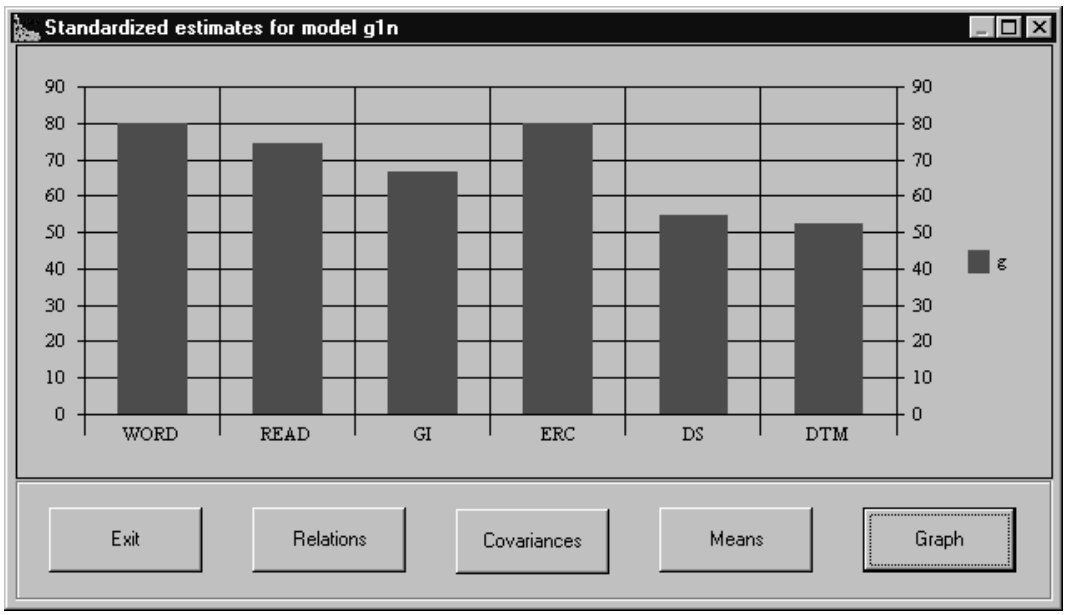


In this case the test-statistic is highly significant, and the RMSEA measure also indicates a poor fit of the model to data. These results thus indicate that a one-factor model does not reproduce the observed covariance matrix particularly well.

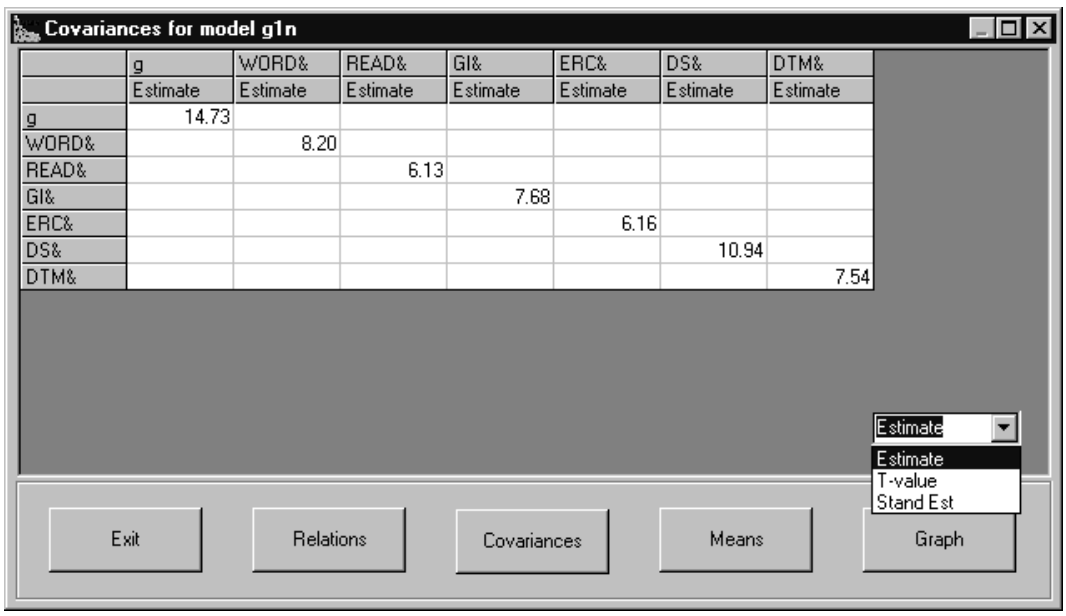
If the button labeled **Graph & Grid** is clicked a grid is presented:



When the **Graph** button on this form is clicked a chart is shown, which displays the standardized loadings on the latent variables (using a different color for each latent variable) for each manifest variable:



This chart often is useful for spotting patterns and trends in the relations among manifest and latent variables. The window also includes buttons labeled **Relations, Covariances** and **Means**. When the **Relations** button is clicked the grid shown above is presented. Clicking the **Covariances** button yields the following results:



Here the estimated variances and covariances are presented, and the drop-down menu may be used to select whether unstandardized estimates, t-values or standardized estimates are to be presented.

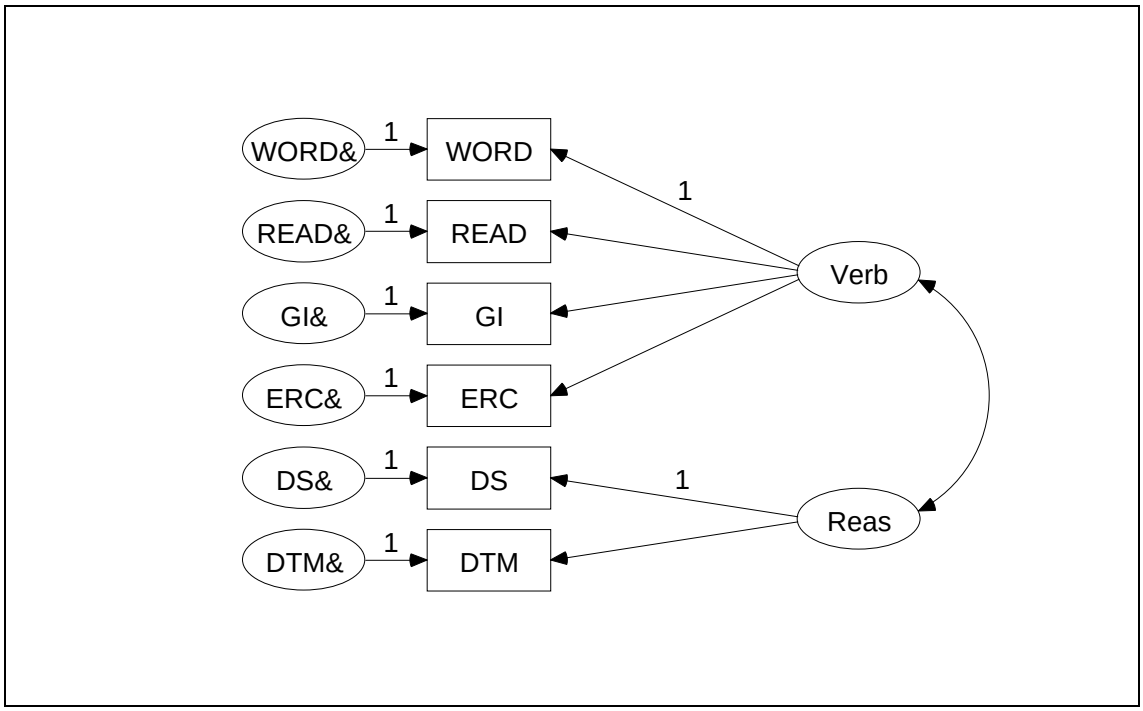
After the results have been inspected the model may be changed and reestimated. The post-processor listing file is left through clicking the **Model Building** button (or clicking

the **F8** function key, or selecting the **Model Building** item under the **Window** menu). The model may then be edited and reestimated. It should be observed, however, that unless the model name is changed the edited version will overwrite the original version of the model, which thus will be lost. To keep the old version of the model, the name should be changed by clicking the **Model** button, and entering a new name in the **Model Name** field under the **Model Description** tab.

3 Specifying and Editing Models

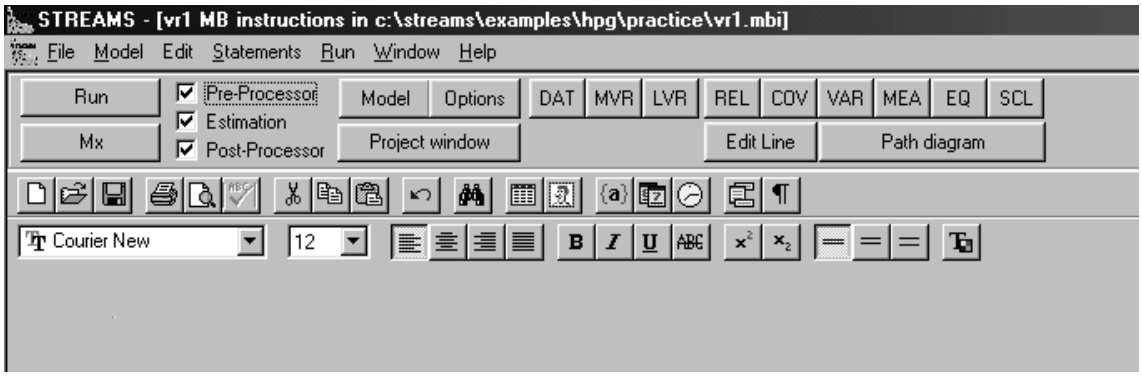
The present chapter describes the techniques for specifying models with the MB language and with path diagrams created with Amos 4.0. It will be assumed that a new model is to be created, but often an existing model is taken as a starting point, as is described in Chapter 2.

We will continue using the example discussed in Chapter 2. Having found that a one-factor model does not fit the 6 subtests of the SweSAT, we will instead try a model with two correlated factors, as is shown in the path diagram below:



One factor is hypothesized to be a reasoning factor (*Reas*) related to the two manifest variables DS and DTM, and the other factor is taken to be a verbal knowledge factor (*Verb*), with relations to WORD, READ, GI, and ERC.

The first step is to open the HPG project using the procedure described in Chapter 2. The **Models** tab on the *Project Window* is then clicked. Next an appropriate folder is opened or selected. There is already a folder named *Practice*, and this is where we will put the new model. The *Practice* folder is thus double-clicked, and then the **New Model** button on the **Model** tab is clicked. This opens up the *Model Building* form, and a blank edit area:



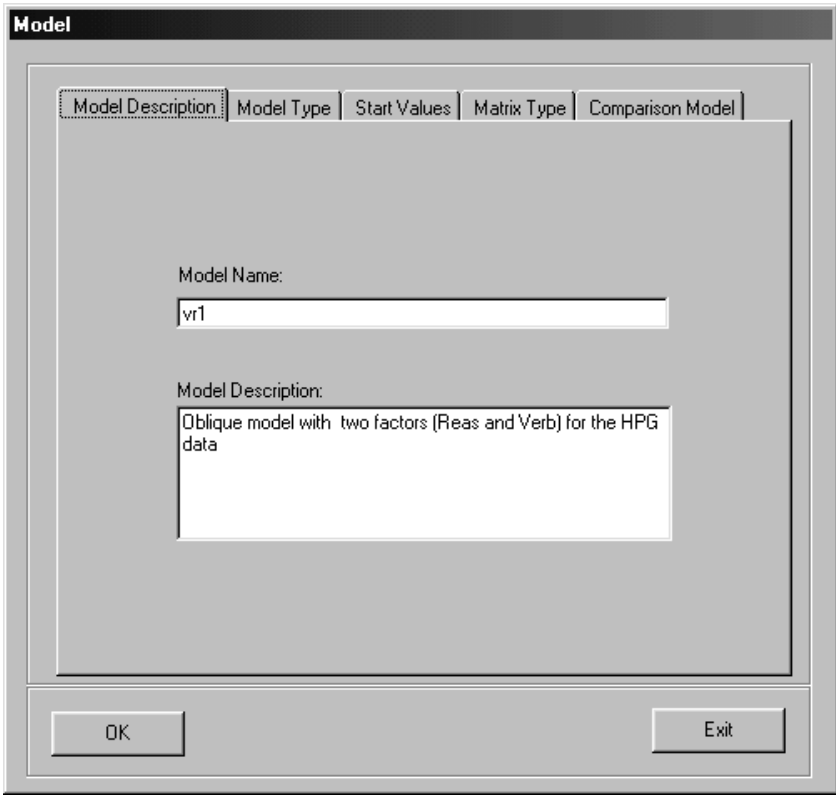
When a new model is to be specified the user typically clicks each of the buttons on the Modeling toolbar, starting with the **Model** button and going right.

Specifying a Model with the MB language

Whether the model is to be specified as a path diagram, or as a set of MB statements, some basic information about the model needs to be given, and options concerning the estimation need to be specified. This is done through two forms (The *Model* form and the *Options* form), which are common to the two modes. We will go through these quickly, only bringing up the most essential information.

The Model Form

When the **Model** button on the Modeling toolbar is clicked the *Model* form is presented:



The *Model* form has five tabs:

- **Model Description**
- **Model Type**
- **Start Values**
- **Matrix Type**
- **Comparison Model**

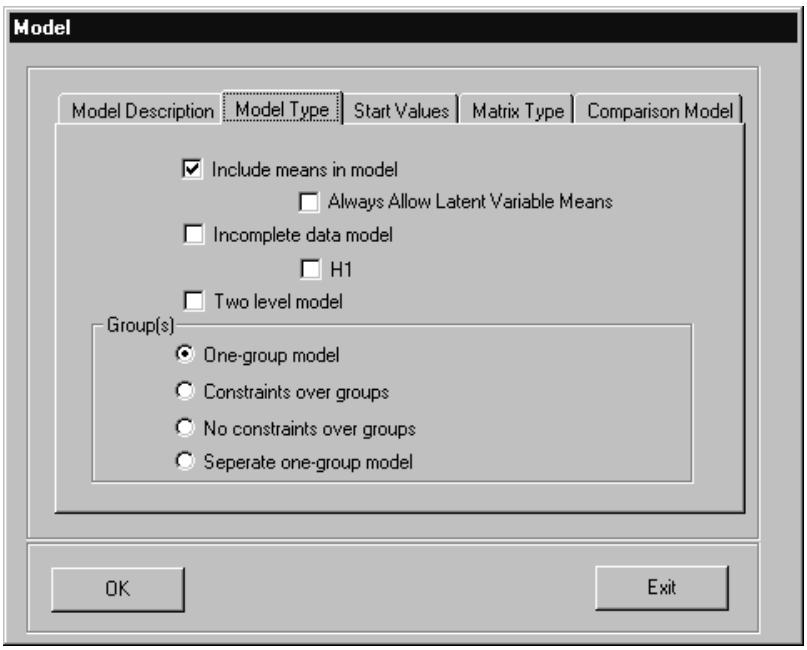
Model Description

On the **Model Description** tab the name of a new model may be entered into the **Model Name** field (1-64 characters). Here we enter the name VR1. If an existing model is edited, a name that has previously been entered into this field may be changed. This corresponds to the **Save Model as ...** function under the **Model** menu.

A description of the model must also be provided in the **Model Description** field. This information is used by STREAMS to construct the TI statements. Here a short description of the nature of the model is provided ("Oblique model with two factors ...").

Model Type

When the **Model Type** tab is selected the *Model* form presents options concerning the basic structure of the model.



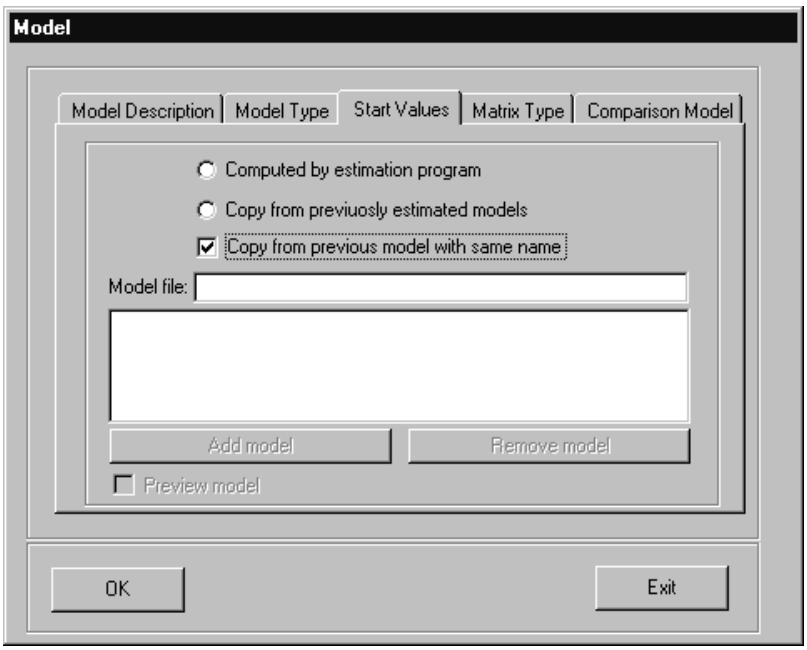
Here we can use the default options, which are that means will be included in the model, and that a one-group model will be fitted.

In a one-group model inclusion of means does not affect the results in any way if constraints of equality are not imposed over the means, because the means of the manifest variables are treated as free parameters to be estimated, and there are as many parameters as there are manifest variables. If the model in a later step is to be developed into a multiple-group model these estimates may, however, be useful as a source of start values. When the model comprises multiple groups, different kinds of models are created depending on whether means are included or not. The decision whether to include means or not must thus be made on the basis of the nature of the substantive problem that is being studied.

This form also offers check-boxes for identifying an **Incomplete Data Model**, **Two Level Model** and **H1 model**. These options may be used to specify certain types of models for complex observational data. More information about these advanced models is given in Chapters 4, 5 and 6. The **Model Type** tab also offers possibilities for requesting three different types of muItiple group models. The meaning of these options are explained in Chapters 4 and 12.

Start Values

When the **Start Values** tab is selected the *Model* form offers several options about procedures for determining start value for free parameters.



The three options are:

- **Computed by Estimation Program**. This option implies that STREAMS will rely on the procedures, if any, which may be available in the estimation programs to compute start values.
- **Copy from Previously Estimated Models.** When this option is selected the **Add Models** and **Remove Models** buttons are enabled, which makes it possible to identify one or more models from which start values will be copied. Here, however, we will assume that no such models are available.
- **Copy from Previous Model with Same Name**. When this option is selected an instruction is generated to take start values from a previous version of the current model. It is recommended that this option is regularly used.

More information about how the system of copying start values works and may be used is given in Chapter 13.

Matrix Type

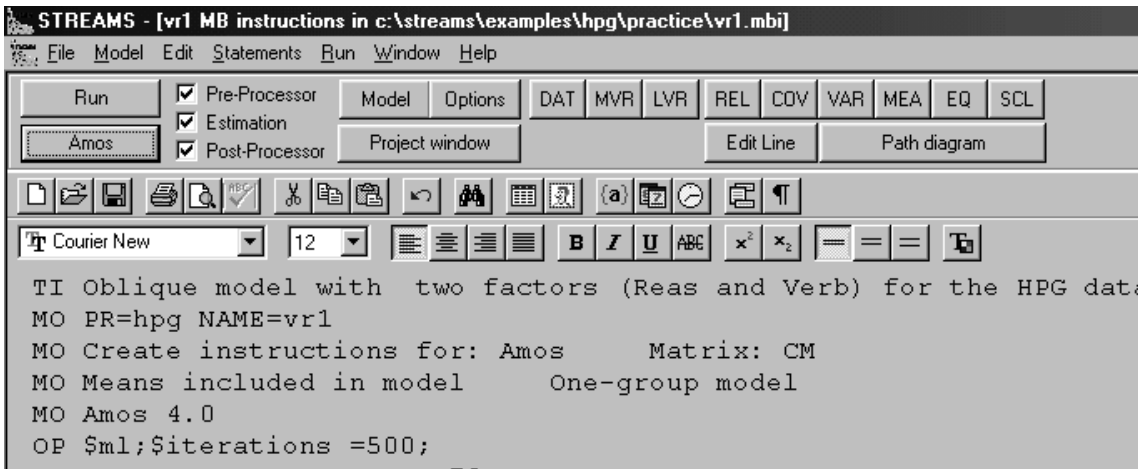
When the **Matrix Type** tab is clicked, three options concerning the type of matrix to be analyzed are presented: **Covariance Matrix**, **Pearson Correlation Matrix**, or **Polychoric Correlation Matrix**. The default is **Covariance Matrix**, which kind of matrix will be used in the present example.

Comparison Model

The **Comparison Model** tab may be used to select a model with which the goodness-of-fit of the model to be estimated will be compared. This option will not be used in the present example, however.

Constructed Statements

When the **OK** button is clicked on the *Model* form STREAMS writes a set of MB statements into the edit area:

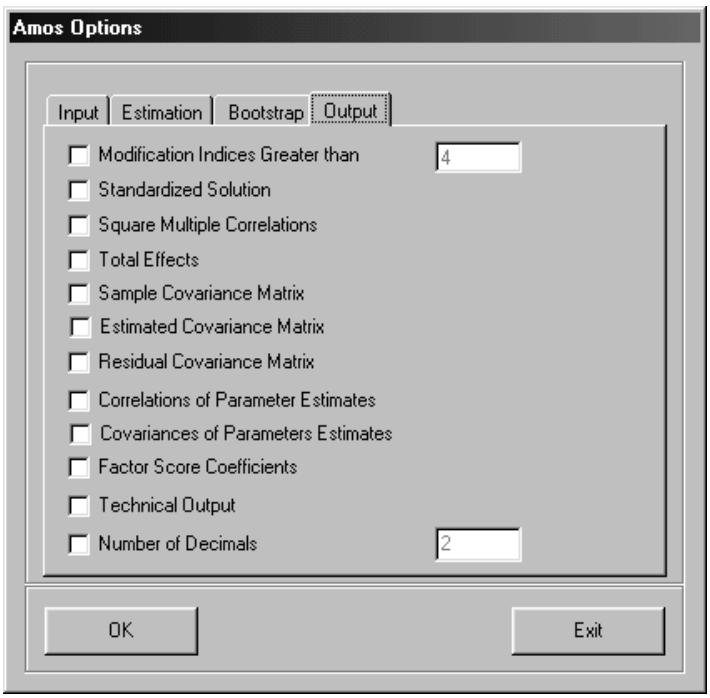


STREAMS has constructed TI, MO, STA and OP statements, which provide basic information about the name of the model, the project, start values, estimation program and so on. These MB statements are more or less self-explanatory, but are more fully explained in Chapter 12. The OP statement specifies default options for the Amos program.

The default OP statement may be changed, however. One way to do this is to double-click on the OP statement, and another way is to press the **Options** button. Both these methods bring forward the *Options* form which is appropriate for the selected estimation program: *Amos Options, EQS Options, LISREL 8 Options, Mplus Options* or *Mx Options.* Full information about these options is given in Chapter 13, so here we will just take a quick look at the *Amos Options* form.

The Options Form

Clicking the **Options** button (which is to the right of the **Model** button) produces the following form, when Amos has been selected as the estimation program:



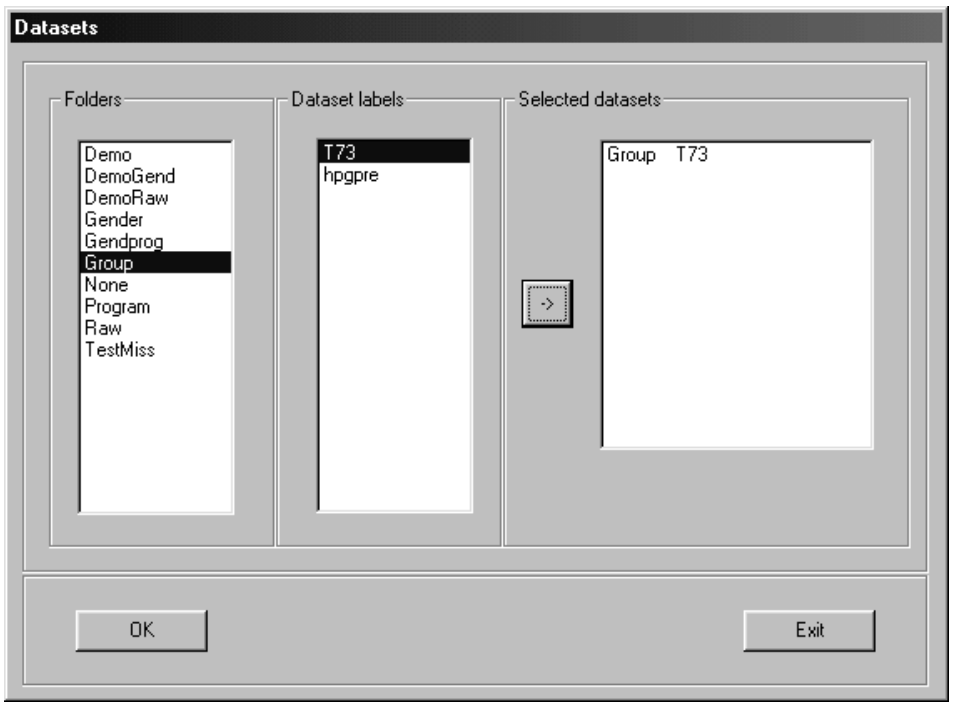
This form has three tabs:

- **Input**
- **Estimation**
- **Bootstrap**
- **Output**

The options available on these tabs are described in Chapter 13.

Selecting the Data to be Analyzed

The next step in the model specification sequence is to identify data for one or more groups to be included in the model. This is done through clicking the **DAT** button. When this is done the *Datasets* form is shown:



The list-box on the left-hand side presents the folders that have been defined for the project, and when one of these is selected the dataset labels which have been defined for this folder are displayed. To identify a dataset both the folder and the dataset label must be selected. When this has been done for one or more datasets the button marked with an arrow may be clicked, which causes the selected datasets to be moved to the list-box on the right-hand side. Datasets may also be deselected through moving them back again. Here we want to analyze the covariance matrix for the total group of cases, which is in the folder *Group* and has the dataset label *T73*.

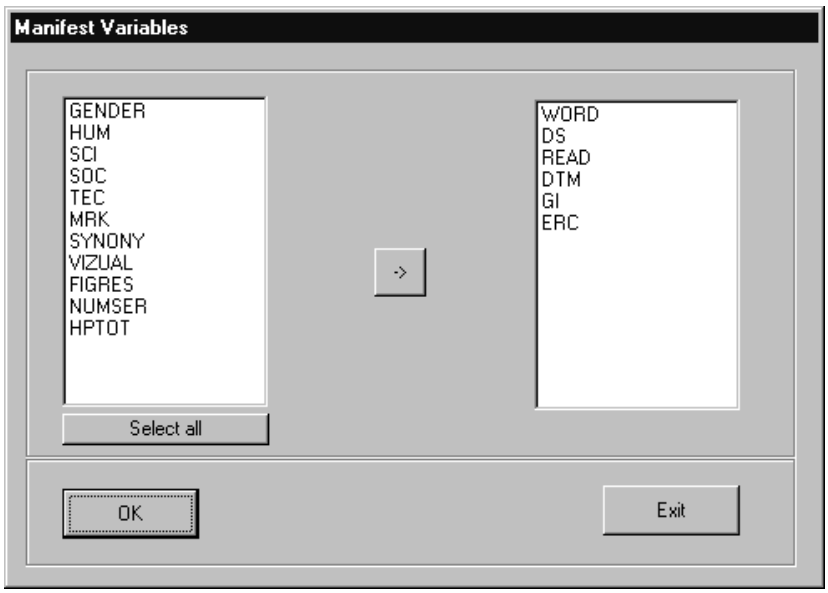
When the **OK** button is clicked one or more DAT statements corresponding to the selections made are constructed. Thus, in the present case STREAMS inserts the following statement into the edit area:

```
DAT FOLDER=Group DATLAB=T73
```

If we want to add or remove datasets, the DAT button may be clicked at any time to retrieve the *Datasets* form. Double-clicking on a DAT line will also produce this form.

Selecting Manifest Variables

The MB language requires that all manifest variables are declared, which is done with the MVR statement. To select all variables or a subset of variables from the project the **MVR** button is clicked. When this is done the *Manifest Variables* form is shown:



Variables are identified through selecting one or more variables in the list on the left-hand side and when the arrow button is clicked these variables are moved to the list on the right-hand side. This process may be repeated any number of times, and the same procedure may also be used to move variables from the right-hand side to the left-hand side.

When the **OK** button is clicked an MVR statement is put into the edit area by STREAMS. For example, if the variables WORD, DS, READ, DTM, GI and ERC were selected the MVR statement would be:

```
MVR WORD DS READ DTM GI ERC
```

To change the selection of variables the MVR button may be clicked again, or an MVR statement may be double-clicked.

Identifying the Latent Variables

The latent variables must be declared as well, and because these are unknown to STREAMS, labels of the latent variables must be supplied. This is done on the *Latent Variables* form, which is presented when the **LVR** button is clicked.

Labels of the latent variables are entered in the top white field, and then the **ADD** button is clicked, which moves the new label to the list of latent variables. This process is repeated as many times as there are labels to be entered. To delete an already entered label, select it in the list and click the **Delete** button. When the list contains the labels for the latent variables to be included in the model, click the **OK** button. This will cause STREAMS to construct an LVR statement, and add it to the edit area.

If, for example, the latent variables Verb and Reas are entered, STREAMS inserts the line:

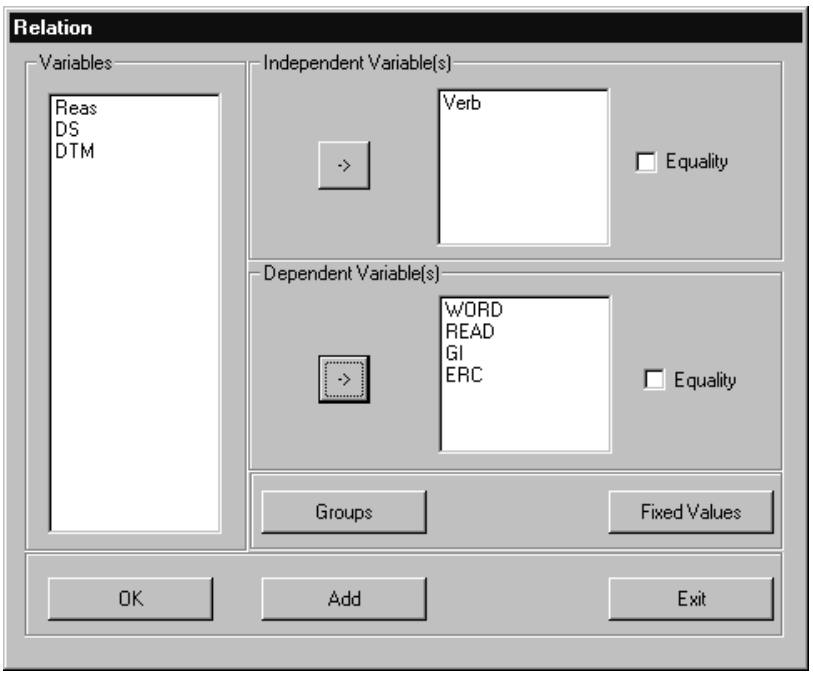
```
LVR Verb Reas
```

Latent variables may be added and deleted at any time, through double-clicking on the LVR statement, or through clicking the **LVR** button.

Specifying the Model

When groups have been selected, and the manifest and latent variables have been declared, the actual model specification may be started. The Model Building toolbar offers a set of buttons (REL, VAR, COV, MEA and SCL) some or all of which may be used in the process of model building.

Most models involve one or more relations, and to specify these the *Relations* form is used. Clicking on the **REL** button causes this form to be shown:



This form displays the variables available for modeling in the **Variables** list on the left hand side. This list includes the declared manifest and latent variables, along with residual variables for those manifest and latent variables that have previously been defined as dependent variables. The residual variables have the same label as the dependent variables, but with an ampersand (&) added to the label.

The form also presents one list-box for independent variables, and one list-box for dependent variables. Initially these are empty (unless the *Relations* form has been opened by double-clicking on a REL statement) but variables in the **Variables** list may be moved to either of these list boxes. This is done through selecting one or more variables in the list using the standard techniques, and then clicking the appropriate button with an arrow. The same technique may be used to move variables from the lists of independent or dependent variables to the **Variables** list.

Next to the list boxes for independent and dependent variables are check boxes labeled **Equality**. When these are checked (through clicking) equality constraints are imposed for the marked category of variables.

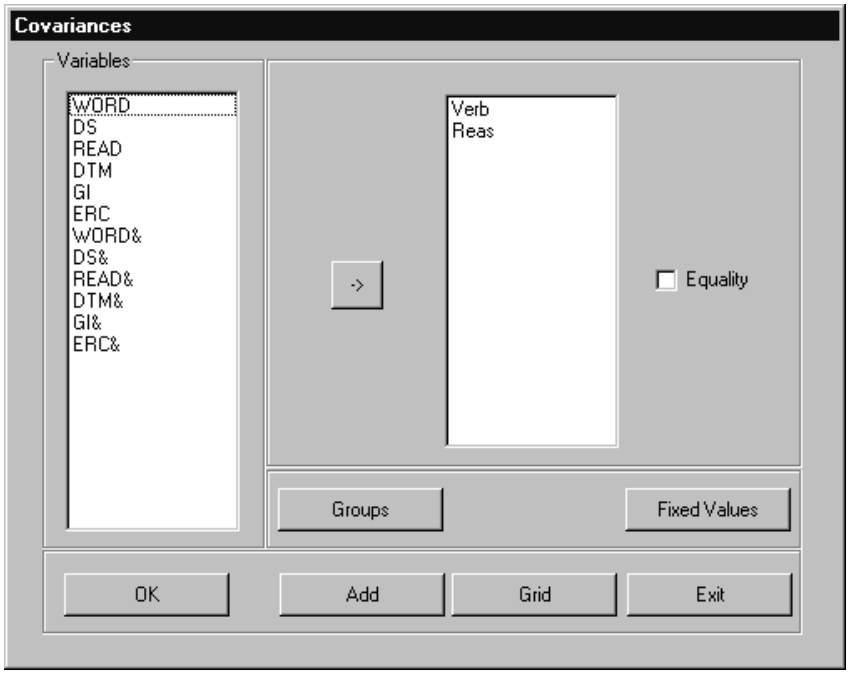
When the relations have been specified as desired the **OK** button on the *Relations* form is clicked. This causes STREAMS to contruct a REL statement which is inserted with the other statements in the edit area.

To define the relations between the two latent variables and the manifest variables we may thus use this technique to construct the following two REL statements

```
REL Verb -> WORD READ GI ERC
REL Reas -> DS DTM
```

In order to change an existing REL statement the REL word in the statement may be double-clicked. To introduce another REL statement, the **REL** button should be clicked again.

In order to complete the specification of the oblique two-factor model we also need to specify a covariance between *Verb* and *Reas*. When the **COV** button is clicked the *Covariances* form is shown:



This form is used to select a set of variables among which covariances are estimated. In the Variables list box on the left hand side the complete list of available variables, including residuals in manifest and latent variables, are shown. Using the procedure described above two or more of these variables may be moved to the empty list box on the right hand side. The selected variables will be included in a COV statement which is constructed when the **OK** button is clicked.

Thus, if the Verb and Reas variables are moved to the right hand side and the **OK** button is clicked the following statement is produced:

```
COV Verb Reas
```

This statement implies that the two latent variables *Verb* and *Reas* are allowed a covariance, because STREAMS by default assumes all variables to be uncorrelated.

The model specification is now complete, and the model may be estimated in the manner described in Chapter 2. If that is done we will find that the model has a reasonably good fit (χ^2 (8) = 19.11, p < .01, RMSEA = .049) and that the parameter estimates seem to be in order.

Editing the Instructions

It has now been demonstrated how the command buttons on the *Model Building* form may be used to construct the MB statements. However, the editor may also be used as any ordinary editor to enter and edit the MB instructions. For keyboard editing the standard set of editing tools is available (e. g., cut, copy, and paste). It should also be observed that the

characters **/*** function as comment characters when put in positions 1-2 of the line. These characters thus cause the pre-processor to disregard the line in its entirety.

When an MB statement is double-clicked, the appropriate form is brought up with the information contained in the statement. Another method to accomplish the same thing is to put the insertion point on the statement to be edited, and click the **Edit Line** button. The functions on the form may then be used to edit the statement, and when the OK button on the form is clicked the edited line is written back to the edit area.

Advanced Editing Tools

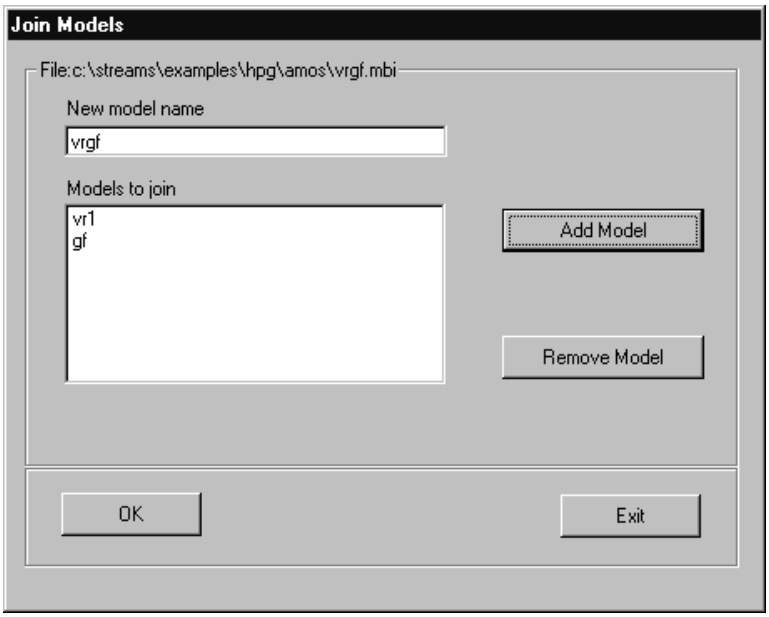
In MB mode STREAMS also offers some advanced editing tools.

Joining models

The function for combining two or more models into one is an extremely useful editing tool, and particularly so when several large submodels are to be joined into one model. Here we will illustrate the procedure with two small models.

Suppose that we want to combine the two-factor model for the SweSAT (*vr1.mbi*), with a model for the cognitive tests also included in the HPG project. A separate one-factor model (*Gf.mbi*) has been tested for the three tests VIZUAL FIGRES NUMSER, which are assumed to measure the broad non-verbal reasoning dimensions Fluid Intelligence (*Gf;* see, e. g., Carroll, 1993; Gustafsson & Undheim, 1996). A one-factor model with only three manifest variables is just-identified, so the fit of this submodel cannot be tested. However, the type of tests employed here have in many other studies been shown to be good measures of the Gf-dimension.

In order to combine the *vr1.mbi* and the *gf.mbi* models the menu item **Join Models** on the **Edit** menu is selected, which produces the *Join Models* form:



The field labeled **New model name** should be completed with a name for the combined model (here *vrgf*). The models to be joined should then be identified, one by one, which is done through clicking the **Add Model** button. This produces the standard file open dialogue box, which allows selection of an *.mbi*-file in one of the folders belonging to the project. After a model has been added it may be removed. To do that the model should be selected, and the **Remove Model** button clicked.

When all model files have been added, the **OK** button is clicked, which combines the models in such a way that all manifest variables and all latent variables in the models are included in the joined model, along with the structural relations identified. After the joined model has been constructed, it must be opened, using either the **Models** tab on the *Project* window, or the **Open model** function on the **Model** menu. In our example the following model results:

MB instructions for the combination of the vr1 and Gf models

```
DAT FOLDER=Group DATLAB=T73
MVR DTM DS WORD READ GI ERC VIZUAL FIGRES NUMSER
LVR Reas Verb Gf
REL Verb -> WORD READ GI ERC
REL Reas -> DS DTM
REL Gf -> VIZUAL FIGRES NUMSER
COV Reas Verb
```

The joined model statements may require some editing. Thus, if the different models have been fitted to different matrices, the resulting statements will include multiple DAT statements, but a one-group model may only include a single DAT statement. It will, of course, also be necessary to add statements which relate variables in the original models to one another. Here, for example, we might consider expanding the COV statement to include all three latent variables, i. e.:

```
COV Reas Verb Gf
```

If this is done, the three-factor model achieves a very good fit (χ^2 (24) = 40.42, p < .02, RMSEA = .034), with the following standardized estimates:

Standardized estimates for the combined model

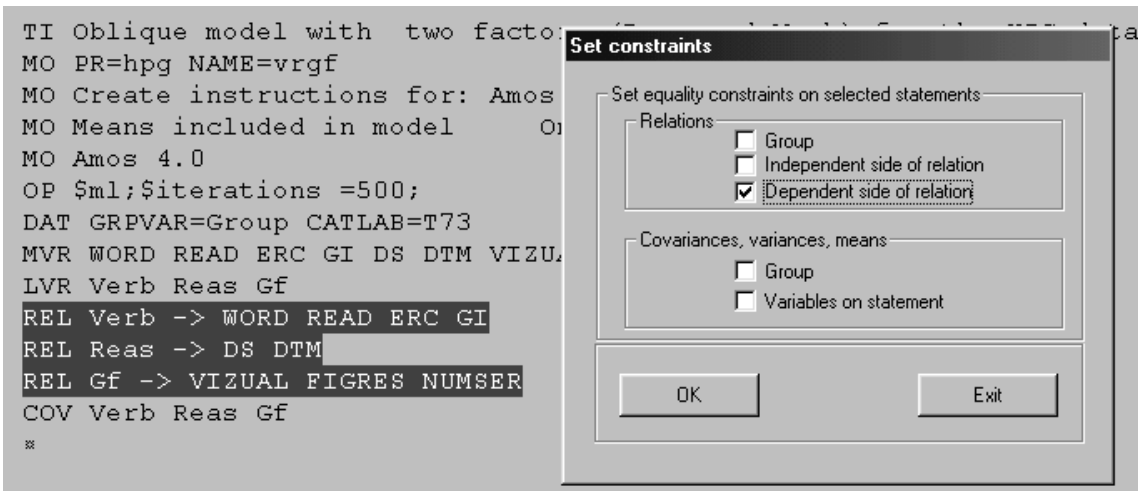
```
Standardized estimates:

DTM      =    +0.75*Reas       +0.66*DTM&
DS       =    +0.80*Reas       +0.60*DS&
WORD     =    +0.82*Verb       +0.58*WORD&
READ     =    +0.74*Verb       +0.67*READ&
GI       =    +0.67*Verb       +0.74*GI&
ERC      =    +0.80*Verb       +0.60*ERC&
VIZUAL   =    +0.66*Gf         +0.75*VIZUAL&
FIGRES   =    +0.74*Gf         +0.67*FIGRES&
NUMSER   =    +0.49*Gf         +0.87*NUMSER&


Cov(Verb,Reas)        =    0.63
Cov(Gf,Reas)          =    0.87
Cov(Gf,Verb)          =    0.44
```

There is, thus, a very high correlation between the *Reas*-dimension, and the *Gf*-dimension.

Set Constraints

The MB language allows simple and efficient means of imposing equality constraints over variables and groups (see Chapter 12). This is useful for model testing purposes, and as demonstrated in Chapter 7, imposing strong constraints on a model is also an efficient way of achieveing a solution, which then provides start values for the final, less constrained, model. This kind of application makes it useful, however, to have a method for imposing constraints, and relaxing constraints, for many statements at the same time.

One way of doing this is the following. Select a range of MB statements, and then chose the **Set constraints** option on the **Edit** menu. This produces the *Set constraints* form:



The form offers options for the REL statement in one set of check-boxes, and options for the other statements in separate check-boxes. When a check-box is marked, this implies that constraints of equality will be imposed over a particular part of the statement, such as the one referring to groups or the dependent variables in a relation.

In the example shown here, equality constraints are imposed over the dependent variables of the selected REL statements. When the **OK** button is clicked, the following result is produced:

```
REL Verb -> (WORD READ GI ERC)
REL Reas -> (DS DTM)
REL Gf -> (VIZUAL FIGRES NUMSER)
```

To remove the equality constraints, the statements are selected again, the check-box on the *Set constraints* form is unchecked, and the **OK** button clicked.

Auto-Removal of Manifest and Latent Variables

When variables are removed from the MVR and LVR statements it often is a good idea to delete the removed variables from all statements in which they appear, and to delete all the statements which have lost their meaning when these variables have been removed. STREAMS offers such a function, which by default is enabled. The function may, however, be disabled, which is done with the **Preferences** option on the **File** menu. The **General** tab offers a check-box labeled **Autoremove variables on change**, which

should be unchecked to disable the function.

If, for example, the Gf-factor in the *vrgf.mbi* model is deleted from the LVR form, this causes the REL statement with Gf as an independent variable to be deleted from the model, and it causes Gf to disappear from the COV statement. It should be observed, however, that the three manifest variables which were related to Gf are still in the MVR statement, and if they are not to be used in any other statement, they must be removed from the MVR statement.

Using Amos Graphics with STREAMS

STREAMS 2.5 takes advantage of the programmability features of Amos 4 (Arbuckle & Wothke, 1999), which, among other things, makes it possible to generate a path diagram from an MB specification, use the Amos Graphics editing tools to edit the path diagram, and then translate the edited model back to the MB language. This brings several advantages:

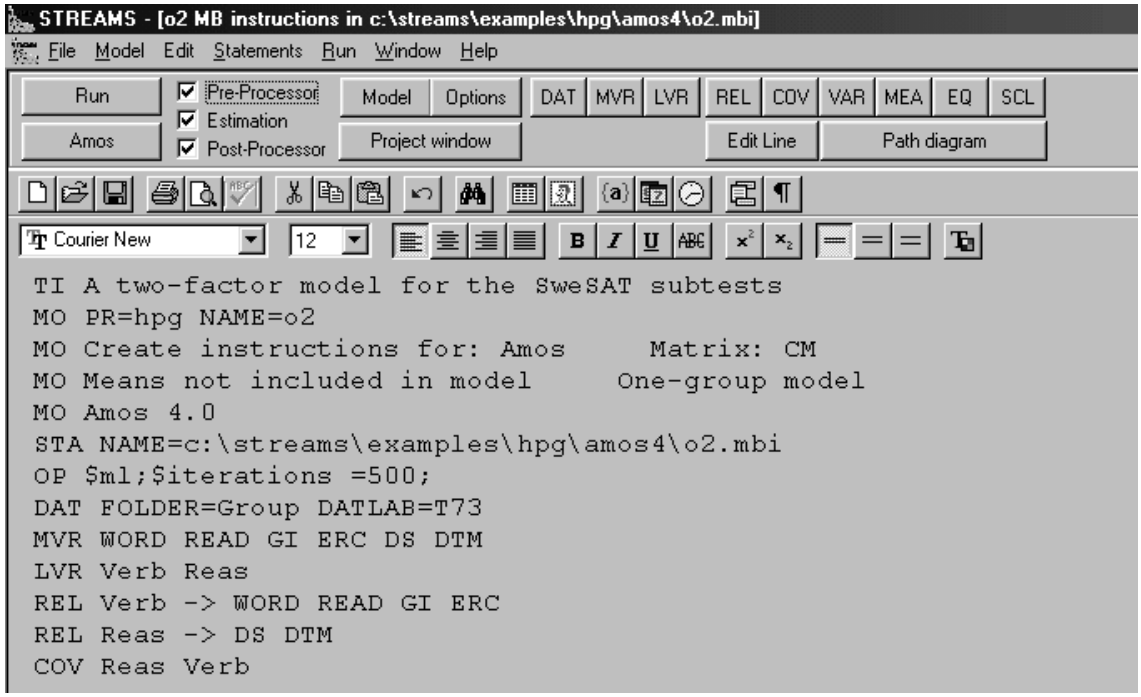
- The MB model is effortlessly transformed into a publication quality path diagram.
- The powerful and user friendly editing tools available in Amos 4 are made available to the STREAMS user. This makes, indirectly, Amos Graphics available as a user interface for all SEM programs.
- The MB model and the path diagram are aligned with one another, and the user may freely swith between the two modes of representation, which makes it easy to take advantage of the relative strengths of the two ways of model specification. The MB language thus is efficient for specifying and editing large models with many variables, while the path diagram representation is useful for presenting complex model structures.

When the **Path diagram** button is clicked Amos Graphics is started and the path diagram is created. Amos Graphics is only available for one-group models. To return from Amos 4, the user must transfer control to STREAMS, either through clicking the STREAMS program button on the Windows taskbar, or through using the Alt Tab function. The allocation of control between STREAMS and Amos 4 is somewhat more complex than usual because two programs are involved. However, if the simple rules and principles described below are followed, the user should be able to take advantage of the great benefits and synergies obtainable by having two collaborating programs.

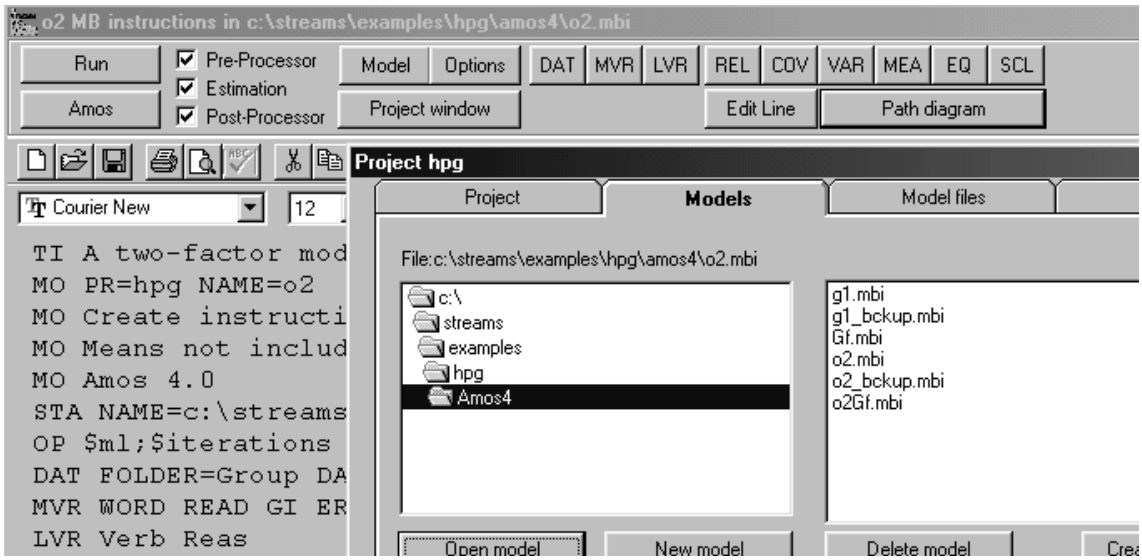
Creating an Amos Path Diagram from an MB specification

When a path diagram is to be created with Amos Graphics, the model should first be specified in the MB language. This is necessary because latent or manifest variables may not be added in Amos Graphics. After the diagram has been created it may, however, be edited in several different ways: latent and manifest variables may be deleted; relations and covariances among variables may be added and deleted; and equality constraints may be imposed or removed, and so on.

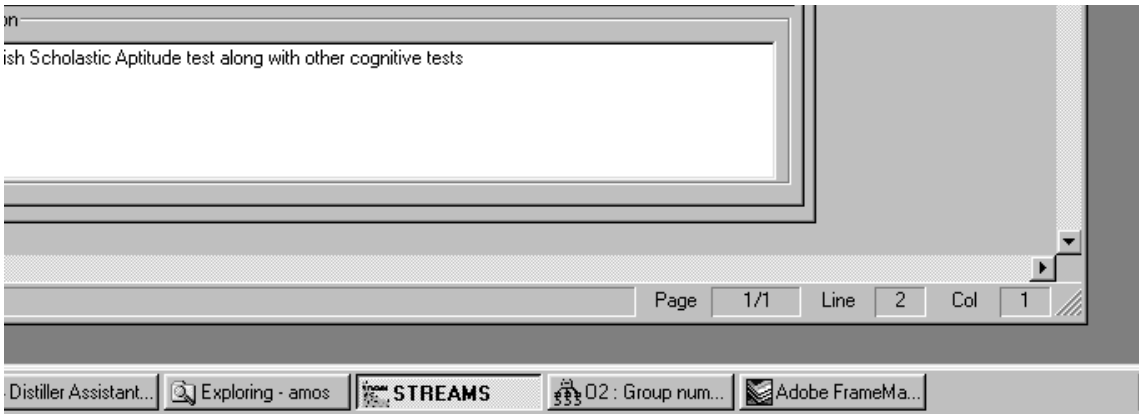
Let us assume that the model shown below has been specified:



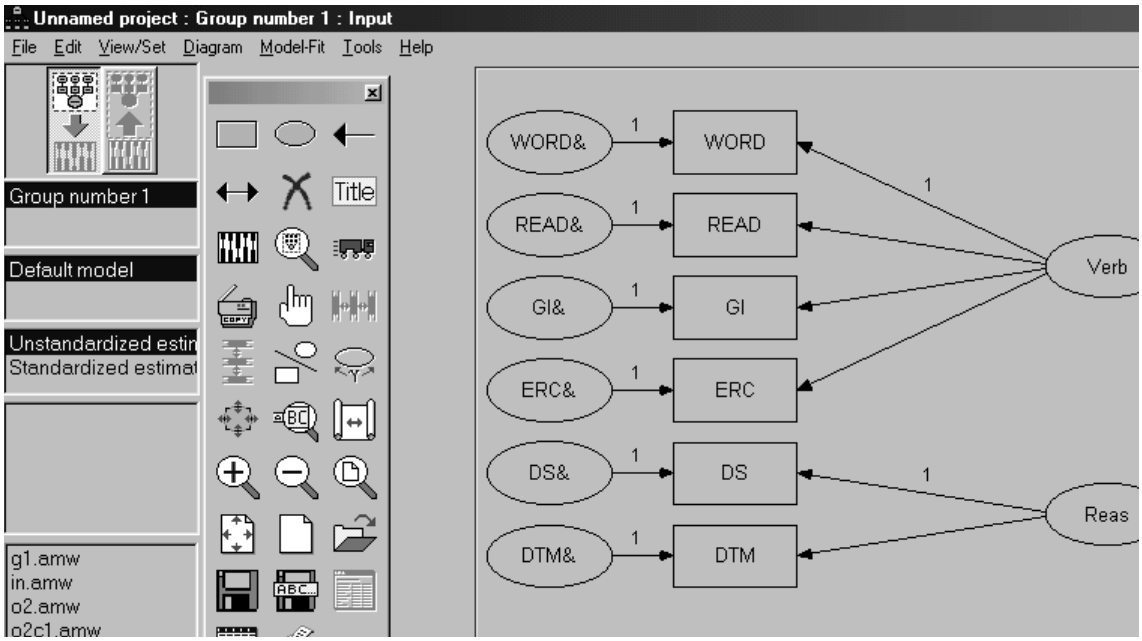
When the **Path diagram** button is clicked, Amos Graphics starts (if it is not already active) and in STREAMS the *Project window* is put on top of the model windows:



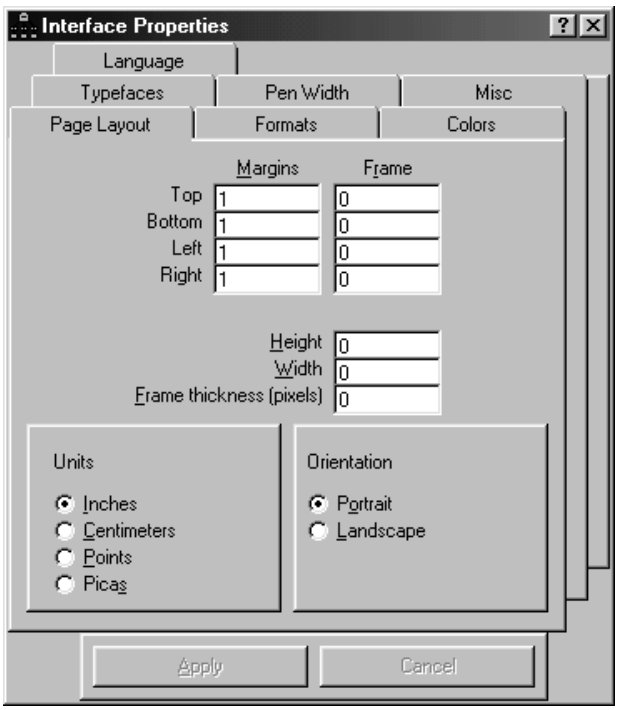
After Amos Graphics has started, the path diagram is constructed and immediately displayed. However, if Amos Graphics is already active, the path diagram is constructed in the background and it is not displayed until the user clicks the Amos program button on the taskbar:



When the **Amos program button** is clicked Amos Graphics becomes the active program and the path diagram is shown:



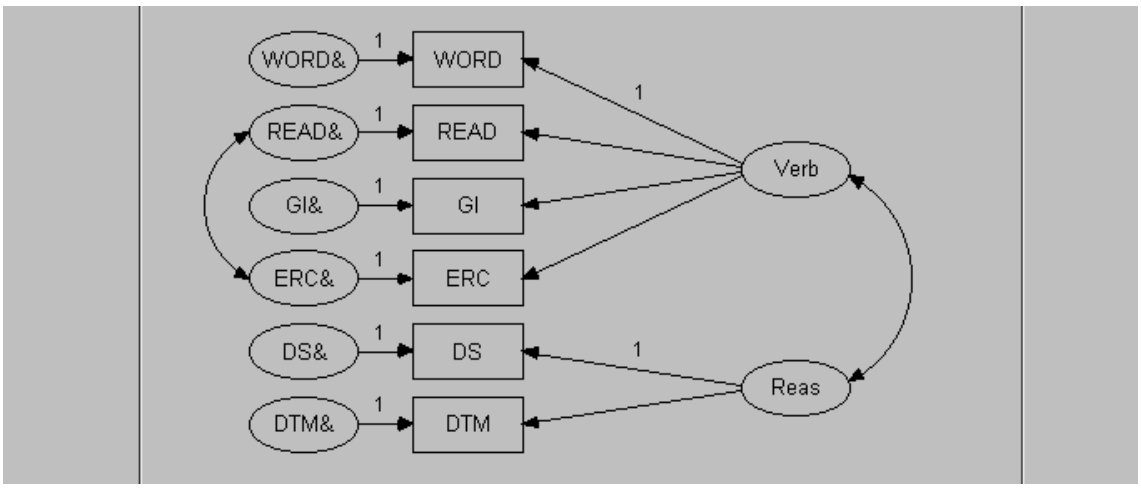
The path diagram created by STREAMS may need some further editing to be aesthetically pleasing and some aspects of the diagram, such as whether the page orientation should be portrait or landscape, cannot be controlled from STREAMS. However, Amos Graphics offers a rich set of powerful and easy-to-use editing tools (see Arbuckle & Wothke, 1999). Thus, to change the page orientation the **Interface properties...** option under the **View/ Set** menu may be used:



For a description of all the other useful editing facilities available in Amos Graphics the documentation and/or help functions of Amos 4 should be consulted.

As has already been mentioned the model may edited quite freely in Amos Graphics, except that addition of latent and/or manifest variables prevents the model from being transformed back into the MB representation. If there is no need to bring the model back to STREAMS again, the model may, of course, be edited completely freely.

Suppose that we have used the Amos Graphics function to resize the path diagram to fit on one page, and that we have used the double-headed arrow tool to add a covariance between READ& and ERC&. This will have caused the path diagram to take on the following appearance:

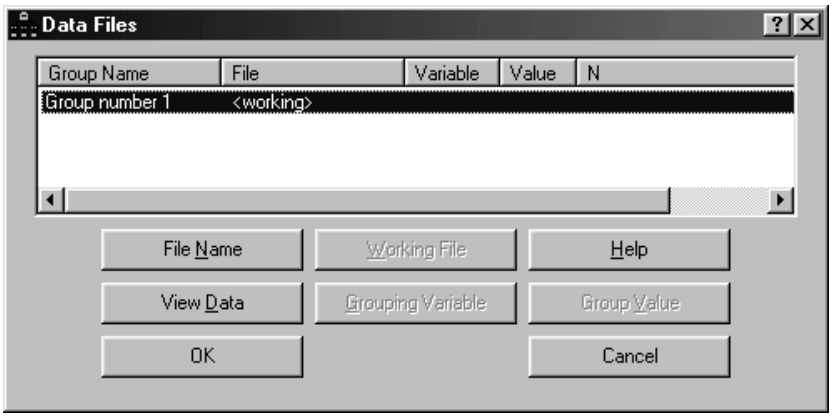


Estimating the Model in Amos Graphics

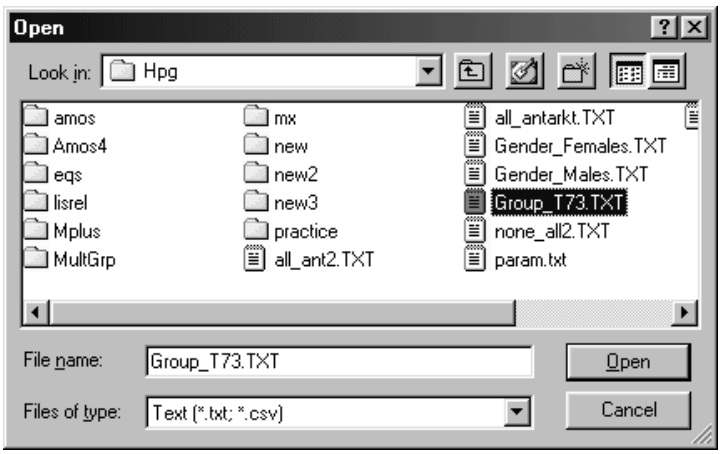
After the model has been created and edited in Amos Graphics the user may wish to estimate the model. One reason for doing this is to have the parameter estimates in the model for purposes of publication, or simply to use the convenient Amos Graphics modeling environment.

STREAMS can, however, not connect the name of the data file to be analyzed with the model, so the user has to do this manually. This is done in the following way:

First the **Data Files ...** option under the **File** menu is selected, which produces the following dialogue:

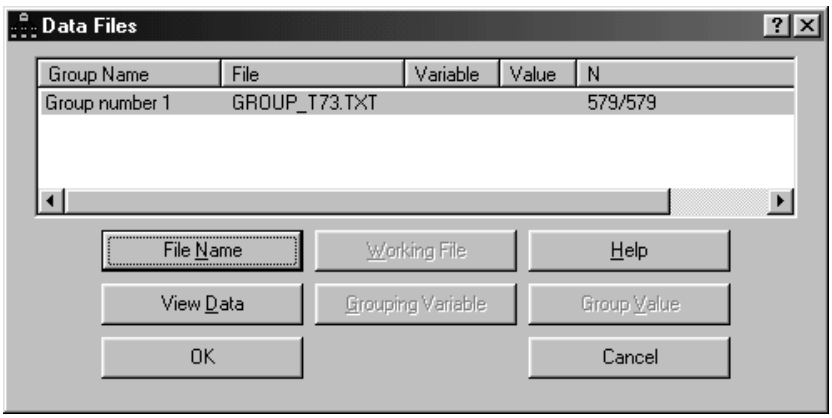


Then click the **File Name** button, which produces the standard file open dialogue:

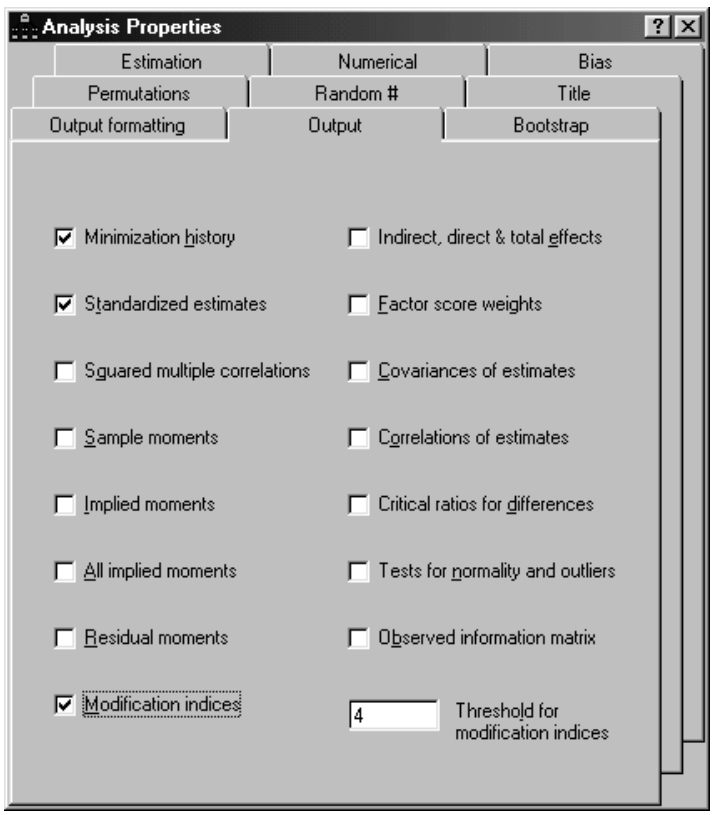


If the directory presented is not the directory where the project dictionary is located, this directory should first be located. The file name is constructed by combining the folder label and the dataset label and the suffix always is *.txt*, except for raw data files for which the suffix is *.sav*. In the current example the file name thus is *Group_T73.txt*.

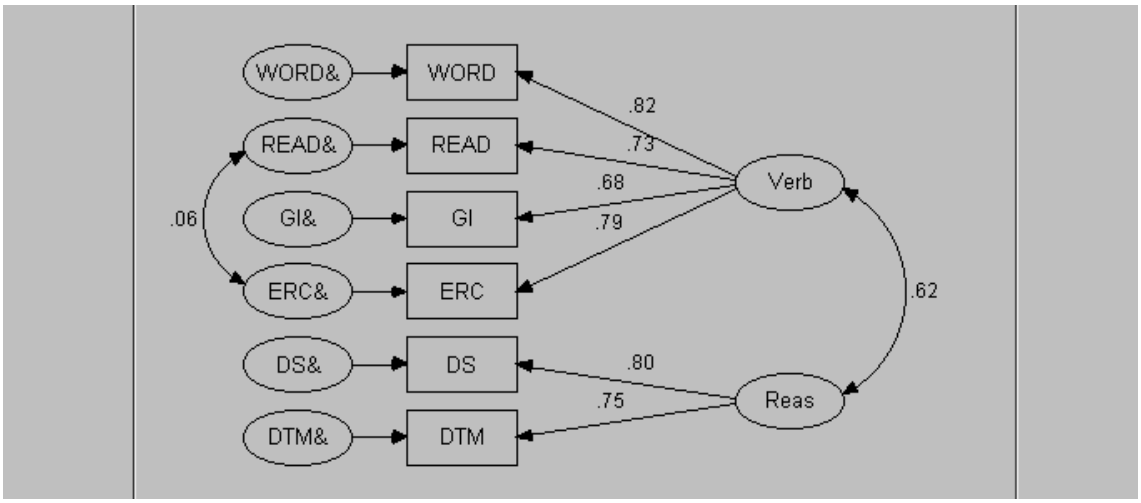
After the file has been opened, some descriptive information is presented in the Data Files dialogue (i. e., file name and N):



After the **OK** button has been clicked the estimates may be calculated by clicking the Calculate estimates icon (the abacus) on the Amos Graphics toolbar or through selecting the **Calculate Estimates** option under the **Model-Fit** menu. To select computation of standardized estimates or other options (e. g., modification indices) the **Analysis Properties** dialogue under the **View/Set** menu may first be used:



Having selected standardized estimates for presentation, this yields the following result:



The model may then be edited, and estimated anew.

Translating an Amos Graphic Model into the MB Language

If we want to have this model translated back into the MB language, in order, for example, to estimate it with a different SEM program than Amos, we can go back to STREAMS. This is done through clicking first the STREAMS program button on the Windows task-bar and then clicking the model window from which the current Amos model originates. When that is done, STREAMS interprets the Amos Graphics model file (here *o2.amw*) and translates it into MB statements. After the translation has been completed, STREAMS issues the message:

This message is obtained whether the model has actually been changed or not. The instructions from which the original Amos Graphics model was created are kept unchanged in the file name given in the message box. Should the changes made in Amos Graphics not be wanted, or incorrectly interpreted by STREAMS, the original instructions may easily be brought up again, just through opening the model.

In this case the following MB instructions are created by STREAMS from the edited

Amos Graphics model:y

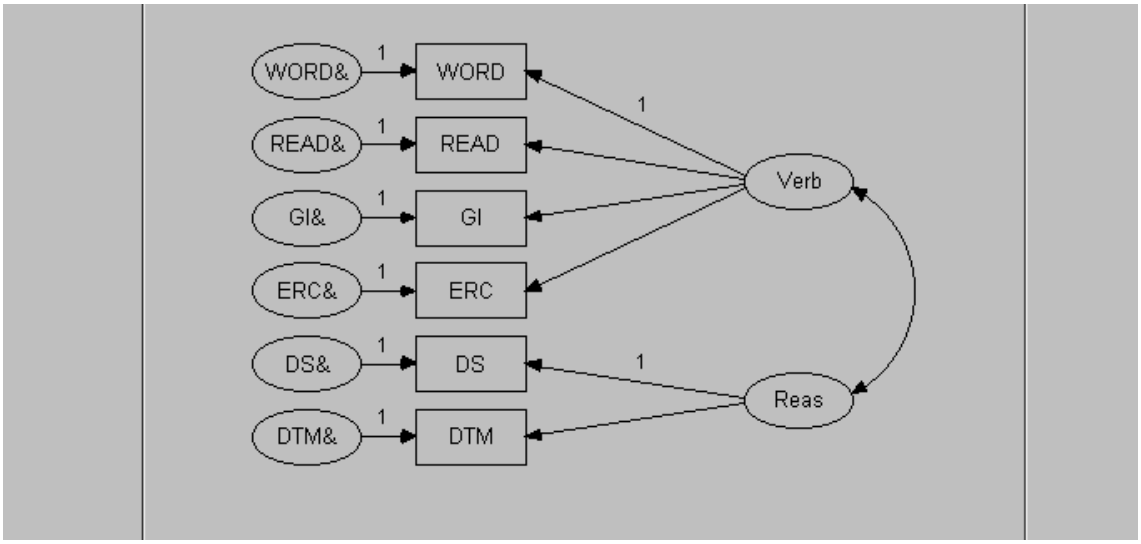
```
TI A two-factor model for the SweSAT subtests
MO PR=hpg NAME=o2
MO Create instructions for: Amos     Matrix: CM
MO Means included in model     One-group model
MO Amos 4.0
OP $ml;
DAT FOLDER=Group DATLAB=T73
MVR WORD READ GI ERC DS DTM
LVR Verb Reas
REL Verb -> WORD READ GI ERC
REL Reas -> DS DTM
COV Reas Verb
COV ERC& READ&
```

The covariance between the residuals of ERC and READ is thus added to the MB instructions.

Let us now assume that the *o2.mbi* model is edited in MB mode, for example, through deletetion of the line:

```
COV ERC& READ&
```

If the **Path diagram** button is clicked anew, the Amos Graphics *o2* model is edited, with the following result:



If there is already an Amos Graphics model with the same same as the MB model, the Amos Graphics model is edited in such a way that changes are made to the existing model. This is sometimes inconvenient, because the changes made in STREAMS may not fit well into the edited Amos Graphics model. In such instances it is recommended that the name of the MB model is changed into a new one, which causes STREAMS to create a new Amos Graphics model.

Using the procedures described above a model a model may thus be transferred back and forth between STREAMS and Amos Graphics. It should just be remembered that when

going from STREAMS to Amos Graphics the **Path Diagram** button must always be clicked, and when going from Amos Graphics to STREAMS, the STREAMS program button on the Windows taskbar should be clicked.

Examples of Models

Below some additional examples of simple one-group models are presented. One purpose is to show how different types of models, including some non-standard models, may be specified in the MB language. Another purpose is to show that there is often reason to consider using more than one estimation program. It is recommended that the user opens the projects, and tries the different models, and modifications of them.

The example project directories are available as compressed files in the STREAMS installation directory. Thus, first of all the compressed file should be decompressed, using the technique shown in Chapter 2. Every model has been estimated with all estimation programs (with a few exceptions), and the models are available in a separate subdirectory with the name of the estimation program (e. g., Amos, EQS, ...). After decompression of the compressed project file it is, of course, possible to delete those model directories which are of no interest.

Multivariate Regression Analysis: Ambition and Attainment

As has already been pointed out, regression analysis may be regarded as a special case of SEM, and such models may easily be specified with MB. The simplest form of regression analysis involves one or more manifest independent variables and a single dependent manifest variable. Regression analysis is applied in many different situations, and with different aims, but in a rough classification a distinction may be made between studies where the main purpose is to predict the dependent variable as well as possible (estimation of missing data values through imputation may be an example), and studies where the aim is to determine the relative amount of influence on the dependent variable due to different independent variables. In SEM the latter type of purpose tends to be the primary one. Estimating a regression model thus involves estimating the regression coefficients (γ_i) which express the change in the dependent variable associated with a unit increase in each independent variable, with all other independent variables held constant.

The term *multiple* in multiple regression analysis usually refers to the number of independent variables. It is easy to generalize regression analysis to allow multiple dependent variables, however, and we will consider a simple example.

The AMB project in *STREAMS\EXAMPLES\AMB* (or the compressed file *AMB.ZIP* in the Streams directory; see “Decompress”, page 175, for information how to unpack the file) contains data from a study by Kerchoff (1974), which comprised 767 12th-grade males. Among the variables measured were intelligence (INTEL), number of siblings (SIBS), father’s education (FEDUC), father’s occupation (FOCCUP), high-school grades (GRADES), educational expectations (EDEXP), and occupational expectations (OCCEXP). In the project there is one folder (*Grp*), and one dataset label (*Tot*).

These data have been reanalyzed by Kenny (1979) and Jöreskog and Sörbom (1989a, Ch. 4). The reanalyses have used the data to illustrate path analysis (see below) and this will be done here too, but first we will consider a simpler multivariate regression model.

A multivariate regression model includes one or more correlated manifest independent variables, just as an ordinary multiple regression model, and it includes two or more dependent variables, with correlated residuals. In the AMB project it is natural to regard GRADES, EDEXP and OCCEXP as dependent variables, and the other variables as independent variables. The MB statements for this model (*mreg.mbi*) are shown below:

MB instructions for the multivariate regression model for the AMB data

```
TI Multivariate regression with  GRADES, EDEXP and OCCEXP as dep vars
MO PR=amb NAME=mreg
MO Matrix: KM     One-group model
MO Means not included in model
MO LISREL DOS/Extender 8.14
OP OU ME=ML AD=OFF
DAT FOLDER=Grp DATLAB=Amb
MVR INTEL SIBS FEDUC FOCCUP GRADES EDEXP OCCEXP
REL INTEL SIBS FEDUC FOCCUP -> GRADES EDEXP OCCEXP
COV INTEL SIBS FEDUC FOCCUP
COV GRADES& EDEXP& OCCEXP&
```

This model may be estimated with all the programs, and the following post-processor output is obtained:

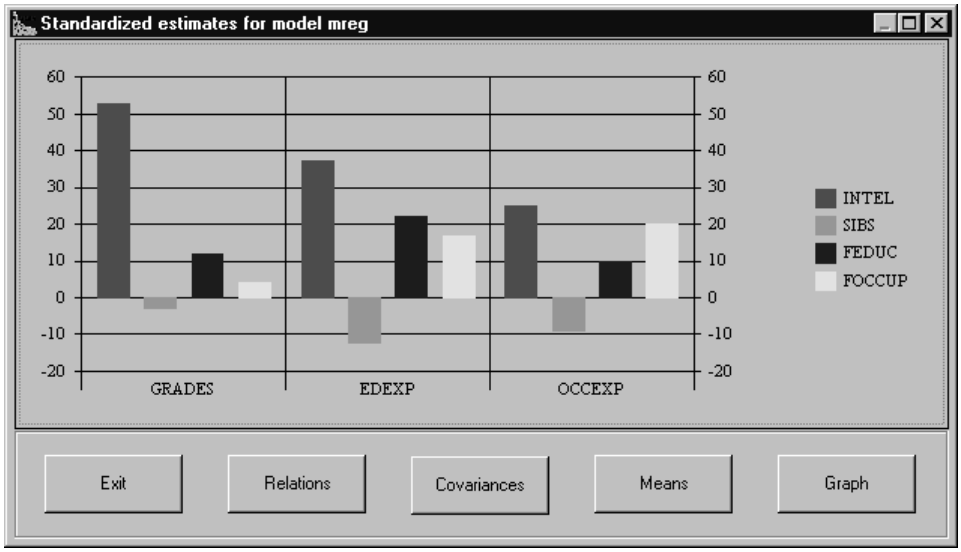
Post-processor output (edited) for the multivariate regression model

```
Goodness of Fit Test:

Chi-square = .00, df = 0, p < 1.00

Unstandardized estimates:

GRADES   =    +0.53*INTEL      -0.03*SIBS       +0.12*FEDUC
              +0.04*FOCCUP     +1.00*GRADES&
EDEXP    =    +0.37*INTEL      -0.12*SIBS       +0.22*FEDUC
              +0.17*FOCCUP     +1.00*EDEXP&
OCCEXP   =    +0.25*INTEL      -0.09*SIBS       +0.10*FEDUC
              +0.20*FOCCUP     +1.00*OCCEXP&

Var(GRADES)  =    1.00   Var(GRADES&) =    0.65   Expl var =   34.90 %
Var(EDEXP)   =    1.00   Var(EDEXP&)  =    0.62   Expl var =   37.65 %
Var(OCCEXP)  =    1.00   Var(OCCEXP&) =    0.81   Expl var =   19.30 %

Cov(EDEXP&,GRADES&)   =    0.26
Cov(OCCEXP&,GRADES&)  =    0.25
Cov(OCCEXP&,EDEXP&)   =    0.38
```

This model is just-identified, and it has zero degrees of freedom. It may also be demonstrated that a multivariate regression model of this kind can be estimated with identical results as three separate models in which the three dependent variables are entered one at a time.

Here a correlation matrix has been analyzed so the unstandardized estimates of the γ-coefficients are the same as the standardized estimates. Almost all coefficients are significant, and several of them are of considerable magnitude. It may also be noted, however, that there are some differences in the pattern of results for the three dependent variables: INTEL has, for example, its strongest relation with GRADES and its lowest relation with

OCCEXP, while FOCCUP has its strongest relation with OCCEXP and its lowest relation with GRADES. These patterns of results are more easily seen in the graph of standardized estimates (and particularly so in colour on the computer screen), which may be produced by first clicking the **Grid & Graph** button, and then the **Graph** button:



Path Analysis: Ambition and Attainment

In the multivariate regression example discussed above, the residuals of the three dependent were taken to be freely correlated. But it could also be argued that there is a logical ordering of the dependent variables such that grades affect educational and occupational aspirations, but not vice versa, and that educational aspirations affect occupational aspirations, but not the other way around. Kenny (1979, pp. 47-73) reanalysed the AMB project data under such assumptions of a causal or logical ordering of the dependent variables. The resulting model is a so called “path model” (see the path diagram

below). Path analysis was invented by the geneticist Sewall Wright (1917**,** 1934) as a technique to investigate the amount of influence exerted of one variable on another in a non-experimental situation.

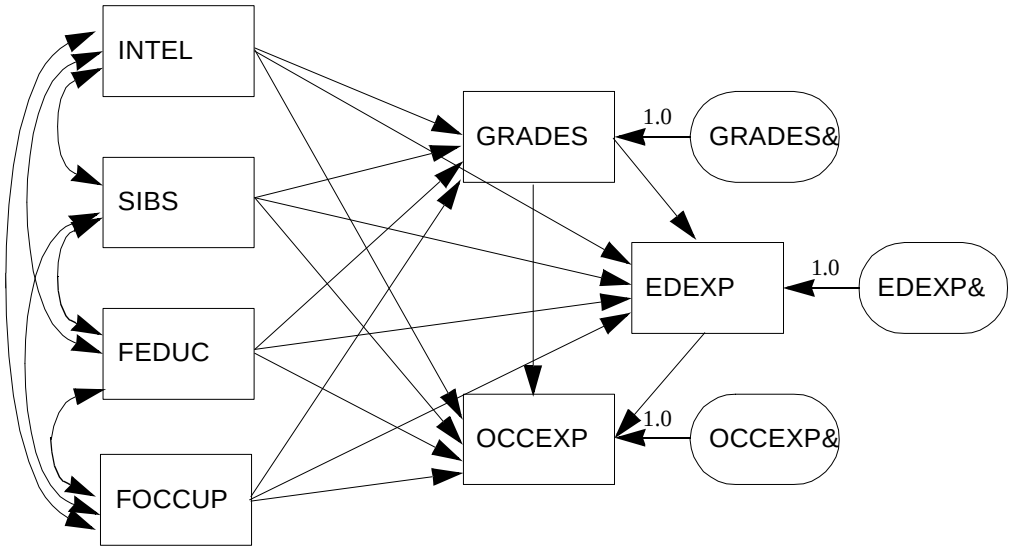


Figure 1. *A Path Model for the AMB data*

Specification and estimation of path models is straightforward in MB and all the structural equation modeling programs. In the *AMB* project there is a model labeled *path.mbi* which expresses the model in the MB language:

MB instructions for the path model for the AMB data

```
TI Path model the AMB data
MO PR=amb NAME=path
MO Matrix: KM     Means not included in model
MO One-group model     LISREL DOS/Extender 8.14
OP OU ME=ML AD=OFF
DAT FOLDER=Grp DATLAB=Amb
MVR INTEL SIBS FEDUC FOCCUP GRADES EDEXP OCCEXP
REL INTEL SIBS FEDUC FOCCUP -> GRADES EDEXP OCCEXP
REL GRADES -> EDEXP OCCEXP
REL EDEXP -> OCCEXP
COV INTEL SIBS FEDUC FOCCUP
```

In edited form the post-processor output from this model is:

Post-processor output (edited) for the path modelfor the AMB data

```
Unstandardized estimates:

GRADES   =    +0.53*INTEL      -0.03*SIBS       +0.12*FEDUC
               +0.04*FOCCUP     +1.00*GRADES&
EDEXP    =    +0.16*INTEL      -0.11*SIBS       +0.17*FEDUC
               +0.15*FOCCUP     +0.41*GRADES     +1.00*EDEXP&
OCCEXP   =    -0.04*INTEL      -0.02*SIBS       -0.04*FEDUC
               +0.10*FOCCUP     +0.16*GRADES     +0.55*EDEXP
               +1.00*OCCEXP&

Var(GRADES)  =    1.00   Var(GRADES&) =    0.65   Expl var =   34.90 %
Var(EDEXP)   =    1.00   Var(EDEXP&)  =    0.52   Expl var =   48.33 %
Var(OCCEXP)  =    1.00   Var(OCCEXP&) =    0.56   Expl var =   44.34 %


t-values:

GRADES   =   +17.21*INTEL      -1.01*SIBS       +3.17*FEDUC
              +1.10*FOCCUP    +19.57*GRADES&
EDEXP    =    +5.00*INTEL      -4.24*SIBS       +5.14*FEDUC
              +4.60*FOCCUP    +12.59*GRADES    +19.57*EDEXP&
OCCEXP   =    -1.17*INTEL      -0.68*SIBS       -1.16*FEDUC
              +2.87*FOCCUP     +4.30*GRADES    +14.65*EDEXP
             +19.57*OCCEXP&

Var(GRADES&) =   19.57
Var(EDEXP&)  =   19.57
Var(OCCEXP&) =   19.57
```

The present path model, which only involves manifest variables and which does not impose any restrictons on the observed covariance matrix, also is just-identified, so the goodness-of-fit test has zero degrees of freedom. In the path model there are no covariances among residuals because the pattern of relations among the dependent variables account for these. It may be noted, however, that the amount of explained variance tends to be higher in the path model than in the multivariate regression model.

In this model there is a rather weak direct relation between INTEL and EDEXP of .16, which may be compared with the considerably stronger relation of .37 in the multivariate regression model. But there is also an indirect effect of INTEL on EDEXP through GRADES. According to Wright's path rules (see, e. g., Loehlin, 1992, Ch. 1) the indirect effect of INTEL on EDEXP is the product of the γ-coefficients on the path in the path diagram between INTEL and EDEXP (i. e., .53 x .41 = .21). The sum of the direct effect and the indirect effect is the total effect. The total effects of INTEL on EDEXP thus is .16 + .21 = .37. Some of the estimation programs (e. g., EQS and LISREL) may be used to compute the direct and indirect effects.

In the path model the flow of causation is in one direction only, and there are no "loops" such that a variable is indirectly influencing itself. Such models are known as "recursive models" (see e. g., Jöreskog & Sörbom, 1989a, p. 121).

Confirmatory Factor Analysis: Nine Psychological Variables

We will next consider another example of a confirmatory factor analysis model, which is classical in the literature on factor analysis and structural equation modeling, namely the "nine psychological variables" data set from Holzinger and Swineford's (1939) study of the structure of human cognitive abilities (see also Gustafsson, 1998). This example has been discussed at length by, for example, Jöreskog & Sörbom (1989a, pp. 97-104; 1993c, pp. 23-27). It will be shown that several alternative models may be fitted to the data, and that the model modifications suggested by different estimation programs may be quite diverse.

The data comprise a subset of 9 tests from a test-battery of 26 tests administered to 145 7th- and 8th-grade children in the Grant-White school in Chicago. The 9 tests were selected to measure three hypothesized ability factors: *Visual Perception* ability, *Verbal* ability and *Speed*, with three indicators of each ability. The path model in the figure presents the labels of the tests, and the hypothesized pattern of relations between latent variables and manifest variables.

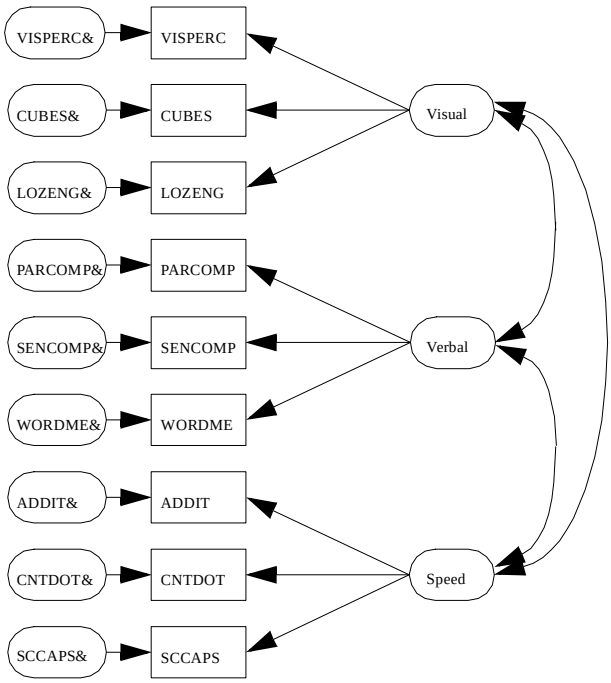


Figure 2. *The hypothesized model for nine psychological variables.*

It should be observed that the labels of some of the variables have been changed to conform to the syntactical requirements of MB. As may be seen from the path diagram the CFA model, just as the one considered earlier in the chapter, takes the latent variables to be independent variables and the manifest variables to be dependent variables in a kind of regression model. For each manifest variable there is thus a residual, which in the path diagram has been labeled according to the MB language conventions.

This model is an example of an oblique, simple-structure, model which was first introduced by Thurstone (1931, 1938, 1947) in the context of exploratory factor analysis. Such models also are referred to as congeneric models by Jöreskog (1971) and they are the first choice when measurement models are fitted.

A project labelled *NINE* has been created in the directory *STREAMS\EXAMPLES\NINE*.

The correlation matrix (see, e. g., Jöreskog & Sörbom, 1993c, p. 23) has been imported into this directory, with the labels shown in the path diagram. The folder is called *Group*, and the dataset label for the correlation matrix is *Grwh* (for Grant-White school). The models estimated for the different estimation programs are in directories beneath the *NINE* directory with the same name as the estimation program (e. g., Amos, EQS, ...), and within these directories the same model name is used for corresponding models.

The MB model for the path model shown above (*obl1*) may be specified in the following way:

MB instructions for the oblique three-factor model for nine psychological variables

```
TI Three oblique factors
MO PR=nine NAME=obl1
MO Means not included in model
MO Matrix: KM     One-group model
MO LISREL DOS/Extender 8.12
OP OU ME=ML AD=OFF MI SC
DAT FOLDER=Group DATLAB=Grwh
MVR VISPERC CUBES LOZENG PARCOMP SENCOMP WORDME ADDIT CNTDOT SCCAPS
LVR Visual Verbal Speed
REL Visual -> VISPERC CUBES LOZENG
REL Verbal -> PARCOMP SENCOMP WORDME
REL Speed -> ADDIT CNTDOT SCCAPS
COV Visual Verbal Speed
```

Observe that the COV statement must be included, because otherwise an orthogonal model will be generated. Estimating this model with Amos, EQS, LISREL, Mplus or Mx, yields essentially the same results. However, according to the goodness of fit test and the various fit indices the fit of the model to data is not so good. When LISREL is run the following results are obtained:

Results from the goodness of fit test for the oblique three-factor model.

```
Goodness of Fit Test:

Chi-square = 52.63, df = 24, p < .00

RMSEA = .091, p-value for RMSEA < 0.05 = .02

Fit Indices: GFI = .93, AGFI = .87, NFI = .89, NNFI = .91, CFI = .94

Maximum Modification Index is   25.1 for:
COV CNTDOT& ADDIT&
```

Both the χ^2 -test and the RMSEA-measure indicate that the model has an unacceptably poor fit.

The largest modification index (25.1) is obtained for the covariance between the errors of CNTDOT and ADDIT. It may be hypothesized that this covariance is due to an element of simple arithmetic in these tests, which both are highly speeded. Jöreskog and Sörbom (1989a, p. 101; 1993c, p. 26) observed that there is also another almost equally high modification index (24.7) which suggests that there should be a path between the SCCAPS test and the *Visual* factor. They hypothesized that such a relation may be accounted for by the fact that the SCCAPS test requires a Visual Perceptual ability in the rapid differentiation between straight and curved capitals.

The Mplus modification indices agree closely with those computed by LISREL. Amos also computes modification indices, but they tend to be somewhat different from those computed by LISREL. The following message is obtained when the oblique three-factor model (*obl1*) is fitted with Amos:

```
Maximum Modification Index is   15.30 for:
REL VISPERC -> SCCAPS
```

The multivariate LM test computed by EQS (with the GFV and PEE sets selected) suggests that two parameters should be freed: the CNTDOT& and ADDIT& covariance, and a covariance between LOZENG& and CUBES&. The latter two tests have (often along with the Flags test) frequently been found to identify a narrow spatial factor (S or SR-O, see Gustafsson, 1977; see also Carroll, 1993) which reflects the ability rapidly to rotate simple figures in three-dimensional space, so there may be a basis for a covariance between the errors of these tests. Several suggestions how to modify the originally hypothesized model are thus offered by the programs, and we will attempt some different alternatives.

To use the same approach as Jöreskog & Sörbom we just add an MB statement which specifies a relation from Visual to SCCAPS, i. e.:

```
REL Visual -> SCCAPS
```

When this model (*obl2*) is estimated the following goodness of fit statistics are computed by LISREL:

Results from the goodness of fit test for the modified oblique three-factor model (obl2l).

```
Goodness of Fit Test:

Chi-square = 28.86, df = 23, p < .18

RMSEA = .042, p-value for RMSEA < 0.05 = .57

Fit Indices: GFI = .96, AGFI = .92, NFI = .94, NNFI = .98, CFI = .99

Maximum Modification Index is    6.2 for:
COV LOZENG& CUBES&
```

The χ^2 -test now is non-significant and the RMSEA-measure is below the recommended critical value .05. It is also interesting to note that the modified model now identifies the covariance between LOZENG& and CUBES& as the second largest source of misfit, as was indeed done by the multivariate LM test computed by EQS for the *obl1* model. A further modification of the model may be made through adding the statement:

```
COV LOZENG& CUBES&
```

This model (*obl3*) has a χ^2 value which is about 6 units lower than the *obl2* model (χ^2 (22) = 22.94, p < .41, RMSEA=.017). The fit of this model is so good that no further modifications should be made.

As an alternative to invoking the relation between *Visual* and SCCAPS the covariance between CNTDOT& and ADDIT& might be allowed. When this is done and the model is reestimated we again get the message that the maximum modification index is due to COV LOZENG& CUBES&. If this covariance between errors is allowed as well (*obl4*)

the goodness-of-fit test is very close to the *obl3* model (χ^2 (22) = 22.95, p < .40, RMSEA = .017).

There is also a third approach to modify the poor-fitting original model. The modification indices and the multivariate LM test which identify covariances between the errors of manifest variables are in fact really saying the there is systematic covariance between manifest variables which is not accounted for by the latent variables. Another way to account for these covariances is to introduce further latent variables. As is shown by Hershberger (1994) models with correlated errors may be reformulated into so called equivalent models with additional latent variables, and vice versa. It has already been hypothesized that the covariance between CNTDOT& and ADDIT& is due to a numerical factor (*Num*), and that the covariance between LOZENG& and CUBES& is due to a spatial factor (*S*). We may thus introduce these latent variables in the following way (*obl5*):

MB instructions for the five-factor model for nine psychological variables

```
MVR VISPERC CUBES LOZENG PARCOMP SENCOMP WORDME ADDIT CNTDOT SCCAPS
LVR Visual Verbal Speed Num S
REL Visual -> VISPERC CUBES LOZENG
REL Verbal -> PARCOMP SENCOMP WORDME
REL Speed -> ADDIT CNTDOT SCCAPS
REL Num -> (ADDIT CNTDOT)
REL S -> (CUBES LOZENG)
COV Visual Verbal Speed
```

Two new latent variables (*Num* and *S*) are hypothesized, and these factors are assumed to be orthogonal to the three original latent variables. Constraints of equality are imposed on the relations between the new latent variables and the manifest variables, which is necessary because otherwise the model will be unidentified. When this model (*obl5*) is estimated it gives exactly the same value of the test-statistic as does the model with two correlated errors. The standardized estimates are presented in the table on the next page.

Standardized estimates for the five-factor model for nine psychological variables

```
Standardized estimates:

VISPERC  =    +0.74*Visual     +0.67*VISPERC&
CUBES    =    +0.41*Visual     +0.42*S          +0.81*CUBES&
LOZENG   =    +0.60*Visual     +0.42*S          +0.69*LOZENG&
PARCOMP  =    +0.87*Verbal     +0.50*PARCOMP&
SENCOMP  =    +0.83*Verbal     +0.56*SENCOMP&
WORDME   =    +0.82*Verbal     +0.57*WORDME&
ADDIT    =    +0.41*Speed      +0.60*Num        +0.68*ADDIT&
CNTDOT   =    +0.53*Speed      +0.60*Num        +0.59*CNTDOT&
SCCAPS   =    +0.97*Speed      +0.26*SCCAPS&


Cov(Verbal,Visual)    =    0.55
Cov(Speed,Visual)     =    0.66
Cov(Speed,Verbal)     =    0.38
```

The sequence of models fitted here very clearly demonstrates that it is possible to fit alternative models which equally well account for the relations among the observed variables, but which carry quite different interpretations and implications. Thus, the models which improve fit through allowing covariances between errors in manifest variables in a sense

make the source of misfit invisible, while the models in which additional latent variables are introduced make the sources of misfit very visible indeed. It is also interesting to observe that the different estimation programs tend to differ with respect to suggested modifications, and that, in particular Amos differs from the others. This suggests that it may not be wise to rely on one program alone.

Stability of Alienation

We will next consider a path-model for longitudinal data. The model by Wheaton, Muthén, Alwin, & Summers (1977) on the stability of alienaton is a another classical example in text-books on structural equation modeling. Data on scales designed to measure the constructs anomia and powerlessness were collected from a sample of 932 persons in 1966, 1967, and 1971, in order to study the stability of attitudes and their relation to education and occupation. The covariance matrix for a subset of the variables in this study is presented by, among others, Jöreskog & Sörbom (1993c, p. 29). These data have been imported into a project called *Alien*, which is available in the directory *STREAMS\EXAMPLES\ALIEN (*or in the compressed file *ALIEN.ZIP* in the *STREAMS* directory).

The hypothesized model is shown in the path-diagram:

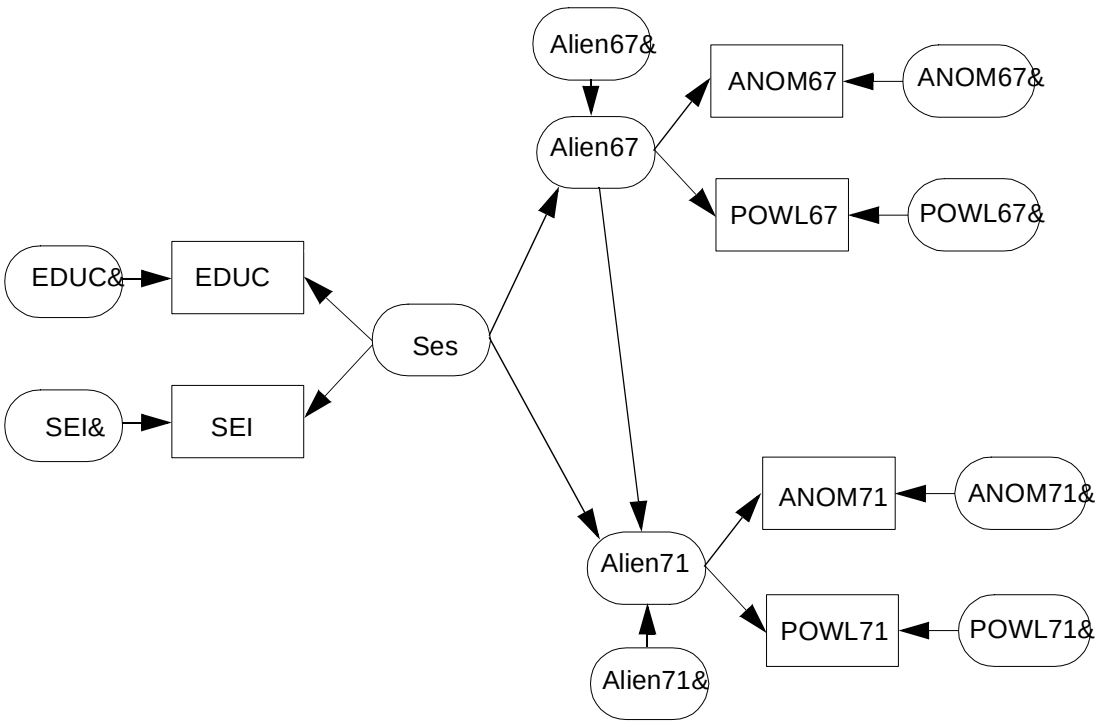


Figure 3. *The hypothesized model for stability of alienation*

It is hypothesized that there is a relation between alienation measured in 1967 and in 1971, and one of the research questions concerns how strong this relationship is. It is also hypothesized that the *Ses* variable affects alienation at both occasions of measurement.

The MB specification of this model is:

MB instructions for the hypothesized model for stability of alienation

```
DAT FOLDER=None DATLAB=Alien
MVR ANOM67 POWL67 ANOM71 POWL71 EDUC SEI
LVR Ses Alien67 Alien71
REL Ses -> EDUC SEI
REL Alien67 -> ANOM67 POWL67
REL Alien71 -> ANOM71 POWL71
REL Ses -> Alien67 Alien71
REL Alien67 -> Alien71
```

This model (*St1*) may easily be estimated with Amos, EQS, Mx, LISREL or Mplus. The fit of the model is not very good, however (χ^2 (6) = 71.47, p < .00, RMSEA = .108). The misfit is, according to the modification indices, caused by covariances among the errors of residuals in manifest variables, and particularly so for ANOM67& and ANOM71&. It is, of course, reasonable to expect an auto-regressive structure of relations among the systematic components of the residuals in the manifest variables over time. To account for these we may add the following two statements:

```
COV ANOM67& ANOM71&
COV POWL67& POWL71&
```

When this is done the model (*St2*) fits very well (χ^2 (4) = 4.73, p < .32, RMSEA = .014). The standardized estimates are presented below:

Standardized estimates from the model for stability of alienation with correlated errors

```
Standardized estimates:

Alien67  =    -0.56*Ses        +0.83*Alien67&
Alien71  =    -0.21*Ses        +0.57*Alien67    +0.71*Alien71&
ANOM67   =    +0.77*Alien67    +0.63*ANOM67&
POWL67   =    +0.85*Alien67    +0.52*POWL67&
ANOM71   =    +0.81*Alien71    +0.59*ANOM71&
POWL71   =    +0.83*Alien71    +0.55*POWL71&
EDUC     =    +0.84*Ses        +0.54*EDUC&
SEI      =    +0.64*Ses        +0.77*SEI&


Cov(ANOM71&,ANOM67&)  =    0.13
Cov(POWL71&,POWL67&)  =    0.04
```

The estimates presented here agree with those obtained by Jöreskog & Sörbom (1993c). About 50 % of the variance in *Alien71* is accounted for by *Alien67* and *Ses*. It should be noted that even though the relation between *Ses* and *Alien71* in absolute terms is lower than the relation between *Ses* and *Alien67*, this indicates that the effect of SES increases over time. This is because for *Alien71* there is both an indirect effect of *Ses* through *Alien67*, and a direct effect from *Ses*.

Covariances among errors of measurement over time is a standard procedure to account for the effects of due to repeated measuring instruments. It could be asked, however, if it would not be equally reasonable to have a relation from the residual in the manifest variable at the earlier time point to the manifest variable at the later time point. The relations over time between latent variables are typically specified as regressions, and the same

could be done with the residuals. An alternative specification would thus be:

```
REL ANOM67& -> ANOM71
REL POWL67& -> POWL71
```

This model *(St3)* may be estimated with EQS, Mx or LISREL, but not with the other programs. Amos regards the model as being unidentified. Mplus estimates the model, but because no distinction is made in the Mplus language between the residual and the corresponding variable, the intended model is not obtained. Instead Mplus estimates the regression onto the manifest variable itself, which is not what we want here.

The fit of the model estimated by the three succesful programs is identical with that of the model with correlated errors. Most of the parameter estimates are identical in the two models but some of them are not:

Standardized estimates from the model for stability of alienation with regression on errors

```
Standardized estimates:

Alien67  =    -0.56*Ses        +0.83*Alien67&
Alien71  =    -0.21*Ses        +0.57*Alien67    +0.71*Alien71&
ANOM67   =   +0.77*Alien67    +0.63*ANOM67&
POWL67   =    +0.85*Alien67    +0.52*POWL67&
ANOM71   =    +0.81*Alien71    +0.21*ANOM67&    +0.55*ANOM71&
POWL71   =    +0.83*Alien71    +0.07*POWL67&    +0.55*POWL71&
EDUC     =    +0.84*Ses        +0.54*EDUC&
SEI      =    +0.64*Ses        +0.77*SEI&
```

The residual variances in ANOM71 and POWL71 thus are smaller in this model than in the model with correlated errors. The main difference between the different ways of formulating the models thus seems to be conceptual: when the new tests are regressed on the residuals the earlier administration is seen as affecting performance on the instrument at a later occasion, while the correlated errors approach does not make any assumption about direction of effect.

Part 2

Specifying and Estimating Complex Models

The second part of the User's Guide shows how to specify and estimate some specialized, and often quite complex structural equation models with STREAMS. Chapter 4 deals with multiple-group models, Chapter 5 with missing-data models, and Chapter 6 with models for twolevel data. Novice users are adviced to skip these chapters and return to them as the need arises to deal with these specialized types of models. Chapter 7 is devoted to a short discussion of issues of efficiency in using structural equation modeling techniques.

4 Specifying Models for Multiple Groups

SEM is a powerful tool for analyzing data from multiple samples, because it allows investigation of group differences in a large number of respects, such as differences in measurement characteristics, differences in means on latent and manifest variables, and differences in strengths of relationship among variables within groups. Such models may easily be specified and estimated within the STREAMS environment.

The MB Language for Multiple-Group Modeling

In MB the user may choose to impose two different types of default equality constraints over groups, which apply if no other statements are made. One possibility is to have default constraints of equality imposed on every parameter over every group of cases. The other option is that no contraints of equality are imposed by default. The syntax of the MB language supports, furthermore, reference to group membership within the statements, which makes it easy to specify models which impose constraints, or relax constraints, on subsets of the parameters of the model.

If, for example, the default option of constraints over groups has been chosen, the following statement relaxes the constraints of equality on factor loadings for males and females:

```
REL Males Females Verb -> READ ERC WORD GI
```

If, to take another example, the default option of no constraints over groups has been chosen, the following statement imposes constraints of equality on factor loadings for males and females:

```
REL (Males Females) Verb -> READ ERC WORD GI
```

Below a concrete example is presented how to specify a two-group model, and what information is obtained from the program.

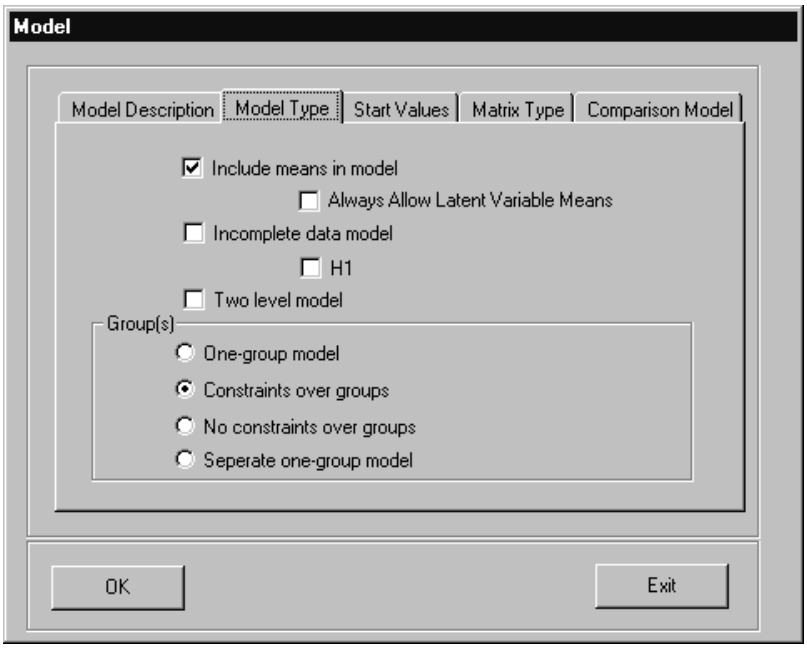
Specifying Multiple-Group Models

In most respects a multi-sample model is specified in the same way as a one-group model (see Chapter 3), except that path diagrams are not available for multiple-group models.

It is often a good idea to specify and estimate a one-group model in a first step, either for the pooled set of cases, or for one of the sub-samples. The one-group model is easier to estimate than is the more complex multi-sample model, and after it has been estimated, start values may be copied from this model to the multi-group model (see also Chapter 7). It is also a trivial task to transform a one-group model to a multiple-group.

Let us, as an example, take a starting point in the two-factor model for the SweSAT data specified in the previous chapter, and specify this model as a two-group model for males and femles. Thus, the first step is to open the *HPG* project. Then go down to the *MultGrp* directory and further down to the folder with the preferred estimation program. Then open the two-factor model, which is here called *vr1.mbi*. The name of the model should then immediately be changed, to *o2mf.mbi*, for example.

In order to specify the type of model the **Model** button should be clicked, and the **Model Type** tab clicked:



The *Group(s)* frame offers four radio buttons to identify type of model:

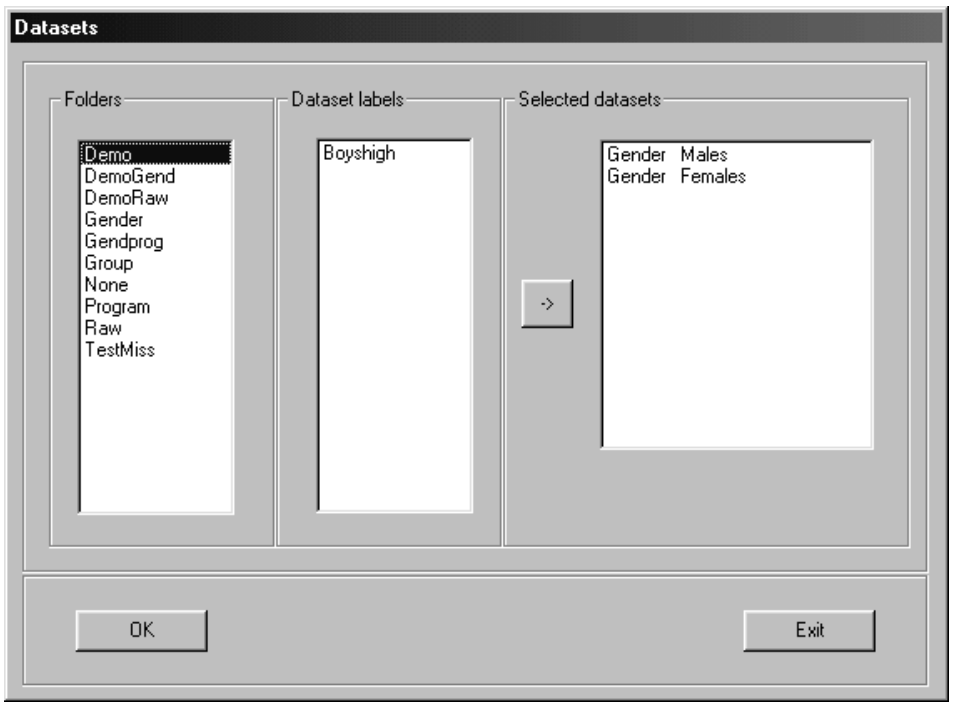
- **One-group model**
- **Constraints over groups**
- **No constraints over groups**
- **Separate one-group models**

The **Constraints over groups** option implies that by default every parameter is constrained to be equal over every group of cases, while the **No constraints over groups** option implies that no parameter is constrained to be equal over groups. The **Separate**

one-group models option implies that the pre-processor specifies as many models as there are groups of cases. This option is, however, only available when LISREL is used as the estimation program.

It is recommended that the **Constraints over groups** option is used in the first step, because a more highly constrained model is easier to estimate (see Chapter 7), and, at least if the fit is good, a more constrained model is to be preferred over a less constrained model. This is also the option chosen here. The default option to **Include means in model** is also chosen.

In order to transform the one-group model to a two-group model we also need to remove the DAT statement which refers to the pooled group of cases and instead include DAT statements which refer to covariance matrices for males and females, respectively. This is done through clicking the DAT button which produces the **Datasets** form. First the *T73* group is removed from the **Selected datasets** box. Then the *Gender* folder is clicked, and the *Males* and *Females* groups selected and transferred to the **Selected datasets** box:



When the **OK** button is clicked the following MB statements are generated:

MB instructions for a completely constrained two-group model

```
TI A two-factor model for the SweSAT subtests by gender
MO PR=hpg NAME=o2mf
MO Create instructions for: LISREL Y-model     Matrix: CM
MO Means included in model     Multiple groups with constraints
MO LISREL DOS/Extender 8.20
STA NAME=h:\fkod\examples\hpg\Amos\g1ml.mbi
OP OU ME=ML AD=OFF XM
DAT FOLDER=Gender DATLAB=Males
DAT FOLDER=Gender DATLAB=Females
MVR DTM DS WORD READ GI ERC
LVR Reas Verb
REL Verb -> WORD READ GI ERC
REL Reas -> DS DTM
COV Reas Verb
```

It will be remembered that the fit of the oblique two-factor model was quite excellent for the total set of data. However, the two-group model has a very poor fit (χ^2 (35) = 138.62, $p < .00$, RMSEA = .072). This indicates that there are differences between males and females with respect to one or more of the parameters of the oblique two-factor model. The differences could, however, pertain to one or more of the different parameters of the model, such as means of latent variables, intercepts of manifest variables, variances of latent variables, residual variances of manifest variables, and/or covariances among latent variables. In order to clarify in what respects the models for males and females differ it is necessary to investigate models which impose fewer constraints of equality over groups.

Testing Differences Between Groups

An overall test of equality of the models for males and females is obtained if the fit of the model which imposes full constraints over groups is compared with the model which imposes no constraint of equality. The test-statistic for the model with no constraints of equality (*o2mf0.mbi*) over groups is χ^2 (16) = 28.94, $p < .02$, RMSEA = .037, which implies that the difference test is highly significant (difference χ^2 (19) = 109.68).

In order to investigate more closely in what part of the model the gender differences are located the following sequence of successively more relaxed models may be fitted (cf Gustafsson, 1997):

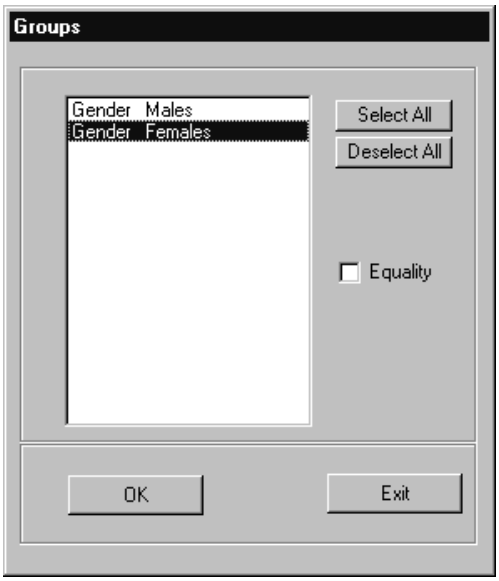
1. No constraints on latent variable means.
2. No constraints on intercepts of manifest variables (i. e., remove means from model altogether).
3. No constraints on variances of residuals in manifest variables.
4. No constraints on variances of latent variables.
5. No constraints on covariances of latent variables.
6. No constraints on relations between latent and manifest variables.

Between each of these steps a difference χ^2 test may be computed to assess the significance of the group difference.

In order to relax the constraint of equality on the latent variable means, the following statement may be added to the *o2mf* model (model *o2mf1*):

```
MEA Females Verb Reas
```

This may be done in several different ways, such as entering the command line via the keyboard. Another possibility is to click the **MEA** button and select the *Verb* and *Reas* variables. The **Groups** button on the Means form is then clicked, which produces the *Groups* form. On this form the *Females* group should be selected:



After the **OK** button is clicked on the *Groups* form, as well as on the *Means* form, the command line is written into the editor area.

It should be observed that the mean cannot be estimated in both groups, and the program assumes by default that the mean of all latent variables is zero in the first group (here the Males group). Should, however, reference be made to the first group in a MEA statement, the program takes no action unless the check box **Always allow latent variable means** on the **Model Type** tab of the *Model* form has been clicked. When the check box has been clicked the program allows estimation of latent variable means in the first group of multiple-group models, as well as in one-group models. This option must be used when certain kinds of models, such as growth-curve models, are specified.

The test statistic for model *o2mf1* is χ^2 (33) = 68.63, p < .00, RMSEA = .043, so this model fits considerably better than does the compeletely constrained model *o2mf* (difference χ^2 (2) = 69.99) (observe, however, that the different programs tend to yield marginally different χ^2 statistics; from Mplus the value 69.0 thus was obtained). However, the model which only relaxes constraints of equality over groups on the latent variable means does not fit as well as does the model without any constraints of equality over groups (difference χ^2 (17) = 39.69). It may thus be concluded that there are significant differences with respect to the means for one or both of the two latent variables, and that there also are

smaller differences between one or more of the other parameter estimates of the models for males and females.

In step 2 of the recommended sequence constraints on means are removed altogether. This can be done in different ways: One possibility is to uncheck the check box **Include means in model** on the **Model Type** tab of the *Model* form and remove the MEA statement from the model. Another possibility, which produces equivalent results, is to replace the MEA statement which refers to the latent variables with one which refers to all the manifest variables (or their residuals, which also gives the same results):

```
MEA Females WORD DS READ DTM GI ERC
```

This statement may be constructed with the forms or through using the editor.

This model (*o2mf2*) has a χ^2 of 53.54, with 29 df, which is significantly better than the *o2mf1* model (difference χ^2 (4) = 15.09). Thus, over and above the gender difference with respect to latent variable means there is a gender difference with respect to the intercepts of one or more of the manifest variables. It should be noted, however, that the differences with respect to intercepts seem trivially small compared to the difference with respect to the latent variable means.

According to the RMSEA measure both models which relax constraints on the means for males and females have an acceptable degree of fit. We could, thus, stop the investigation of group differences here, concluding that there are gender differences with respect to level of performance on the SweSAT, but not with respect to other characteristics. However, in order both to illustrate the procedures of multi-group modeling, and to account for the remaining gender difference, we will continue testing for group differences.

In step 3 constraints are removed with respect to error variances. This may be done through adding the following statement to model *o2mf2* (model *o2mf3*):

```
VAR Males Females WORD& DS& READ& DTM& GI& ERC&
```

This statement may, for example, be constructed through clicking first the **VAR** button, selecting the residual variables, then clicking the **Groups** button on the *VAR* form and selecting all groups.

The test statistic for this model is χ^2 (23) = 36.76, p < .03, RMSEA = .032, which implies a significant improvement of fit (difference χ^2 (6) = 16.78). It may, thus, be concluded that for one or more of the six residuals in manifest variables there is a significant gender difference with respect to variance.

In step 4 gender differences with respect to the variance of the latent variables is investigated, through adding the following statement (model *o2mf4*):

```
VAR Males Females Reas Verb
```

The test statistic of this model is χ^2 (21) = 36.35, p < .02, RMSEA = .036, which implies that there is no significant difference with respect to variances in latent variables for males and females (difference χ^2 (2) = 0.41).

In step 5 gender differences with respect to the covariance among the latent variables is investigated, through transforming the statement COV Verb Reas to the following statement (model *o2mf5*):

```
COV Males Females Verb Reas
```

This model achieves almost the same value on the test statistic as did the previous model (χ^2 (20) = 36.34, p < .01, RMSEA = .038), so it may be concluded that there is no gender difference with respect to the covariance between the two latent variables.

In the final step we may test the homogeneity of regressions of the manifest variables on the latent variables, through changing the two REL statements in the following way (model *o2mf6*):

```
REL Males Females Verb -> WORD READ GI ERC
REL Males Females Reas -> DS DTM
```

The test statistic of this model is χ^2 (16) = 28.94, p < .02, RMSEA = .037, which is, of course, the same as the completely unconstrained model (*o2mf0*). The difference between this model and the previous model is not significant (difference χ^2 (4) = 7.40), so it may be concluded that there are no gender differences with respect to the relations between manifest and latent variables of the model.

Output from Multiple-Group Models

The output from multiple-group models may be presented in several different ways: as a listing file (post-processor output), in a grid and as a graph, and some of these options will be illustrated below. Results will be presented from model *o2mf3*, which fits as well as the model without any constraints.

The post-processor listing output from multiple-group models is arranged in the same way as output from one-group models (i. e., goodness-of-fit information, unstandardized estimates, t-values, and standardized estimates), except that the results from different groups are presented adjacently, to allow easy comparison. The table below presents unstandardized estimates of the relations between manifest and latent variables:

Unstandardized estimates of relations from the o2mf4 *two-group model for the SweSAT*

```
Unstandardized estimates:


Males    DTM      =    +0.74*Reas       +1.00*DTM&
Females  DTM      =    +0.74*Reas       +1.00*DTM&

Males    DS       =    +1.00*Reas       +1.00*DS&
Females  DS       =    +1.00*Reas       +1.00*DS&

Males    WORD     =    +1.00*Verb       +1.00*WORD&
Females  WORD     =    +1.00*Verb       +1.00*WORD&

Males    READ     =    +0.71*Verb       +1.00*READ&
Females  READ     =    +0.71*Verb       +1.00*READ&
```

The estimates in the two groups are all identical, which is, of course, because of the constraints of equality. It should also be pointed out that even though there are differences between the coefficients for different variables these are not interpretable because they are influenced by the scales of the manifest variables.

The unstandardized estimates of residual variances are presented below:

Estimates of residual variances for males and females from the o2mf3 *two-group model*

```
Males    Var(DTM)     =    7.94   Var(DTM&)    =    3.17  Expl var =   60.06%
Females  Var(DTM)     =   10.51   Var(DTM&)    =    5.74  Expl var =   45.41%

Males    Var(DS)      =   13.44   Var(DS&)     =    4.73  Expl var =   64.78%
Females  Var(DS)      =   14.55   Var(DS&)     =    5.85  Expl var =   59.82%

Males    Var(WORD)    =   23.47   Var(WORD&)   =    8.18  Expl var =   65.15%
Females  Var(WORD)    =   22.39   Var(WORD&)   =    7.10  Expl var =   68.29%

Males    Var(READ)    =   13.84   Var(READ&)   =    6.10  Expl var =   55.90%
Females  Var(READ)    =   13.64   Var(READ&)   =    5.90  Expl var =   56.72%

Males    Var(GI)      =   13.37   Var(GI&)     =    7.28  Expl var =   45.54%
Females  Var(GI)      =   13.93   Var(GI&)     =    7.84  Expl var =   43.71%

Males    Var(ERC)     =   16.48   Var(ERC&)    =    5.87  Expl var =   64.35%
Females  Var(ERC)     =   16.80   Var(ERC&)    =    6.20  Expl var =   63.11%
```

It may be observed that for most of the variables the residual variances are quite similar. For DTM, however, the residual variance is higher for females than for males, and there is a tendency in the same direction for DS. It would, of course, be easy to conduct statistical tests of the equality of these parameters but it must also be warned that such tests need to be replicated on a new sample, because the significance levels are disturbed by the large number of tests being conducted.

The *o2mf3* model does not include the means of the manifest variables, so to see the estimates of the group differences in latent variable means we need to consider instead the *o2mf1* model (or a modified version of this model which takes into account the other differences found). The results with respect to the latent variable means, in edited form, are presented below:

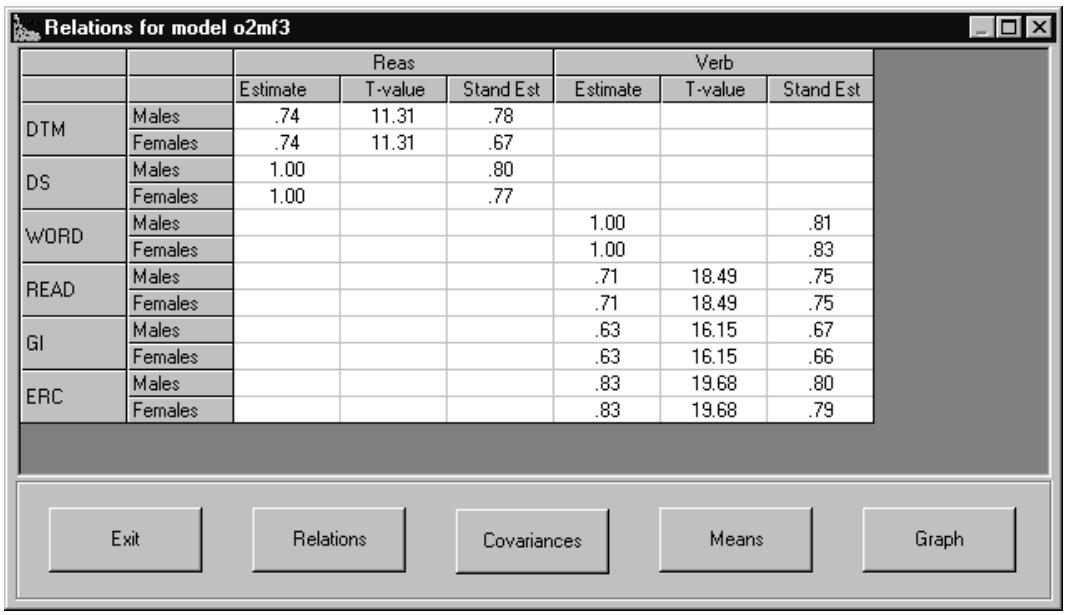
Estimates of latent variable means for males and females from the o2mf1 *model*

```
Unstandardized estimates:
...
Females  Mean(Reas)          =   -2.46
Females  Mean(Verb)          =   -0.97
...
t-values:
...
Females  Mean(Reas)          =   -8.22
Females  Mean(Verb)          =   -2.74
...
Standardized estimates:
...
Males    Mean(Reas)          =    0.00
Females  Mean(Reas)          =   -0.84

Males    Mean(Verb)          =    0.00
Females  Mean(Verb)          =   -0.25
```

For both latent variables the Female group has a lower estimated mean than has the Male group because the estimates are negative. These differences are, however, expressed on a scale which does not allow comparison between the latent variables and which is difficult

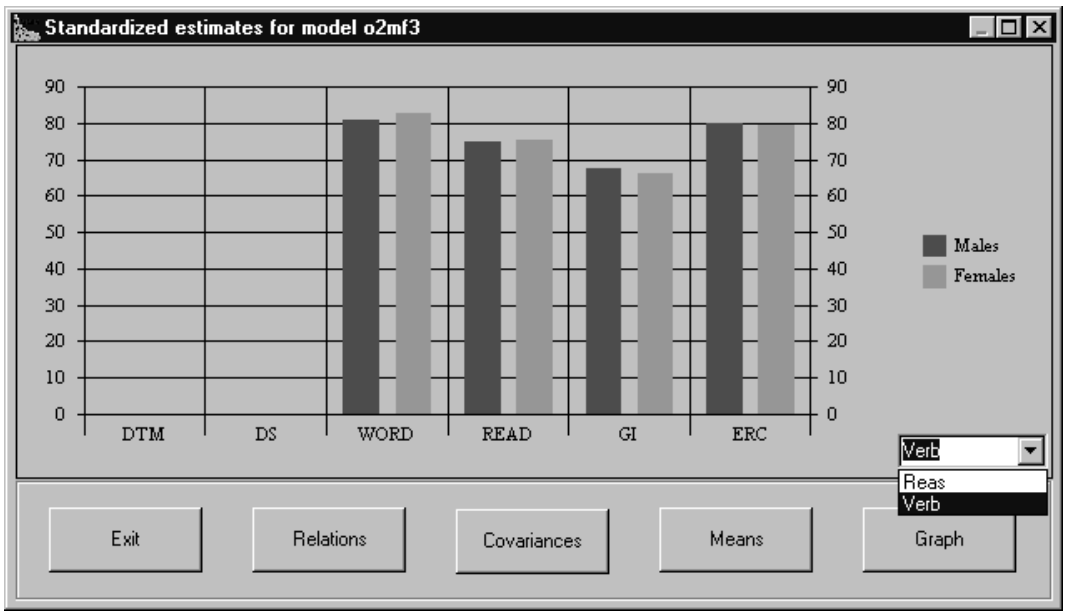
to interpret. The t-values indicate that there is a significant difference with respect to both latent variables, but most highly so for the *Reas* variable. The standardized estimates present the group differences in terms of the pooled within-group standard deviations (or z-scores). For the *Reas* variable there is a considerable difference which amounts to no less than .84 sd units, while for the *Verb* variable there is a smaller difference of .25 sd units. These differences seem to a large extent to be due to differential processes of selection among males and females into the group of test-takers.

Let us now return to the *o2mf3* model again. When the **Grid & Graph** button on the post-processor tool-bar is clicked the results are presented in a grid instead, which presents the unstandardized estimates, t-values, and standardized estimates at the same time:



		Reas			Verb		
		Estimate	T-value	Stand Est	Estimate	T-value	Stand Est
DTM	Males	.74	11.31	.78			
	Females	.74	11.31	.67			
DS	Males	1.00		.80			
	Females	1.00		.77			
WORD	Males				1.00		.81
	Females				1.00		.83
READ	Males				.71	18.49	.75
	Females				.71	18.49	.75
GI	Males				.63	16.15	.67
	Females				.63	16.15	.66
ERC	Males				.83	19.68	.80
	Females				.83	19.68	.79

Clicking the **Graph** button presents a graph of the standardized estimates for all the groups for one latent variable at a time:



The drop-down menu may be used to switch from one variable to another. It should be observed that the standardized estimates are not exactly equal in spite of the fact that constraints of equality were imposed on the unstandardized estimates. This is because the standardization is done in such a way that the standard deviations of the manifest variables is taken into account, and with respect to these there are some differences between males and females.

5 Specifying Models for Incomplete Data

Normally SEM requires valid data on every variable for every case. However, often the data is incomplete in the sense that observations on one or more variables are missing for a smaller or larger proportion of the cases. There are many ways to deal with missing data, and one of these, which has appeared in recent years, is to take the missingnes into account in modeling of the data. Such procedures are now becoming available in SEM programs. Thus, Amos, EQS, Mplus and Mx offer procedures which compute ML estimates from rawdata with missing observations, and the programs may also be used to compute ML estimates from covariance matrices for groups of cases with different patterns of observed variables. This chapter describes how these procedures may be used in STREAMS. It should be observed that the modelling techniques presented in this chapter tend to be somewhat complex, so the novice SEM user is adviced to skip this chapter until the need arises to deal with missing data.

Types of Missing Data and Methods of Solution

It may be noted that missing data may occur for different reasons, but a basic distinction is between *structurally* missing data, and *accidentally* missing data. Structural missingness (or missingness by design) is the consequence of decisions not to observe all variables for all subjects, such as when different subsets of cases are given partially different sets of tasks, or when a longitudinal design is used in which a subset of cases only is followed up. Accidental missingness occurs when the planned set of observations could not be obtained for reasons such as non-response or coding errors, just to mention two examples. These two types of missing data should be dealt with in different ways.

There are five basic procedures for dealing with missing data:

- *Modeling incomplete data.* The modeling approach implies that the estimation algorithms in structural equation modeling are adapted to deal with missing observations. Until recently no practical modeling procedures were available, but now Amos, Mplus and Mx offer procedures which compute ML estimates from rawdata with missing observations (this procedure will be referred to as the *rawdata-based* procedure of missing-data modeling), and Amos, LISREL and Mx may all be used to compute ML estimates from covariance matrices for groups of cases with different patterns of observed variables (Allison, 1987; Muthén, Kaplan, & Hollis, 1987). The latter approach requires a separate covariance matrix and mean vector for each subset of cases with a particular pattern of missing observations, and a special multigroup specification (this procedure will be refered to as the *matrix-based* procedure; this procedure is implemented internally into Mplus). STREAMS offers full support for this approach, through an automatic procedure for computing separate matrices for all existing combinations of missing values, and through automatic specification of the multigroup model for Amos, LISREL and Mx. Because accidental missing data tend to generate a very large number of different missing data patterns, this approach works best for structurally missing data, while the procedure built into Amos, Mplus and Mx may be most useful for accidental missing data.
- *Estimation of the complete covariance matrix.* Recently techniques have been developed which provide maximum likelihood estimates of the covariance matrix with the so called EM algorithm (Little & Rubin, 1987). There are several programs available for obtaining such a covariance matrix (e. g., the EMCOV program, written by John Graham at Pennsylvania State University). After the covariance matrix has been estimated, it may be imported into STREAMS using methods described in Chapter 10, and modeled in the same way as any matrix based on complete data.
- *Imputation.* This procedure implies that missing observations are replaced with estimates of likely observed values. There are numerous imputation procedures to choose among, and some are implemented in EQS, SPSS and PRELIS. The simplest procedure is to replace missing observations with the mean for the variable. This procedure is available in STREAMS, and one or more stratification variables may be used to do the imputation for subgroups of cases (see Chapter 9). But other procedures are also available (see, e. g., Little & Rubin, 1987), in which the information available in the data is used to predict the missing scores. It is recommended that imputation is used to solve the problem of (limited extents of) accidental missing data, while structurally missing data is probably best dealt with through the modeling approach. It should also be emphasized that the imputation techniques are based on strict assumptions of randomness of missingness, and that they cause variances and covariances for the variables with many imputed values to be underestimated. There may be reason to quote a word of warning from Dempster and Rubin (1983) here:

  > *The idea of imputation is both seductive and dangerous. It is seductive because it can lull the user into the pleasurable state of believing that the data are complete after all, and it is dangerous because it lumps together situations where the problem is sufficiently minor that it can be legitimately handled in this way and situations where standard estimators applied to the real and imputed data have substantial biases.*

- *Pairwise deletion.* This procedure implies that only those combinations of variables for which one or both observations are missing for a case are excluded when matrices are computed. PRELIS 2 offers this procedure when computing certain matrices (e. g., covariance matrices without asymptotic matrices). One problem with this approach is uncertainty about the number of cases on which the matrix has actually been computed, and another problem is that the matrix may occasionally not be possible to analyze, because it is not positive definite.
- *Listwise deletion.* This procedure implies that cases with missing observations on one or more variables are excluded from computations. STREAMS and PRELIS offer this method for dealing with missingness when matrices are computed, and data imported. The problem with this approach is that a large proportion of the sample may be excluded when there are many accidentally missing observations and many variables are analyzed. When there are structurally missing observations this approach may not be at all applicable. It is, of course, also likely that the cases with complete data are systematically different from the cases for which one or more variables are missing. Simulation studies indicate, furthermore, that listwise deletion is often an inefficient method for dealing with missing data (see Roth, 1994, for a review; Wothke, in press).

Often the best solution is to apply different missing data treatment methods in combination. Thus, when the modeling approach is used to solve the problem of structurally missing observations, the accidentally missing observations should first be replaced with imputation techniques. It is, of course, also possible to combine imputation methods and listwise deletion in such a way that cases with many missing values are excluded, while cases with few missing values are retained.

There is also reason to make a distinction in statistical terms between different missing-data mechanisms. Data on a particular variable Y may, thus, be “missing completely at random” (MCAR), which means that those cases which lack scores on Y are no different from cases who have scores on Y (Rubin & Litte, 1987). When the MCAR assumption is valid listwise deletion, pairwise deletion, and the simple imputation methods described in Chapter 9 provide correct parameter estimates (but generally not correct standard errors or goodness-of-fit tests). However, data on Y may also be “missing at random” (MAR), which means that the pattern of missingness is random given a set of other variables. Thus, if the sample (in theory at least) can be divided into subsamples on the basis of scores on other varables and the MCAR assumption holds within these groups, the data are MAR. When the data are MAR, but not MCAR, the procedures of listwise deletion, pairwise deletion, and simple imputation lead to biased estimates, as well as incorrect estimates of standard errors and goodness-of-fit statistics.

The modeling procedures are based on the assumption that data are MAR, and when this assumption (along with the other assumptions) is true, unbiased estimates and correct standard errors are obtained. The MAR assumption is considerably less restrictive than is the MCAR assumption, but it is not easy to test the validity of the MAR assumption in any particular situation (see Little & Rubin, 1987). However, even though the MAR assumption is not easily tested, and even though it may be incorrect in many situations, the MAR estimates may, nevertheless, be expected to be less biased than estimates which rely on the MCAR assumption.

But even though the modeling of incomplete data with maximum-likelihood techniques represents major progress, these procedures have some disadvantages as well. Thus, both

the rawdata-based estimation procedures in Amos, Mplus and Mx, and the matrix-based procedure are computationally much more intensive than are the ordinary estimation methods. When the sample is large this is, in particular, true for the rawdata-based procedure, and when there are many patterns of missing observations the matrix-based procedure becomes tedious because there is one matrix for each pattern. Another problem, is that the model specification is complex and lengthy, which has prevented general application of the matrix-based procedure since it was first developed by Allison (1987) and Muthén, Kaplan, & Hollis (1987). With STREAMS most of these problems are solved, but it is still a fact that the model specifications may become quite lenghty. A third problem is that many of the standard test statistics och diagnostic tools are not easily available when modeling incomplete data. Thus, the goodness-of-fit test is more difficult to compute when modeling incomplete data, than when modeling complete data, and the descriptive fit statistics, as well as the modification indices, can normally not be obtained. It is, thus, in most cases advisable to combine the simpler MCAR-assuming deletion and imputation techniques with the maximum-likelihood etimation methods, in such a way that the simpler technqies are used in the first steps of modeling, and the more complex techniques are applied in the later modeling phases.

The Rawdata-Based Estimation Procedures

When rawdata with missing data are available, Amos and Mx can employ special maximum-likelihood estimation procedures (Arbuckle, 1996, 1997; Neale, 1995) which use all the information available in the record of each case. Mplus too estimates missing data models directly from rawdata, through using all the observed missing data patterns. As long as missing data codes are defined in the raw data input files, STREAMS uses this information to automatically specify missing data models for Amos 4, Mplus and Mx. In Amos 3.6 the missing-data estimation procedure is invoked when the *$missing = code* directive is supplied. Here *code* is a value (e. g., -1), common to all variables, which indicates a missing value. In STREAMS this directive is available on the **Input** tab of the *Amos Options* form. In other respects the model is specified as an ordinary model. In Mx the Raw Maximum Likelihood (RM) procedure is invoked whenever raw data is input. On the **Input** tab of the *Mx Options* form it is also possible to supply a missing data code (e. g., -1.00) common to all variables. Observe that in Mx the missing data code must be entered exactly as it appears in the raw data. This implies that if -1.0 is given as the missing code, the data values -1 and -1.00 will be used as valid data. It is thus essential that the raw data is inspected and that the correct code string is entered.

A missing-data model estimated with the rawdata-based procedure does not, however, yield the standard χ^2 goodness-of-fit test unless some further computations are made. Amos 3.6 and Mx only present the minimum of the fitting function, along with the number of estimated parameters, and when comparing two models this information may be used to compute a difference test, which yields the ordinary χ^2 difference test. When a model is compared to the “saturated” model or “H1-model” (i. e., a model which estimates all variances and covariances for a set of observed variables, and where thus the number of estimated parameters is the same as the number of elements in the covariance matrix) the ordinary χ^2 goodness-of-fit test is obtained. Thus, to obtain this test two models must be run. The steps involved in this procedure are described in greater detail below. When Mplus is used it is also necessary to compute the H1 model to obtain the χ^2 goodness-of-

fit test, but this is taken care of automatically by Mplus when the H1 model is requested.

Preparing Rawdata for Analysis

In order to use the rawdata estimation procedures the rawdata must be imported into a STREAMS project, using the procedures described in Chapter 9. This is true also for the SPSS version of Amos 3.6, which thus cannot access SPSS *.sav* files when run under STREAMS. However, STREAMS can access the *.sav* file and import the data into the project.

Before the data is imported there is reason, however, to make some preparations. Thus, variable labels should be changed so that they are at most 7 characters long (or 6 characters if two-level models are to be fitted).

Estimating the Saturated Model

In order to obtain the standard χ^2 goodness-of-fit test it is necessary first to run the so called "saturated" model, or the "H1-model" (Muthén, Kaplan & Hollis, 1987). This is a model which simply fits all variances and covariances for a set of manifest variables.

The process of estimating the saturated model is best described with an example. Arbuckle (1997) presents (Example 17) a small set of cases (N=73) and variables from the Holzinger and Swineford (1939) study (see Chapter 3), in which some 27 % of the information has been made missing artificially. The six variables in the dataset (*grnt_x.sav* in the Amos *Examples* directory) have been imported (after renaming some of the variables, and after recoding *sysmis* to -1) into a project called *GRNT* which is available in the directory *STREAMS\EXAMPLES\GRNT* (or in the *GRNT.ZIP* file in the STREAMS installation directory).

In the first step the satuarated model has been fitted. The MB instructions for estimating this model with Amos 3.6 are:

```
TI Saturated model for Amos Example 17
MO PR=grnt NAME=satur
MO Create instructions for: Amos     Matrix: CM
MO One-group model     Means included in model
MO Amos 3.6
OP $missing=-1
DAT FOLDER=Raw DATLAB=Tot
MVR VISPERC CUBES LOZENG PARAGR SENTEN WORDME
COV VISPERC CUBES LOZENG PARAGR SENTEN WORDME
VAR VISPERC CUBES LOZENG PARAGR SENTEN WORDME
```

The model specification consists of two statements: one COV statement, and one VAR statement. It must thus be observed that the COV statement does not imply that variances will also be estimated. Observe also that the means are included in the model, which always must be done when estimating missing-data models. It should also be emphasized that the model type is an ordinary one-group model when using the rawdata-based estimation procedure.

Estimation of this model required 14 iterations, and the only output produced by the post-

processor is the following:

```
Goodness of Fit Test:

Minimum for H0-model = 1363.59, number of parameters 27
```

The post-processor only retrieves the minimum of the fitting function and the number of estimated parameters from the Amos output. Thus, no parameter estimates are presented, because the post-processor only presents estimates of relations. However, the Amos listing file gives estimates of all parameters.

If the estimated covariance matrix is requested (see the **Output** category of the *Amos Options* form) it is written, along with other information, to a file with *.amp* as suffix and the model name as prefix. From this file the estimated covariance matrix may be retrieved and imported into the project, and could thus be used for further modeling. (Observe, however, that the post-processor must be turned off when this is done, because the post-processor deletes the .amp file after it has been used.) Analysis of this matrix will not, however, yield the correct standard errors for parameter estimates and the goodness-of-fit test will be incorrect. However, approximate modification indices will be computed, which at times may be quite useful. Normally, however, the continued modeling would be done from rawdata, as is described below.

Mx may also be used to estimate the saturated model, using the same specification as that shown for Amos above. When this is done the following output is obtained:

```
Goodness of Fit Test:

Minimum of function = 1893.97, number of parameters = 27
```

Thus, with Mx a different minimum of the estimation function is obtained, which is because the Raw Maximum Likelihood function in Mx is somewhat differently defined than is the estimation function used in Amos. As will be demonstrated below the programs do nevertheless yield the same results.

Estimating the Restricted Models

The models which impose restrictions are specified as ordinary MB-models, using the same technique as when estimating a saturated model. If, however, a saturated model has been estimated for a set of data, it is possible (but not necessary) to refer to this model as a comparison model, which makes it possible for the post-processor to compute the χ^2 goodness-of-fit test. To identify a saturated model to be used as a comparison model, the **Comparison Model** tab on the *Model* form is used to identify the model. Make sure, however, that the comparison model has been fitted to exactly the same variables and group of cases that the restricted model is fitted to, because otherwise the test will be incorrect. The comparison model must also have been estimated with the same program, because the estimation functions are somewhat differently defined.

The MB specification for fitting a two-factor model with one *Verbal* and one *Spatial* fac-

tor to the six variables is:

```
TI Two oblique factors
MO PR=grnt NAME=o2a
MO Comparison model: Satur
MO Create instructions for: Amos     Matrix: CM
MO One-group model     Means included in model
MO Amos 3.6
OP $missing=-1
DAT FOLDER=Raw DATLAB=Tot
MVR VISPERC CUBES LOZENG PARAGR SENTEN WORDME
LVR Verb Spat
REL Verb -> PARAGR SENTEN WORDME
REL Spat -> VISPERC CUBES LOZENG
COV Verb Spat
```

This is the same model as Arbuckle's (1997) Model B for Example 17. Estimation of this model required 14 iterations, and now the post-processor presents the following goodness-of-fit information, along with parameter estimates identical to those obtained by Arbuckle:

```
Test-statistic for the model = 1375.13, df = 19.

Test-statistic for comparison model = 1363.59, df = 27.
Chi-square difference test = 11.55, df = 8.
```

The χ^2 test statistic is 11.55, and with 8 df this is nonsignificant. There is, thus, no reason to reject the two-factor model.

The restricted model may also be estimated with Mx, using the same model specification as for Amos. When supplied with start values, Mx produces the same parameter estimates as those obtained with Amos. The following goodness-of-fit test is computed by Mx:

```
Minimum of function = 1905.68, number of parameters = 19

Test-statistic for comparison model = 1893.97, df = 27.
Chi-square difference test = 11.71, df = 8.
```

Thus, Mx gives a χ^2 test statistic of 11.71, which is very close to the value obtained with Amos (i. e., 11.55). The fact that quite different minima of the estimation function are obtained with the two programs is thus not important. It is obvious, however, that a model estimated with one of the programs may not be used as a comparison model for the other program.

Here the saturated model was selected as comparison model. It is not necessary, however, to use the saturated model as a comparison model. Any less restricted model may be selected, in which case the difference χ^2 test is computed.

The same model may also be estimated with Mplus, and when this program is used the χ^2 test statistic may be obtained in a single run. To do that the options **Incomplete data model** and **H1** on the **Model Type** tab of the *Model* form must both be checked. STREAMS automatically takes care of the specification of missing data codes in the Mplus setup, granted that missing data codes have been defined in the data file. Thus, the

following instructions should be used;

```
TI Two oblique factors
MO PR=grnt NAME=o2
MO Create instructions for: Mplus     Matrix: CM
MO Means included in model     One-population model
MO Model Type: Incomplete Data H1
MO Mplus 1.0
OP ANAL ESTIMATOR=ML;
OP ANAL TYPE=MEANSTRUCTURE MISSING H1;
POP Raw
DAT FOLDER=Raw DATLAB=Tot POP=Raw
MVR VISPERC CUBES LOZENG PARAGR SENTEN WORDME
LVR Verb Spat
REL Verb -> PARAGR SENTEN WORDME
REL Spat -> VISPERC CUBES LOZENG
COV Verb Spat
```

When this model is estimated with Mplus the following goodness-of-fit statistics are obtained:

```
Goodness of Fit Test:

Chi-square = 11.71, df = 8, p < .16

RMSEA = .080, p-value for RMSEA < 0.05 = .27
```

The test statistic thus is identical with that obtained with Mx, and the parameter estimates also are identical with those obtained with Amos and Mx.

The Matrix-Based Estimation Procedure

There is also an alternative, matrix-based, procedure for obtaining maximum-likelihood estimates of model parameters from missing data. This procedure, which has been described by Allison (1987) and Muthén, Kaplan & Hollis (1987), and which is also the procedure implemented in Mplus, assumes that one covariance matrix and one mean vector is computed for each pattern of missing data. When there are many such patterns this approach is, of course, less useful, but when there is a limited number of patterns it may be considerably more efficient than the rawdata-based procedure. Another advantage of the matrix-based procedure is that it is not restricted to the Amos, Mplus and Mx programs. STREAMS supports it for Amos, LISREL and Mx. EQS is not supported, because EQS does not write information to an output file when multiple-group models are fitted. However, the EQS manual (Bentler, 1995, pp. 197-200) describes how such models may be specified in EQS as well.

The matrix-based procedure thus works best when there is a limited number of missing-data patterns, which is more often the case when there are structurally missing data than when there is accidentally missing data. Data to be modeled often consists of several identifiable subgroups of cases for which partially different sets of variables have been observed. Observations may, thus, be missing by design for a large number of reasons, such as because:

- A matrix sampling design was used so that different subsets of cases were given different combinations of items.
- The subjects were divided into groups according to level of performance, and were given different forms of a test, matched to their level of performance.
- A high-performing subset of cases was selected from a larger population, and was followed up with respect to achievement in an education.

The basic principle of the matrix-based technique for modeling incomplete data is that the different categories of cases with different combinations of valid variables are regarded as being samples, although not necessarily representative, from the same population (Allison, 1987; Muthén, Kaplan, & Hollis, 1987). The model is specified in such a way that equality constraints over groups are imposed on each and every parameter, including means of observed variables, which must be included in the model. The LISREL program expects all input matrices to have the same number of variables, so when this program is used some special preparations are made. In the covariance matrix, missing variables are thus replaced with dummy variables which have zero covariance with every other variable and unit variance. In the model these dummy variables are not subject to any constraints.

The model estimated in this way gives, under certain assumptions, the correct estimates, but the χ^2 test is incorrect, as is the reported df. This is because the χ^2 test is not only sensitive to deviations between model and data, but also to differences in the covariance matrices and mean vectors between the different groups that form each population. The dummy variables in the covariance matrices also cause LISREL to overestimate the degrees of freedom for the model (see Jöreskog & Sörbom, 1989a). As is demonstrated by Muthén et al. (1987) it is, however, possible to correct for both those problems through estimating another model (i. e., the H1 model) which is the saturated model, and which essentially tests the homogeneity of the covariance matrices and mean vectors for groups belonging to the same population. Because the deviations between model and data (the H0 model) and the differences between subgroups (the H1 model) are additive, the correct test statistic is obtained through taking the difference between the two test statistics. Thus, the same procedure as used in the rawdata approach must be employed.

Below the different steps are described in more concrete terms.

Preparing Data for Modeling

Chapter 9 describes how STREAMS may be used to compute covariance matrices, and how these procedures may be used when preparing matrices for incomplete data modeling. The data should be prepared in such a way that all variables for all cases are collected in one large data file, with as many variables as the union of variables over groups. This data file may be used as an input file to STREAMS, which sorts the cases into different groups according to the pattern of missing values, and computes a covariance matrix and a mean vector for each group which is larger than a specified minimum number of cases. Because this procedure may generate a very large number of groups when there are many variables and accidental missingness it may be wortwhile to consider replacement of some of the missing values with mean values.

Missing variables are not kept in the matrix in the project dictionary, but when the matrix is retrieved to be included in a model for incomplete data it is automatically expanded with the missing variables, through addition of dummy variables which have unit variance and zero covariance with all other variables.

The MB Language for Modeling Incomplete Data with Multiple Matrices

All the basic MB statements are identically the same when modeling incomplete data as when ordinary data is modeled, but the DAT statement has a slightly different appearance. When ordinary multiple-group models are fitted each group of cases (i. e., each DAT line) corresponds to a population. However, when incomplete data is modeled with the matrix-based procedure each population is represented by more than one group of cases, and STREAMS must be informed about which groups of cases belongs to which population. First the population labels must be declared, which is done with the POP statement. This statement has the following syntax:

```
POP labels
```

The population labels may be freely chosen, and may contain 1-8 alphanumeric characters. An example could be:

```
POP Males Females
```

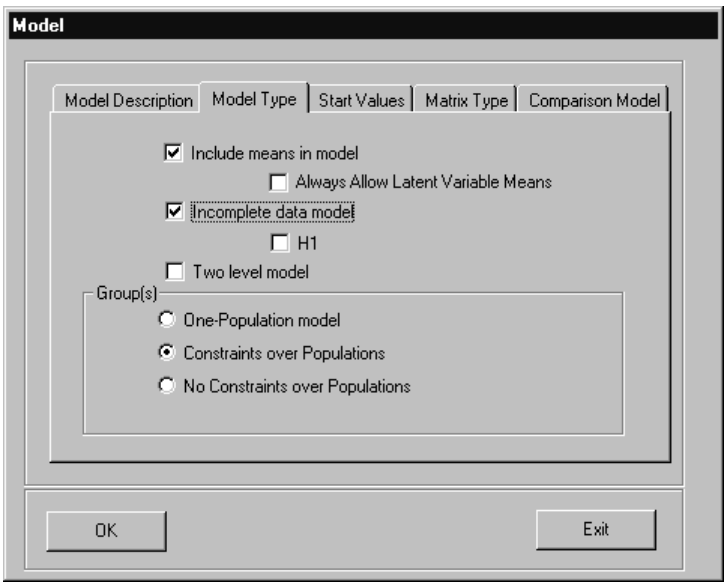
The population labels are used in the DAT statements to assign each dataset to a particular population. Suppose, for example, that the POP statement above was used to declare the Males and Females populations, and that there are two datasets within each population. The DAT statements could then be:

```
DAT FOLDER=Males DATLAB=BOOK1 POP=Males
DAT FOLDER=Males DATLAB=BOOK2 POP=Males
DAT FOLDER=Females DATLAB=BOOK1 POP=Females
DAT FOLDER=Females DATLAB=BOOK2 POP=Females
```

Normally these statements are generated by STREAMS, in ways described below.

Creating a Model for Incomplete Data with the Matrix-Based Procedure

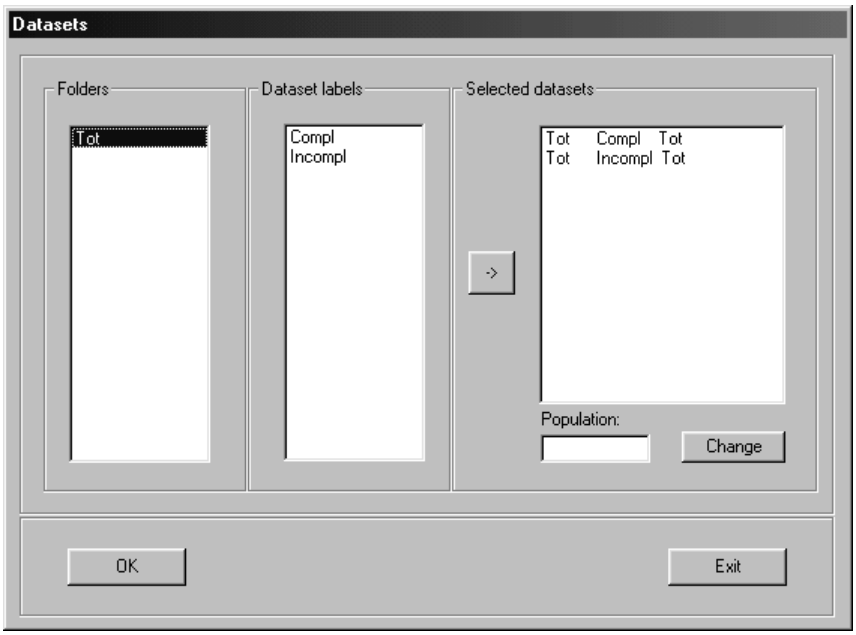
In order to obtain a model for incomplete data the check-box labeled **Incomplete Data Model** on the **Model Type** tab of the *Model* form must be clicked. To specify an H1 model the **Incomplete Data Model** and **H1 Model** check-boxes must be selected.



When the **Incomplete Data Model** option has been selected the *Datasets* form takes on a somewhat different appearance.

The Datasets Form

The *Datasets* form, which is produced when the **DAT** button is clicked, now also displays population label:



When incomplete data is modeled two or more groups represent the same population. It thus is necessary to inform STREAMS that different groups belong to the same population. This is simply done through assigning these groups to the same population. STREAMS assumes by default that the population labels are the same as the label of the folder. This may, however, easily be changed by first selecting one or more datasets in the **Selected datasets** list, then writing a new population label in the field labeled **Population** and finally clicking the **Change** button.

Creating and Estimating the Model

In most respects the specification and estimation of models for incomplete data is done in the same way as for complete data. As has been described above, it is, however, necessary to specify the population to which each of the groups of cases belongs. The population labels are then used in the MB statements in the same way as the group labels are used in modeling of complete data.

In order to get the correct value of the test statistic the H1 model (see below) for any particular combination of groups and variables must first be estimated. This model is then used as the comparison model, in the same way as in the rawdata-based procedure.

We will illustrate the different steps and procedures involved in estimating a model for incomplete data through considering a classical example presented by Allison (1987) and

which is also discussed at some length by Jöreskog & Sörbom (1989a, pp. 258-261).

The example relies on data collected by Bielby et al. (1977), who used a sample of 2020 black men to estimate the correlation between father's occupational status (FAOC) and father's educational attainment (FAED). However, in order to correct the estimate of the correlation for attenuation due to errors of measurement they took a subsample of 348 subjects from the original sample and reinterviewed them some three weeks after the original data collection. In this way the subsample obtained two measures of father's occupational status (FAOC and FAOC2) and two measures of father's educational attainment (FAED and FAED2). One subsample (N=348) thus has complete data, and one subsample (N=1672) has incomplete data.

The covariance matrices and mean vectors for the two samples have been imported into a project called *Bielby* in the directory *STREAMS\EXAMPLES\BIELBY.* The two datasets have group labels *Compl* and *Incompl*, which are both in the project folder *Tot*. The imported matrix for the *Compl* group has four variables, while the imported matrix for the *Incompl* group has two variables (the missing variables should thus not be included in the matrix).

In order to estimate the disattenuated correlation between father's occupation and education we may fit the following model (*CCompl*) for the subset of cases with complete data:

MB instructions for the model with two latent variables for cases with complete data

```
DAT FOLDER=Tot DATLAB=Compl
MVR FAOC FAOC2 FAED FAED2
LVR FaOcc FaEd
REL FaOcc -> FAOC FAOC2
REL FaEd -> FAED FAED2
COV FaOcc FaEd
```

This model fits excellently (χ^2 (1) = 1.96, p < .16) and gives the following standardized estimates according to all the estimation programs:

Standardized estimates from the model with two latent variables for cases with complete data

```
Standardized estimates:

FAOC     =    +0.73*FaOcc      +0.68*FAOC&
FAOC2    =    +0.87*FaOcc      +0.49*FAOC2&
FAED     =    +0.94*FaEd       +0.33*FAED&
FAED2    =    +0.97*FaEd       +0.24*FAED2&

Cov(FaEd,FaOcc)       =    0.62
```

The observed correlation between FAOC and FAED is .43, so the estimated disattenuated correlation of .62 is considerably higher.

To use all data to estimate the disattenuated correlation we have to select an **Incomplete Data Model** in the **Model Type** category of the *Model* form. On the *Datasets* form we then select both the *Compl* and the *Incompl* datasets, which by default will be assigned to

the population *Tot*. The MB specification for this model (*CTot*) thus is:

MB instructions for the incomplete data model with two latent variables

```
POP Tot
DAT FOLDER=Tot DATLAB=Compl POP=Tot
DAT FOLDER=Tot DATLAB=Incompl POP=Tot
MVR FAOC FAOC2 FAED FAED2
LVR FaOcc FaEduc
REL FaOcc -> FAOC FAOC2
REL FaEduc -> FAED FAED2
COV FaOcc FaEduc
```

When estimated with Amos the test-statistic is χ^2 (6) = 7.70. When estimated with LISREL and Mx, the goodness-of-fit test for this model is χ^2 (15) = 7.69. These statistics are, however, incorrect. One reason for this is that the degrees of freedom are overestimated by LISREL and Mx because the missing elements in the matrix for the group with missing variables are included in the count of elements. Another reason is that the model is specified as a two-group model by all programs, which causes the degrees of freedom to be overestimated, and the χ^2 statistic to be influenced by differences between the observed data for groups. Both these problems are solved, however, if the so called H1-model is estimated (Muthén et al., 1987). This is, essentially, a model which constrains all corresponding and actually existing elements in the covariance matrices and mean vectors to be equal over groups. The χ^2 statistic of this model thus is sensitive to the amount of differences between groups, and the degrees of freedom reflect the actual number of elements in the matrices. To obtain a correct test statistic for the model, the test statistic for the H1 model should be subtracted from the test statistic computed by the estimation program for the model. This is done by STREAMS if an H1 model has been estimated for a particular combination of groups and variables, and if this model is selected as a comparison model.

To estimate the H1 model we select the options **H1 Model** and **Incomplete Data Model** on the **Model Type** tab of the *Model* form. This specification causes STREAMS to disregard all MB statements, and to set up an H1 model instead. When such a model is estimated for the present data (*CTotH1*) the only message from the post-processor when LISREL and Mx are used is:

```
Chi-square for H1-model = 6.11, df = 14
```

When Amos 3.6 is used the following result is obtained:

```
Test-statistic for the model = 6.12, df = 5.
```

If the *CTot* model is reestimated after the *CTotH1* model has been estimated the following results are obtained from the goodness-of-fit test from LISREL and Mx:

```
Goodness of Fit Test:

Chi-square = 7.69, df = 15, p < .94

Test-statistic for comparison model = 6.11, df = 14.
Chi-square difference test = 1.58, df = 1.
```

When Amos 3.6 is used the following result is obtained:

```
Test-statistic for the model = 7.70, df = 6.

Test-statistic for comparison model = 6.12, df = 5.
Chi-square difference test = 1.58, df = 1.
```

Thus, after the correction the test statistic for the incomplete data model is the same for all the estimation programs. It may also be observed that the result is close to the test statistic for the model for the complete data.

All programs achieve the same estimates. The standardized estimates are presented below:

Standardized estimates from the incomplete data model

```
Standardized estimates:

FAOC     =    +0.74*FaOcc      +0.67*FAOC&
FAOC2    =    +0.89*FaOcc      +0.45*FAOC2&
FAED     =    +0.94*FaEd       +0.34*FAED&
FAED2    =    +0.97*FaEd       +0.23*FAED2&


Cov(FaEd,FaOcc)       =    0.62
```

These estimates are similar to those obtained in the model for complete data only. However, even though both the standardized and the unstandardized estimates are very similar, the standard errors and the t-values are not. Thus, in the incomplete data model the t-value for the covariance between the two latent variables is 17.82, while in the model for complete data the t-value is only 7.44. The introduction of the entire sample into the model thus causes the precision of estimates to improve.

The reason why the estimates are the same whether an incomplete data model is estimated or not is that in this case the subsample with complete data is a random sample of the complete sample. Here the data thus are MCAR. When the MCAR assumption is fulfilled the modeling of incomplete data may be seen as an optimal combination of data from different subgroups of cases, and no further assumption about the missing data mechanism is needed. But even when there are differences between the groups of cases it is often possible to arrive at estimates of a high quality.

To summarize, the following steps should thus be taken when a model for incomplete data is estimated with the matrix-based procedure:

1. Estimate an H1 model for a particular selection of variables and groups of cases.
2. Estimate the model for the same observed variables and groups of cases and subtract the test-statistic for the H1 model from the restricted model, in order to obtain the correct test statistic.

6 Specifying Models for Two-Level Data

Phenomena studied in social and behavioral research often have a hierarchical structure, where individuals define one level of observation and groups or social organisations define one or more higher levels of observation. In educational research, for example, there is an interest in determining effects of characteristics of the school, the teacher, and the teaching on the development of individual students. However, classrooms are nested within schools, and students are nested within classrooms, so the observational structure is unavoidably hierarchical.

Hierarchical data structures are exceedingly difficult to analyze properly (Bock, 1989), and as yet there does not exist a fully developed methodology for how to analyze such data with structural equation modeling techniques (Hox, 1994). However, Muthén (1989, 1990, 1991, 1994) has shown how approximate maximum likelihood estimates of parameters in a two-level model may be obtained with standard software for structural equation modeling, such as Amos, LISREL and LISCOMP. The resulting model specification is quite complex, however, so there have only been few applications of this approach so far (see, however, Gustafsson, 1997, 1998; Härnqvist, Gustafsson, Muthén, & Nelson, 1994; Muthén, 1990, 1991, 1994).

The MB language is, however, easily extended to allow two-level modeling, so with STREAMS two-level models are only marginally more difficult to specify and estimate than are ordinary one-level models. The recently presented Mplus program (Muthén & Muthén, 1998) also supports two-level structural equation modeling in an implementation of the same estimation principles as those used in STREAMS. However, STREAMS also supports the Mplus two-level model specification, so it is possible to take advantage of the general advantages of STREAMS (e. g., starting values and a common modeling environment) here too. The present chapter provides a self-contained description of the steps and procedures involved in preparing data for analysis, and in specifying and estimating two-level models.

Basic Principles and Concepts of Two-Level Structural Equation Modeling

Structural models for multilevel data have been proposed by, among others, McDonald (1993), Goldstein & McDonald (1988), McDonald & Goldstein (1989), Muthén (1989, 1990), and Muthén & Satorra (1989), and this literature makes it quite clear that a general multilevel structural equation model is too complicated to be practically feasible for the time being. It is, however, possible to formulate less than perfectly general models which are quite interesting. One such approach allows formulation of models for differences in means and intercepts between groups, but not for differences in regression coefficients. A simple version of this model was formulated by Cronbach (1976; see also Härnqvist, 1978), but the model was extended and put within a framework of maximum likelihood estimation by Muthén (1989, 1990).

The Two-Level Model

In its two-level form the model assumes that there is a set of N individuals (e. g., students) who belong to G groups (e. g., classes). The individuals have scores on P variables (Y_1, Y_2, ..., Y_p) (e. g., ability and achievement variables) which for each individual are assembled into the vector $\mathbf{Y_i}$. We may also (but need not) have observed variables at the group level (Z_1, Z_2, ..., Z_q) (e. g., class size and teacher characteristics). From these data two matrices of relations among a set of P observed variables may be computed. One is the pooled-within covariance matrix ($\mathbf{S_{pw}}$):

$$S_{PW} = (N - G)^{-1} \sum_{g=1}^{G} \sum_{i=1}^{N_g} (Y_{gi} - \bar{Y}_g)(Y_{gi} - \bar{Y}_g)'$$

This matrix thus is computed as an ordinary covariance matrix except that deviations of the individual scores are computed from group means rather than from the grand means. For this matrix the actual number of observations is N-G. The other matrix is the between groups covariance matrix ($\mathbf{S_B}$):

$$S_B = (G - 1)^{-1} \sum_{g=1}^{G} (N_g)(\bar{Y}_g - \bar{Y})(\bar{Y}_g - \bar{Y})'$$

The between groups matrix is basically computed from the group means, and their deviations around the grand means. This matrix thus is based on G observations.

As is shown by Muthén (1989, 1990) the $\mathbf{S_{PW}}$ estimates the population matrix $\Sigma_{\mathbf{PW}}$. However, it is not possible to model $\mathbf{S_B}$ in order to understand the structure of between-group differences. This is because the observed $\mathbf{S_B}$ matrix is actually a function of both the population between matrix ($\Sigma_{\mathbf{B}}$) and the population pooled within matrix ($\Sigma_{\mathbf{PW}}$). Thus, the

expectation of $\Sigma_{\mathbf{B}}$ is $\Sigma_{\mathbf{PW}} + c\Sigma_{\mathbf{B}}$where

$$c = \frac{N^2 - \sum_{g=1}^{G} N_g^2}{N(G-1)}$$

The constant c thus is a function of the group sizes. When all group sizes are equal c is equal to the common group size, and when group sizes are unequal c tends to be close to the mean group size.

Thus, a proper model for the $\mathbf{S_B}$ matrix must capture both the within-group and the between-group structure, but because $\mathbf{S_{PW}}$ is an unbiased estimate of $\Sigma_{\mathbf{PW}}$ it is possible to devise reasonably simple estimators of both the within- and between-group structures. As was shown by Muthén (1989, 1990) a solution may be formulated in terms of a two-group model, where the $\mathbf{S_B}$ matrix is treated as one group, and the $\mathbf{S_{PW}}$ matrix is treated as another group.

When all group sizes are equal Muthén (1989, 1990) shows that the estimates obtained are maximum likelihood estimates, but when group sizes are unequal this estimator (labeled Muthén's Approximate Maximum Likelihood estimator, MUML) yields only approximate maximum likelihood estimates (even though they are consistent), so standard errors and tests of model fit are not quite correct. It has been shown, however, that the amount of error is quite small in normal situations (see, e. g., Muthén, 1990, 1994). As is shown by Muthén (1990) it is also possible to construct a full information maximum likelihood estimator, but this requires a model with as many groups as there are group sizes, which makes this estimator quite unpractical.

To correctly understand the meaning of two-level structural equation modeling it must be realized that the two-group model specification is just a convenient procedure to obtain the estimates with standard structural equation modeling software. Conceptually, however, the model refers to the total covariance matrix, and the model should be conceived of as a model for one population, with observations at two levels of aggregation (i. e., the individual level and the group level). This may be more clear from a path diagram for a simple two-level confirmatory factor analysis. In the model shown on the next page there are four observed variables (Y1, ..., Y4), and it is assumed that there is one general factor at the individual level (*GenW*) and one general factor at the group level (*GenB*). There also are three group-level observed variables (Z1, Z2, and Z3), but we first discuss the Y-variables.

The individual level variation is captured by *GenW* and for each observed variabble there is also a residual at the individual level which is, as usual, identified with an ampersand as a suffix to the variable name (e. g., Y1&). The group level variation in observed scores is represented by the latent variables which have a 2 as a prefix (i. e., 2Y1, ..., 2Y4). These variables may be thought of as representing group means on observed variables, and in the model they are related to the observed variables (i. e., Y1, ..., Y4) with paths assigned the fixed value $\sqrt{c}$. The group level variation is modeled in terms of the latent variable *GenB*, which accounts for variance in the 2Y1, ..., 2Y4 variables. However, for each group level variable there is also a residual variable (2Y1&, ..., 2Y4&) which represents the group level variability which remains after the *GenB* factor has been taken into account.

At first sight the two-level path diagram may appear somewhat complicated, and it may,

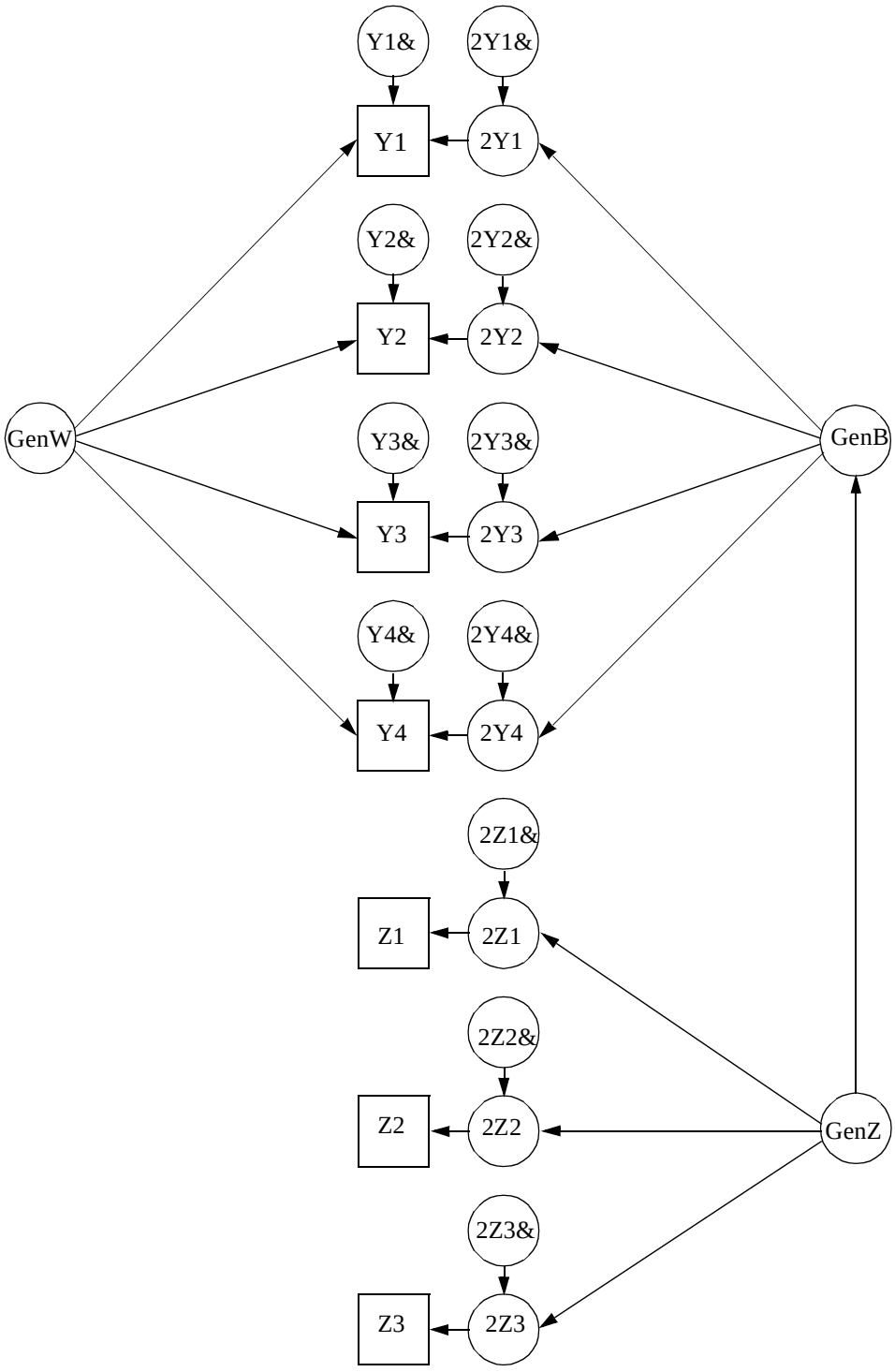


Figure 4. *A path diagram for a hypothetical two-level model*

in particular, be difficult to reconcile with the pooled-within and between-group matrices. If, however, it is realized that the diagram represents the latent sources of variance in the total observed variation (i. e., for the sum of the pooled-within and between matrices) for one population, the path diagram is easier to understand. Thus, for the present model the path diagram basically says that the total variability in the observed variables may be decomposed into four orthogonal sources of variance: individual variability common to

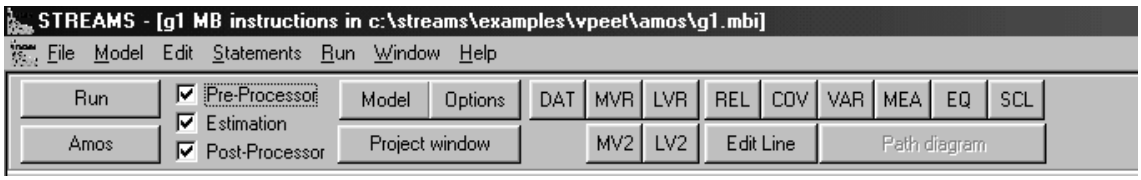
all observed variables (i. e., the *GenW* contribution), individual variability specific to each observed variable (i. e., the Y1&, ..., Y4& contributions, which represent a mixture of random error and specificity), group variability common to all observed variables (i. e., the *GenB* contribution), and, finally, group variability specific to each observed variable (i. e., the 2Y1&, ..., 2Y4& contributions which represent a mixture of random error and group level specificity). With larger sets of observed variables it may, of course, be possible to identify several latent variables at both individual and group levels.

So far we have only discussed the P individual variables, and their corresponding group means. But there also may be manifest variables which are only observed at the group level (e. g., teacher experience, measured in terms of number of active years, and class size) and which may be invoked as potential explanatory variables. Such group level variables have been referred to as Z-variables by Muthén (1990) and will here be referred to as "group-level manifest variables."

In this model there are three observed group-level variables (Z1, Z2, and Z3), which are indicators of a latent group-level variable (GenZ). The *GenZ* variable is hypothesized to be an independent variable in relation to *GenB*. In the same way as for the Y-variables, the Z-variables are connected to latent variables with 2 as a prefix (e. g., 2Z1), with the value of the path coefficient fixed at $\sqrt{c}$. Only these scaled versions of the group-level manifest variables may be involved in models, either as independent or dependent variables. The group-level manifest variables may thus be used in a way which corresponds to using manifest variables in regression analysis, or, as is the case here, as indicators of latent variables.

The MB Language for Two-Level Modeling

In order to allow formulation of two-level models the MB language requires some simple extensions, which follow directly from the path diagram presented above. The ordinary statements refer to the individual level (level 1), but some new statements (MV2, LV2) have been introduced to refer to the group level (level 2) (see Chapter 12). Thus, when a two-level model has been specified as the model type (which is done on the **Model Type** tab on the *Model* form), the Model Building toolbar includes two additional buttons, **MV2** and **LV2**:



The MV2 statement is used to identify the group-level manifest variables to be included in the model. This statement should not be used to declare variables which have been observed at the individual level, because these variables are always automatically available at the group level in the form of group means. Thus, the MV2 statement should only declare variables measured at the group-level but not at the individual level.

In the MB language the group-level variables (both aggregated variables and variables measured at group-level) only are referred to with a "2" as a prefix to the variable name. For example, if an individual variable is called TEST1, the corresponding group level var-

iable is called 2TEST1. It is also important to observe that when a manifest variable has been declared as a group-level manifest variable this variable must also have the “2” as a prefix when it is referred to in MB statements. This is because the MV2 statement refers to the manifest variables in the project dictionary where no distinction is being made between variables at different levels of observation, while the MB statements in a two-level model refer to structures at two levels.

The fact that the character “2” is used as a prefix in variable labels to separate group level variables from individual level variables implies that at most 6 characters may be used in variable labels when two-level modeling will be done.

The LV2 Statemen is used to declare the latent variables at the group level in the two-level model. The names of latent variables may be freely chosen, but it is recommended that they should contain at least one lower-case letter, and at least one letter which indicates that this is a group-level latent variable. It may thus be good practice to use the letter W as a suffix in labels for the individual latent variables, and the letter B as a suffix in labels for the latent variables at group level.

Preparing Data for Two-Level Modeling

To use the MUML estimator for two-level modeling it is first of all necessary to compute the pooled-within and between-group covariance matrices, and also to compute the c constant, which plays an important role in the model specification. These preparations may be done with STREAMS (see Chapter 9), which includes a rewritten version of the public domain program BW constructed by Muthén (see Nelson & Muthén, 1991; the BW program is also available with Hox, 1994). The code has been rewritten to read the data twice, which implies that the program accepts non-sorted data and does not require information about group sizes as input. The revision also has reduced the problems of numerical instability which afflict BW in some circumstances. The practical procedure for preparing matrices for two-level analysis is described in Chapter 9.

When Mplus is used for two-level modeling only rawdata is accepted as input. Thus, in a first step the individual rawdata must be imported into the STREAMS project (see Chapter 10). However, the current version of STREAMS does not support multiple-group two-level modeling for Mplus, so if data is imported for subsets of cases, these may only be analyzed in separate one-group models.

Specifying and Estimating Two-Level Models

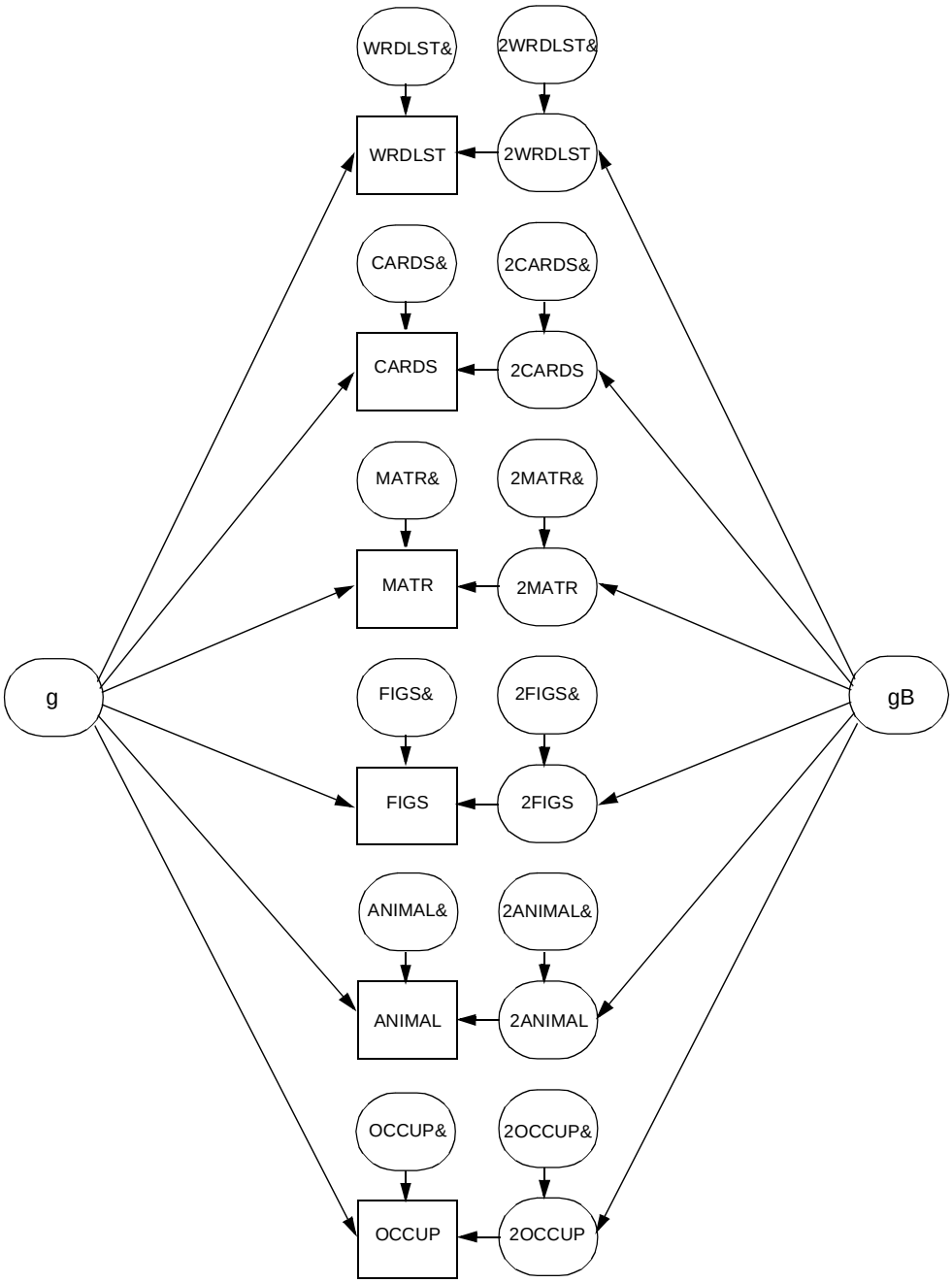
In most respects STREAMS is used in the same way when two-level models are specified and estimated as when ordinary one-level modeling is being conducted. Below the procedures are illustrated with a concerete example.

Hox (1994) presents an example of a two-level confirmatory factor analysis of data originally collected by van Peet. The data are scores on six measures of cognitive ability for 187 children from 37 families. The six measures are: Word List (WRDLST), Cards (CARDS), Matrices (MATR), Figures (FIGUR), Animals (ANIMAL) and Occupations (OCCUP). The exact characteristics of these tests are not clear from the information sup-

plied, but it seems that there are verbal, spatial, reasoning and perceptual speed tests represented in the list. The variable FAMILY represents family belongingness.

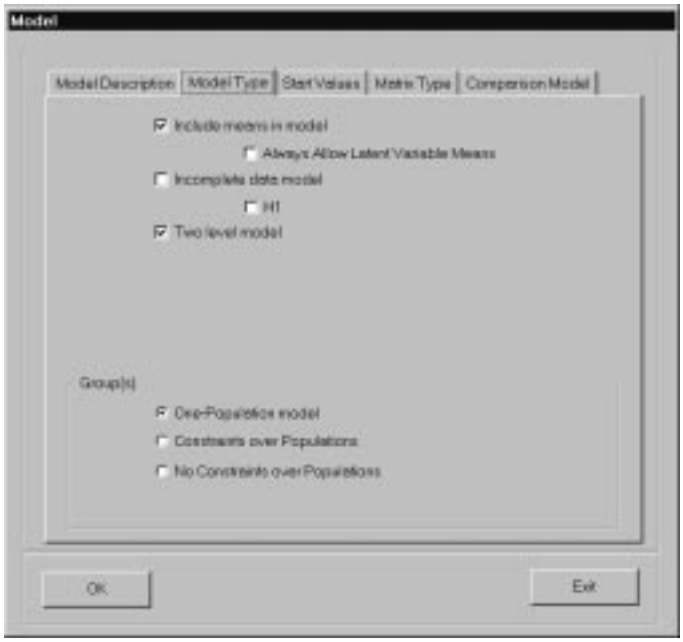
Chapter 9 describes how the matrices needed for two-level analysis of these data may be prepared.

Below a model will be specified which includes one general factor both at individual and at family level, as shown in the path diagram below:



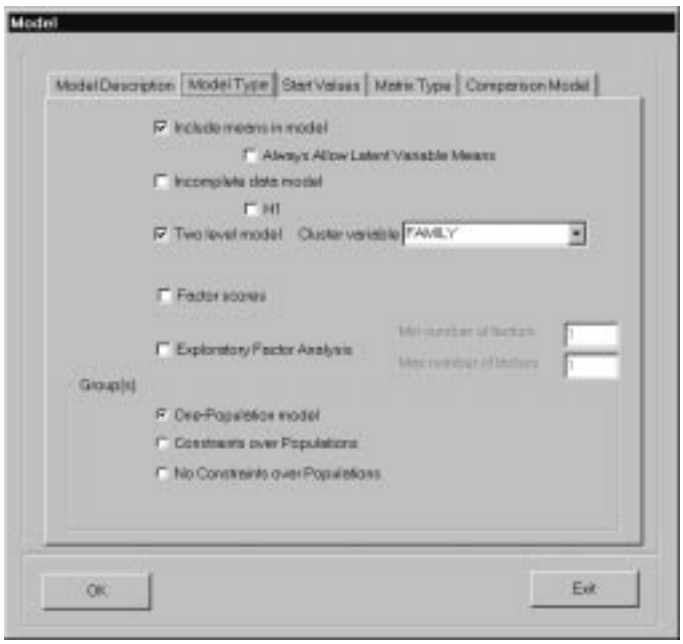
Specifying the Two-Level Model

In order to obtain a two-level model the check-box labeled **Two Level Model** on the **Model Type** tab of the *Model* form should be checked.



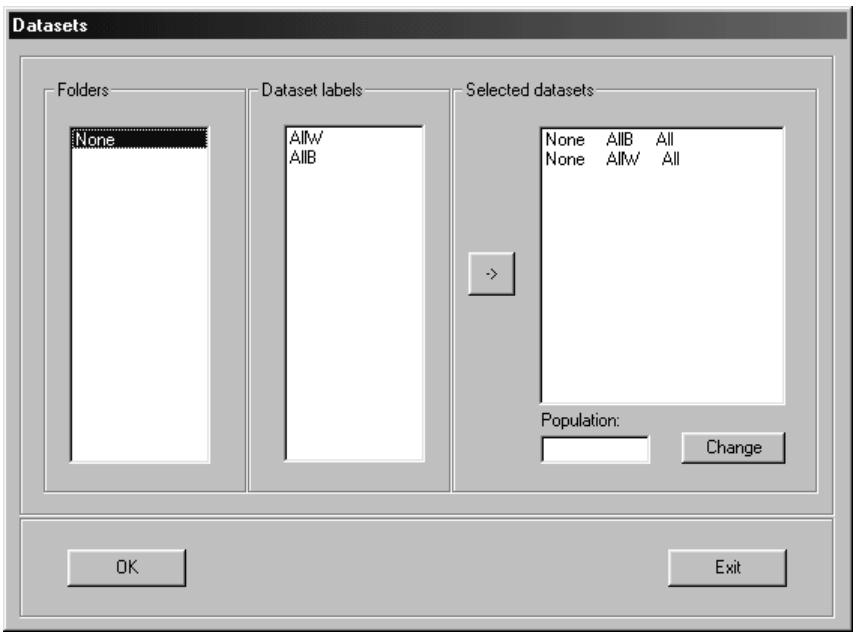
It may be observed that the references to *groups* are replaced with references to *populations*. This is because a two-level model always involves two matrices for each population, which makes the term *group* ambigous.

When Mplus has been selected as estimation program the **Model Type** tab has a somewhat different appearance:



Next to the **Two level model** check box is a drop down menu which is labeled **Cluster variable**. The menu is used to select the variable in the raw data set (which must be identified on the *Datasets* form which is produced by the **DAT** button, see below) which identifies the group membership variable (here FAMILY). This is because with Mplus only raw data input is allowed.

After possible options for the estimation program have been specified with **Options** button, the **DAT** button should be clicked. This reveals that the *Datasets* form is somewhat different as compared with one-level modeling:



Two-level modeling involves, just as modeling of incomplete data, specification of a multiple-group model, but where in fact two or more groups represent the same population. It thus is necessary to inform STREAMS that both the between-group matrix and the pooled-within matrix are different aspects of a sample from one population. This is done through assigning these matrices ("groups") to the same population. In two-level modeling STREAMS assumes by default that the common prefix part of the dataset label is the population name (e. g., All). Thus, both for one- and multiple-population model this specification is practically always handled automatically by STREAMS, and the user rarely needs to change anything. Population labels may, however, easily be changed by first selecting one or more groups in the **Selected datasets** list, then writing a new population label in the field labeled **Population**, and finally clicking the **Change** button.

It is necessary to select both the between-group matrix and the pooled-within group matrix for a two-level model, and in the MB specification the between-group matrix must always precede the within-group matrix within each population. When STREAMS generates the DAT statements this is (mostly) done the proper way. However, if the MB statements are edited, or generated through some other procedure, it is essential that the DAT statements are properly ordered.

Next the **MVR** button is clicked and the individual (i. e., level one) variables are selected as usual. In the present example this results in the statement:

```
MVR WRDLST CARDS MATR FIGS ANIMAL OCCUP
```

The individual level latent variable is defined on the *Latent Variables* form produced by the **LVR** button. Here a single individual latent variable (*g*) is hypothesized:

```
LVR g
```

The family level latent variable is defined on the *Latent Variables Level Two* form which may be produced by clicking the **LV2** button:

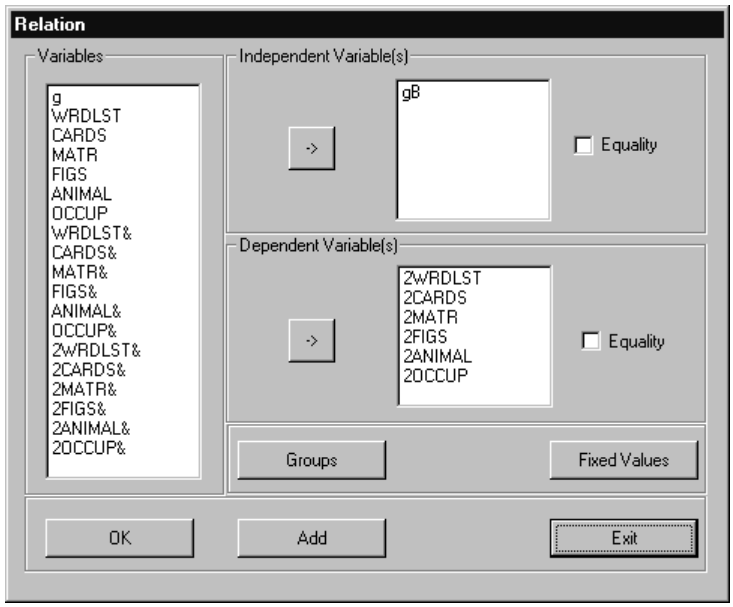
Here a single family level latent variable (*gB*) is hypothesized, and when the **OK** button is clicked the following statement is created:

```
LV2 gB
```

It should be observed that the present model does not include any group-level (i. e., Z) variables, so there is no MV2 statement.

After manifest and latent variables at both levels have been declared, the specification of the two-level model in the MB language may begin. The same MB statements are available for specifying a two-level model as for specifing a one-level model, but these statements operate upon a broader set of variables than when a one-level model is specified. The lists of variables thus include both the individual level variables and the group level variables, and as has already been described the latter have a “2” as a prefix.

Thus, when the **REL** button is clicked the list-boxes on the *Relation* form have the following content:



In order to specify the model we need two REL statements, one for the individual level relation, and one for the family level relation. These may easily be specified with the REL form:

```
REL g -> WRDLST CARDS MATR FIGS ANIMAL OCCUP
REL gB -> 2WRDLST 2CARDS 2MATR 2FIGS 2ANIMAL 2OCCUP
```

Estimating and Interpreting the Two-Level Model

The complete set of MB statements for the two-level model with one general factor at both levels for the van Peet data thus is:

MB instructions for the two-level one-factor model for the van Peet data

```
TI One general factor, between and within
MO PR=vpeet NAME=g1
MO Create instructions for: LISREL Y-model     Matrix: CM
MO One-group model
MO Model Type: Two Level
MO Means not included in model     LISREL DOS/Extender 8.14
OP OU ME=ML AD=OFF MI
POP All
DAT FOLDER=None DATLAB=AllB POP=All
DAT FOLDER=None DATLAB=AllW POP=All
MVR WRDLST CARDS MATR FIGS ANIMAL OCCUP
LVR g
LV2 gB
REL g -> WRDLST CARDS MATR FIGS ANIMAL OCCUP
REL gB -> 2WRDLST 2CARDS 2MATR 2FIGS 2ANIMAL 2OCCUP
```

Amos, LISREL, Mplus or Mx may be used to estimate a two-level model and normally the three-step procedure is run in the ordinary fashion (observe, however, that multiple-population models cannot be specified for Mplus, and that models with means and/or group-level variables cannot be specified for Mx). It must be emphasized, however, that two-level models may be more difficult to estimate than one-level models, and this is par-

ticularly true for large, multiple-population models which include means. One reason for this is that LISREL is not able to produce optimal start values for this type of models, and Mplus also needs to be supplied with start values. When working with two-level models it is, therefore, more essential than ever that the strategies for obtaining convergence described in Chapter 7 are adopted. It is in particular recommended that equality constraints are imposed on loadings over all manifest variables for both level one and level two latent variables.

The output from the estimation program is quite difficult to interpret for two-level models, so the post-processor plays a more important role here. The post-processor output from two-level models also is partially different from that of one-level models.

Goodness-of-fit Statistics

The following goodness-of-fit results were obtained:

Goodness-of-fit statistics for the two-level one-factor model for the van Peet data

```
Goodness of Fit Test:

Chi-square = 57.57, df = 18, p < .00

RMSEA = .109, p-value for RMSEA < 0.05 = .00

Fit Indices: GFI = .91, NFI = .70, NNFI = .60, CFI = .76

Maximum Modification Index is   29.1 for:
COV OCCUP& ANIMAL&
```

The goodness-of-fit measures for this type of two-level model may be interpreted in the usual way, so it may be concluded that the hypothesized model with one general factor at both levels of observation does not fit the data. It should be observed, however, that even though the programs agree on the χ^2 test (Mplus had a somewhat higher value of 58.21, though) they differ widely in their estimates of RMSEA (Amos .109, LISREL .154, Mplus .100 and Mx .132). There may be reason to have more trust in the Mplus RMSEA estimate than in the other ones because this program has been designed for the two-level model.

When Mplus is used to estimate the two-level model there is also a choice of two additional, robust, estimators: the MLM estimator with robust standard errors and mean-adjusted chi-square and the MLMV estimator with robust standard errors and mean-and variance adjusted χ^2. These estimators produce the same estimates as does the ordinary ML estimator, but the standard errors and the χ^2 statistic are adjusted to compensate for deviations from multivariate normality. For the one-factor model the MLM estimator gives $\chi^2\,(18) = 62.43,\ p < .00$ and the MLMV estimator $\chi^2\,(11) = 38.15,\ p < .00$. It should thus be observed that use of the MLMV estimator reduces the degrees of freedom. The two robust estimators do not result in any different conclusion than the ML estimator, but it should be observed that the t-values tend to be lower for the MLM and MLMV estimators.

When means are included in the model, the degrees of freedom determined by LISREL are incorrect, which is also true when group-level manifest variables are included in the

model. In these cases the pre-processor determines the correct degrees of freedom and adds a DF-statement on the LISREL OU line which corrects the degrees of freedom. This cannot be done for Mx, however, so two-level models with means and/or group-level manifest variables are not supported for this program.

Unstandardized Estimates

The unstandardized estimates, which are identical for all estimation programs, are presented first for the individual level:

Individual level estimates for the two-level one-factor model for the van Peet data

```
Unstandardized estimates:

WRDLST   =    +1.00*g          +1.00*WRDLST&
CARDS    =    +2.33*g          +1.00*CARDS&
MATR     =    +2.19*g          +1.00*MATR&
FIGS     =    +1.00*g          +1.00*FIGS&
ANIMAL   =    +0.75*g          +1.00*ANIMAL&
OCCUP    =    +0.33*g          +1.00*OCCUP&

Var(g)       =    1.54
Var(WRDLST)  =    9.54   Var(WRDLST&) =    8.00   Expl var =   16.12%
Var(CARDS)   =   18.74   Var(CARDS&)  =   10.42   Expl var =   44.41%
Var(MATR)    =   14.05   Var(MATR&)   =    6.68   Expl var =   52.50%
Var(FIGS)    =   18.10   Var(FIGS&)   =   16.55   Expl var =    8.55%
Var(ANIMAL)  =   17.75   Var(ANIMAL&) =   16.89   Expl var =    4.87%
Var(OCCUP)   =   15.20   Var(OCCUP&)  =   15.03   Expl var =    1.11%
```

The estimated relations at the individual level are presented first, and then the variance accounted for in the manifest variables is analyzed. This analysis is based on the estimated pooled within-group covariance matrix and the estimated error variances in the level one model. The regression coefficients are of course not directly comparable because the variables are measured on different scales. It may be noted, however, that the amount of variance explained by the general factor at the individual level varies considerably between the manifest variables.

The estimated relations at the group level are then presented, and these of course involve the manifest variables with a “2” as prefix. Again it must be observed that the coefficients

are not directly comparable across variables:

Unstandardized family level estimates for the two-level one-factor model for the van Peet data

```
2WRDLST  =    +1.00*gB          +1.00*2WRDLST&
2CARDS   =    +1.13*gB          +1.00*2CARDS&
2MATR    =    +0.74*gB          +1.00*2MATR&
2FIGS    =    +0.49*gB          +1.00*2FIGS&
2ANIMAL  =    +1.02*gB          +1.00*2ANIMAL&
2OCCUP   =    +0.53*gB          +1.00*2OCCUP&

Var(gB)      =    4.30
Var(2WRDLST&)=    1.56
Var(2CARDS&) =    4.14
Var(2MATR&)  =    0.00
Var(2FIGS&)  =    2.06
Var(2ANIMAL&)=    0.72
Var(2OCCUP&) =    5.27
```

The post-processor also computes and presents estimates of the contributions of variability between groups and within groups to the total variance of each manifest variable:

Estimates of sources of variance in manifest variables for the two-level one-factor model

```
Est var WRDLST    Total   15.40, between    5.86, within    9.54
Est var CARDS     Total   28.33, between    9.60, within   18.74
Est var MATR      Total   16.43, between    2.38, within   14.05
Est var FIGS      Total   21.18, between    3.08, within   18.10
Est var ANIMAL    Total   22.95, between    5.20, within   17.75
Est var OCCUP     Total   21.67, between    6.48, within   15.20

Per cent var      WRDLST          between   38.07, within   61.93
Per cent var      CARDS           between   33.87, within   66.13
Per cent var      MATR            between   14.48, within   85.52
Per cent var      FIGS            between   14.53, within   85.47
Per cent var      ANIMAL          between   22.65, within   77.35
Per cent var      OCCUP           between   29.88, within   70.12
```

The estimated covariance matrices at the both levels are computed, and from these the estimates of variance are derived. It is, of course, interesting to compare these estimates with the total variances and intraclass correlations computed from raw data by STREAMS. The general patterns of result agree quite well but there also are some slight deviations. These deviations are presumably due to the fact that the model does not fit the data particularly well.

For certain types of models STREAMS then presents a decomposition of sources of var-

iance in the sum of scores:

Estimates of sources of variance in the sum of scores for the two-level one-factor model

```
Estimated components of variance in sum of scores
 Within:
  g           88.83
  Error       73.56
 Between
  gB         103.50
  Error       13.75

 Estimated total within variance:   162.38
 Estimated total between variance:  117.25
 Estimated total variance:          279.64

 Estimated within reliability:        0.55
 Estimated between reliability:       0.88
```

The algorithms for computing the sources of variance in the sum of scores are the same as those used in ordinary one-level analysis, i. e. the contribution of a latent variable is a function of the square of the sum of unstandardized factor loadings and the factor variance. The error variance of the sum is the sum of the residual variances of the components (Reuterberg & Gustafsson, 1992). Because the individual and group levels are orthogonal the estimated total variance is simply the sum of the estimated variances at the two levels.

The post-processor also estimates an individual level (within) reliability, which is defined as the ratio of true individual variance and observed individual variance, and a group level (between) reliability, defined as the ratio of true group variance and observed group variance (see Gustafsson, 1997). It should be observed that these values are meaningful only when it is reasonable to construct an aggregate score which is the unweighted sum of the observed variables.

t-values

Next t-values for the significance of the estimates of free parameters are presented (note, however, that t-values are not available with Mx).

t-values for individual level estimates in the two-level one-factor model

```
t-values:

CARDS    =    +3.71*g
MATR     =    +3.63*g
FIGS     =    +2.56*g
ANIMAL   =    +2.04*g
OCCUP    =    +1.06*g

Var(g)       =    2.11
Var(WRDLST&) =    7.87
Var(CARDS&)  =    4.74
Var(MATR&)   =    3.73
Var(FIGS&)   =    8.27
Var(ANIMAL&) =    8.43
Var(OCCUP&)  =    8.59
```

The t-values inform us that almost all the estimated coefficients for relations between latent and manifest variables are significant.

For the group-level part of the model, however, only two of the residual variances of manifest variables are significant.

t-values for family level estimates in the two-level one-factor model

```
2CARDS   =    +3.43*gB
2MATR    =    +3.78*gB
2FIGS    =    +2.03*gB
2ANIMAL  =    +3.45*gB
2OCCUP   =    +1.82*gB

Var(gB)      =    2.26
Var(2WRDLST&)=    1.41
Var(2CARDS&) =    2.29
Var(2MATR&)  =    0.00
Var(2FIGS&)  =    1.51
Var(2ANIMAL&)=    0.51
Var(2OCCUP&) =    2.57
```

This indicates that one general family factor accounts for almost all variance at the group level.

Standardized Estimates

Finally, the standardized estimates are presented, first for the individual level, and then for the group level.

Standardized estimates for the two-level one-factor model

```
Standardized estimates:

WRDLST   =    +0.32*g          +0.72*WRDLST&
CARDS    =    +0.54*g          +0.61*CARDS&
MATR     =    +0.67*g          +0.64*MATR&
FIGS     =    +0.27*g          +0.88*FIGS&
ANIMAL   =    +0.19*g          +0.86*ANIMAL&
OCCUP    =    +0.09*g          +0.83*OCCUP&

2WRDLST  =    +0.53*gB         +0.32*2WRDLST&
2CARDS   =    +0.44*gB         +0.38*2CARDS&
2MATR    =    +0.38*gB         +0.01*2MATR&
2FIGS    =    +0.22*gB         +0.31*2FIGS&
2ANIMAL  =    +0.44*gB         +0.18*2ANIMAL&
2OCCUP   =    +0.24*gB         +0.49*2OCCUP&
```

The standardization is done with reference to the total variance in observed variables, which implies that the contributions from the two levels are expressed on the same scale. When all latent variables are orthogonal, as is the case here, the squares of the standardized coefficients should sum to unity, and it is easy to verify that this is true here. This standadization is thus not the same as that computed by Mplus and the other programs, which computes the standardized solution separately for the within- and the between-group matrices. However, as has been argued above, the two-level modeling approach should be conceived of as a variance decomposition of the total covariance matrix, which

implies that this matrix should be used in standardization. If this is not done, the differing amounts of variance accounted for by the group and individual levels in different variables is not taken into account.

When inspecting the results it may be observed that at individual level the MATR and CARDS tests have the highest relations with g, while at the family level WRDLST has the highest relation with gB. This suggests that the nature of the general factor may be different at individual and family level (see Härnqvist, et al., 1994). The fact that the model does not fit particularly well indicates that interpretations should be cautious, however, and that a better-fitting model should be found.

Modifying the Model

For purposes of comparison, it is interesting first of all to make an ordinary confirmatory factor analysis of the total covariance matrix. Such a matrix has been computed for the entire set of cases, which is in the folder *Group*, and has the dataset label *Tot*. A model with one general factor (model *t1*) does not fit the data (χ^2 (9) = 37.49, p < .00, RMSEA = .130), and according to modification indices the misfit is mainly caused by a covariance between ANIMAL& and OCCUP&. This may, tentatively, be interpreted in terms of a perceptual speed factor running through these tests. When such a factor (P) is introduced, the largest modification index identifies a relation between P and FIGS. When this relation is allowed as well, a very good fit is obtained for the two-factor model (model *t2*: χ^2 (6) = 5.14) The standardized estimates are presented below:

Standardized estimates for the two-factor model for the total matrix

```
Standardized estimates:

WRDLST   =    +0.56*g          +0.83*WRDLST&
CARDS    =    +0.73*g          +0.68*CARDS&
MATR     =    +0.74*g          +0.67*MATR&
FIGS     =    +0.35*g          +0.22*P          +0.91*FIGS&
ANIMAL   =    +0.38*g          +0.66*P          +0.64*ANIMAL&
OCCUP    =    +0.14*g          +0.53*P          +0.83*OCCUP&
```

The pattern of loadings on the general factor indicates that this factor may be interpreted as a Fluid Intelligence factor (see Gustafsson & Undheim, 1996), and both ANIMAL and OCCUP have quite substantial loadings on P. It is, thus, reasonable to try a model which at the individual level identifies the same two factors as were found in the analysis of the total matrix. A path diagram for this model is shown below:

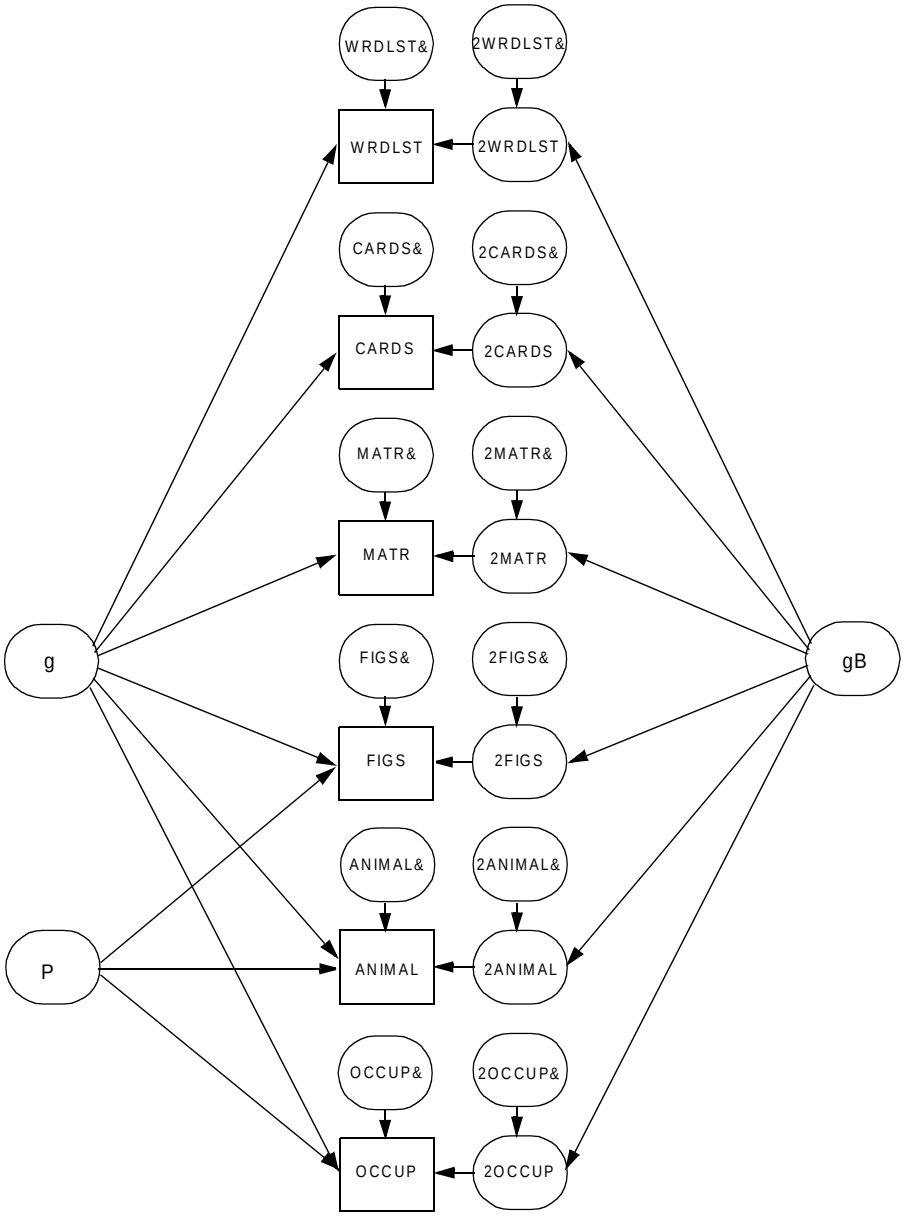


Figure 5. *A path diagram for the modified two-level model for the van Peet data*

The MB specification for this model is:

MB instructions for the two-level two-factor model

```
POP All
DAT FOLDER=None DATLAB=AllB POP=All
DAT FOLDER=None DATLAB=AllW POP=All
MVR WRDLST CARDS MATR FIGS ANIMAL OCCUP
LVR g P
LV2 gB
REL g -> WRDLST CARDS MATR FIGS ANIMAL OCCUP
REL gB -> 2WRDLST 2CARDS 2MATR 2FIGS 2ANIMAL 2OCCUP
REL P -> ANIMAL OCCUP FIGS
```

This model (*g2*) fits the data excellently (χ^2 (15) = 15.89, p < .39, RMSEA = .018) and the standardized estimates are presented below:

```
Standardized estimates:

WRDLST   =    +0.31*g          +0.72*WRDLST&
CARDS    =    +0.55*g          +0.60*CARDS&
MATR     =    +0.68*g          +0.63*MATR&
FIGS     =    +0.24*g          +0.25*P          +0.85*FIGS&
ANIMAL   =    +0.15*g          +0.65*P          +0.57*ANIMAL&
OCCUP    =    +0.04*g          +0.50*P          +0.67*OCCUP&


2WRDLST  =    +0.53*gB         +0.32*2WRDLST&
2CARDS   =    +0.45*gB         +0.38*2CARDS&
2MATR    =    +0.39*gB         +0.00*2MATR&
2FIGS    =    +0.20*gB         +0.33*2FIGS&
2ANIMAL  =    +0.43*gB         +0.22*2ANIMAL&
2OCCUP   =    +0.19*gB         +0.51*2OCCUP&
```

The pattern of relations between the cognitive tests and the two general factors is quite interesting. At the individual level the non-verbal reasoning test MATR has the highest relation with the general factor, but at the family level the verbal test WRDLST test has the highest relation with the general factor. It thus seems that at the individual level the general factor has an interpretation which comes close to the dimension labeled Fluid Intelligence, while at the family level the general factor is closer to the dimension labeled Crystallized Intelligence. Härnqvist et al. (1994) report a similar finding in an analysis of ability structures at class and individual levels. It should be observed, however, that when the standardization is done separately for the between- and within-group matrices this pattern of results does not appear. In the standardization computed by Mplus, for example, the 2MATR variable thus

When two-level models are fitted it is often practical to work with one level at a time. In order not to impose any structure at the between level, the variables at this level may be taken to be freely correlated, i. e:

MB instructions for the two-level two-factor model without any structure at the family level

```
POP All
DAT FOLDER=None DATLAB=AllB POP=All
DAT FOLDER=None DATLAB=AllW POP=All
MVR WRDLST CARDS MATR FIGS ANIMAL OCCUP
LVR g P
REL g -> WRDLST CARDS MATR FIGS ANIMAL OCCUP
REL P -> ANIMAL OCCUP FIGS
COV 2WRDLST 2CARDS 2MATR 2FIGS 2ANIMAL 2OCCUP
```

This model has, of course, a very good fit (χ^2 (6) = 4.18, p < .65). This baseline model may also be used to test if one general family factor is sufficient to account for the covariances at the family level, through taking the difference between test statistics in the usual way. This test is not significant (χ^2 (9) = 11.71), so the hypothesis that a one-factor model is sufficient cannot be rejected.

It must be observed, however, that because the number of families and the number of tests

is so limited in the present data it is not possible to draw any strong conclusions about the number of factors at the family level. Several attempts have been made to impose a two-factor model at the family level as well, but all these attempts were unsuccessful, either because the model did not converge, or because negative estimates were obtained of the factor variance. It is, however, the case that for three of the tests (CARDS, FIGS, and OCCUP) there remains a significant amount of variance at the family level which is not accounted for by the gB factor. This residual variance may be due to errors of measurement or to systematic sources of variance, but with the present data it does not seem possible to get any further information.

Hox (1994, pp. 90-91) also arrived at a model with two factors at the individual level, and one factor at the family level. Hox fitted, however, an oblique two-factor model at the individual level, and interpreted one factor as "reasoning" and the other as "fluency." The general factor at the family level was interpreted as a factor of general intelligence. The interpretation of this factor was, however, based on the within-group completely standardized solution computed by LISREL (and Amos and Mplus), under the assumption that a two-group model has been specified. This standardization is quite different from that computed by STREAMS, which is based on the total estimated variance of the manifest variables. According to the two-group standardization solution the highest loading on gB is obtained for 2MATR (1.02), next highest for 2ANIMAL (.86), and the third highest for 2WRDLST (Hox, 1994, p. 93). These results are very different from those presented above, both in terms of the absolute level of the estimates, and in terms of the rank-ordering of the tests. The reason for this is, of course, that the two-group standardization computed by Amos, LISREL and Mplus does not take into account the different amounts of variance contributed by family variability in the different tests. For the proper interpretation of results it seems, however, that the intraclass correlations must be taken into account.

Examples of Two-Level Models

We will now briefly present some further examples of two-level models.

Two-Level Confirmatory Factor Analysis

Two-level confirmatory factor analysis models are of great interest in themselves, and they are also important as measurement models in two-level structural equation models. The model for the van Peet data which has already been described is an example of a two-level confirmatory factor model, and we will discuss some further examples of such models. The first set of examples illustrate modeling of data from multiple populations, and it should be remembered that STREAMS does not support such models for Mplus. It should also be observed that Amos 4 does not estimate two-level models with means. This is because the program checks to see if a mean vector is supplied for each group, which is not the case for the pooled-within matrices. Nor does Mx estimate two-level models with means.

A Two-Level Measurement Model in Multiple Populations

As another confirmatory factor analysis example we will consider a measurement model fitted in three populations, and here too we will rely on data from Hox (1994). He presents

an example of a two-level path analysis using data collected by Schijf and Dronkers in 1971 (references missing in Hox), and we will use these data in several examples.

The Schijf and Dronkers study comprised 1379 pupils in 58 schools of three different denominations (Protestantic, Nondenominational, and Catholic). The major purpose of the study was to see if school denomination affects pupil achievement and teacher's advice to students about secondary education, controlling for the home background of the students. At the pupil level the following variables are available: father's occupational status (FOCC) and education (FEDUC), mother's education (MEDUC), family size (FSIZE) gender (SEX), number of repeated classes (REPEAT), score on an achievement test (GALO), and teacher's advice (ADVICE). At the school level only the denomination variable is available (DENOM, protestant=1, nondenominational=2, catholic=3). These data (copied from Hox, 1994) are available in the directory *STREAMS\EXAMPLES\DENOM* (or in the *DENOM.ZIP* file in the installation directory) in a file under the name *denom.raw* along with a data dictionary.

A project called *Denom* has also been created in this directory. In the folder *None* matrices for two-level analysis have been computed for the total set of data, with the group label *Tot*. The between school matrix is based on 58 observations, and the pooled within matrix on 1321 observations. The between matrix includes the group-level manifest variable DENOM. In the folder *Den* separate sets of matrices have also been computed for each of the three denominations. The *Prot* group comprises 10 schools, and 192 individual observations; the *NonD* group 39 schools and 921 individual observations; and the *Cath* group 9 schools and 208 individual observations. The number of schools is much smaller than the minimum recommended number (30-50) so the analyses reported here should mainly be seen as illustrations of the technique.

We will first consider the four variables which are hypothesized to measure a latent socioeconomic status (*Ses*) variable (i. e., FOCC, FEDUC, MEDUC, and FSIZE), and investigate if the measurement model is invariant over the three populations of schools. A model with one *Ses* variable at the individual level and a *2Ses* variable at the school level (*bt1*) was first fitted to the entire set of schools, and this model fits very well (χ^2 (4) = 4.93, p < .29, RMSEA = .013). All four programs produce the same results, except that the LISREL RMSEA estimate is higher (.018) than the estimate computed by Amos and Mx (0.13), and by Mplus (.010).

In the next step a three-population model without means was fitted, with constraints of equality over populations on every parameter (*bpi*):

MB instructions for the two-level one-factor model for three populations

```
POP Prot NonD Cath
DAT FOLDER=Den DATLAB=ProtB POP=Prot
DAT FOLDER=Den DATLAB=ProtW POP=Prot
DAT FOLDER=Den DATLAB=NonDB POP=NonD
DAT FOLDER=Den DATLAB=NonDW POP=NonD
DAT FOLDER=Den DATLAB=CathB POP=Cath
DAT FOLDER=Den DATLAB=CathW POP=Cath
MVR FOCC FEDUC MEDUC FSIZE
LVR Ses
LV2 2Ses
REL Ses -> FOCC FEDUC MEDUC FSIZE
REL 2Ses -> 2FOCC 2FEDUC 2MEDUC 2FSIZE
```

This model does not fit quite as well (χ^2 (44) = 89.86, p < .00, RMSEA = .028; χ^2 = 91.49 was obtained with Amos), which indicates that there may be differences in the measurement model over the populations. The same model without constraints over populations (*bpf*) has an excellent fit (χ^2 (12) = 9.49, p < .66; χ^2 = 9.78 with Amos), and the standardized estimates from that model are presented below:

Standardized estimates for the two-level one-factor model for three populations

```
Standardized estimates:

Prot     FOCC     =    +0.58*Ses        +0.70*FOCC&
NonD     FOCC     =    +0.49*Ses        +0.71*FOCC&
Cath     FOCC     =    +0.61*Ses        +0.67*FOCC&

Prot     FEDUC    =    +0.82*Ses        +0.34*FEDUC&
NonD     FEDUC    =    +0.75*Ses        +0.33*FEDUC&
Cath     FEDUC    =    +0.79*Ses        +0.44*FEDUC&

Prot     MEDUC    =    +0.65*Ses        +0.68*MEDUC&
NonD     MEDUC    =    +0.51*Ses        +0.70*MEDUC&
Cath     MEDUC    =    +0.57*Ses        +0.74*MEDUC&

Prot     FSIZE    =    -0.18*Ses        +0.93*FSIZE&
NonD     FSIZE    =    -0.07*Ses        +0.97*FSIZE&
Cath     FSIZE    =    -0.05*Ses        +0.99*FSIZE&

Prot     2FOCC    =    +0.41*2Ses       +0.06*2FOCC&
NonD     2FOCC    =    +0.49*2Ses       +0.10*2FOCC&
Cath     2FOCC    =    +0.41*2Ses       +0.13*2FOCC&

Prot     2FEDUC   =    +0.45*2Ses       +0.12*2FEDUC&
NonD     2FEDUC   =    +0.59*2Ses       +0.00*2FEDUC&
Cath     2FEDUC   =    +0.40*2Ses       +0.17*2FEDUC&

Prot     2MEDUC   =    +0.32*2Ses       +0.11*2MEDUC&
NonD     2MEDUC   =    +0.51*2Ses       +0.07*2MEDUC&
Cath     2MEDUC   =    +0.39*2Ses       +0.00*2MEDUC&

Prot     2FSIZE   =    -0.12*2Ses       +0.29*2FSIZE&
NonD     2FSIZE   =    -0.10*2Ses       +0.23*2FSIZE&
Cath     2FSIZE   =    +0.03*2Ses       +0.14*2FSIZE&
```

It is interesting to see that both at the individual level and at the school level the pattern of relations is, in spite of the small samples, similar over populations. At both levels there is a tendency for the FSIZE variable to be less highly related to the latent variable in the Cath population than in the other two populations, which may be reasonable given the different views on family planning within the populations. We may subject this observation to a more formal statistical test with the following model (*bpff*):

MB instructions for testing invariance of loadings over populations

```
MVR FOCC FEDUC MEDUC FSIZE
LVR Ses
LV2 2Ses
REL Ses -> FOCC FEDUC MEDUC FSIZE
REL 2Ses -> 2FOCC 2FEDUC 2MEDUC 2FSIZE
REL (Prot NonD Cath) Ses -> FSIZE
REL (Prot NonD Cath) 2Ses -> 2FSIZE
```

This model fits as well as the model without any constraints of equality (χ^2 (16) = 12.93, p < .68; χ^2 = 13.33 with Amos), so there are not any significant differences over populations in the size of the loadings of FSIZE on *Ses*. Further analyses reveal, however, the significant difference between populations to be due to the error variance in FSIZE at the individual level. When the following statement is added to the completely constrained model a good fit is obtained (χ^2 (42) = 37.13, p < .68):

```
VAR Prot NonD Cath FSIZE&
```

The difference between the test statistic for the completely constrained model and this test statistic gives a test of the differences over populations in the individual level error variance of FSIZE (χ^2 (2) = 52.73, p < .00).

Differences in Means on Latent Variables

We will use the same data to illustrate how differences in means on latent variables may be modeled in multiple-population two-level models (see also Muthén, Khoo, & Gustafsson, in press). In the first step means were simply added to the model arrived at in the previous section, through clicking the check-box **Include Means in Model** on the **Model Type** tab on the *Model* form. Estimating this model (*bpm*) with Amos produces the following goodness of fit information from the post-processor:

Goodness-of-fit test when means are included in the two-level model

```
Goodness of Fit Test:

Chi-square = 70.79, df = 50, p < .03

RMSEA = .017, p-value for RMSEA < 0.05 = 1.00

Fit Indices: NFI = .96, NNFI = .99, CFI = .99
```

It should be observed that Mx does not handle two-level models with means, and also that a somewhat lower value of the test statistic is obtained with LISREL (66.86).

The value of the test statistic is close to the degrees of freedom, which indicates that there are no important differences in the means. A formal statistical test of the population differences in means on the latent variable *Ses* may, however, be obtained through comparison with a model in which the constraints on the latent variable means have been relaxed (*bpmf*):

```
MEA NonD Cath 2Ses
```

Observe that in two-level models there only are differences in means for variables at the group level. The test statistic for this model (χ^2 (48) = 65.49 with LISREL and χ^2 (48) = 69.26 with Amos) is, however, close to the test statistic for the constrained model, so we may conclude that there is no significant difference in means with respect to *Ses* over the populations.

We may, however, again suspect that there are differences between the populations with respect to FSIZE, even though there are no differences between populations with respect to the other three observed variables. To allow differences in family size between populations we may add the statement:

```
MEA NonD Cath 2FSIZE
```

This statement causes the model (*bpmf2*) to have a much better fit (χ^2 (46) = 38.29 with LISREL and χ^2 (46) = 39.98 with Amos) which thus supports the hypothesis. The following unstandardized estimates are obtained:

Estimates of means on the family size variable

```
Prot     Intercept(2FSIZE)  =    3.66
NonD     Intercept(2FSIZE)  =    3.03
Cath     Intercept(2FSIZE)  =    4.13
```

As expected the means are higher in the *Cath* population.

Two-Level Models Involving Structural Relations

We now will consider some examples of models which involve relations at the group level, either with aggregated individual variables, or with group-level manifest variables.

A Two-Level Model with Relations Among Latent Variables

In order to illustrate how a two-level model with relations among latent variables may be formulated we will bring in some other variables in the *Denom* project as well. These analyses are conducted with all schools pooled.

Hox (1994) presents a rather elaborate path model which involves both manifest and latent variables. The between-school part of this model presents great problems in estimation, however, so here a simpler approach is taken. Socio-economic status (*Ses*) is represented at both individal and school level, using the three indicators FOCC, FEDUC MEDUC. Achievement (*Ach*) is measured by two indicators: GALO and ADVICE. In the model *Ach* is regressed upon *Ses* at both school and individual level as is shown in the path diagram below.

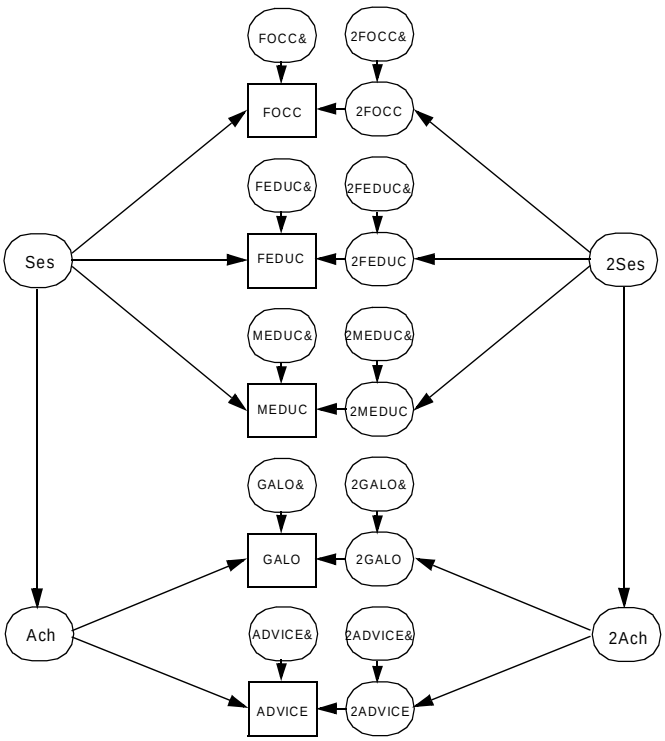


Figure 6. *A path diagram for the two-level regression model*

The MB statements for this model (*str1*) are:

MB instructions for the two-level regression model with latent variables

```
POP Tot
DAT FOLDER=None DATLAB=TotB POP=Tot
DAT FOLDER=None DATLAB=TotW POP=Tot
MVR FOCC FEDUC MEDUC GALO ADVICE
LVR Ses Ach
LV2 2Ses 2Ach
REL Ses -> FOCC FEDUC MEDUC
REL 2Ses -> 2FOCC 2FEDUC 2MEDUC
REL Ach -> GALO ADVICE
REL 2Ach -> 2GALO 2ADVICE
REL 2Ses -> 2Ach
REL Ses -> Ach
```

These simple and straightforward instructions create a model which also is easy to estimate, and which fits quite well (χ^2 (8) = 20.75, RMSEA = .034 for LISREL; χ^2 (8) = 20.81 for Amos; χ^2 (8) = 20.84, RMSEA = .030 for Mplus). The following standardized estimates are obtained:

Standardized estimates for the two-level regression model with latent variables

```
Standardized estimates:

Ach      =    +0.37*Ses        +0.93*Ach&
FOCC     =    +0.53*Ses        +0.70*FOCC&
FEDUC    =    +0.73*Ses        +0.40*FEDUC&
MEDUC    =    +0.56*Ses        +0.69*MEDUC&
GALO     =    +0.82*Ach        +0.42*GALO&
ADVICE   =    +0.85*Ach        +0.38*ADVICE&

2Ach     =    +0.89*2Ses       +0.45*2Ach&
2FOCC    =    +0.46*2Ses       +0.11*2FOCC&
2FEDUC   =    +0.55*2Ses       +0.00*2FEDUC&
2MEDUC   =    +0.46*2Ses       +0.07*2MEDUC&
2GALO    =    +0.37*2Ach       +0.15*2GALO&
2ADVICE  =    +0.39*2Ach       +0.00*2ADVICE&
```

It should be observed that the standardization is done in different ways for manifest and for latent variables. For manifest variables the standardization is done with reference to the total variance (i. e., the sum of squares of standardized loadings over both levels sum to unity), while for latent variables the standardization is done within each level.

The standardized estimate of the relation between *Ses* and *Ach* is .37 at the individual level, which is close to what has been found in other studies. At the school level, however, the relation is much higher (.89) so a considerable part of the variability in level of performance at the school level is correlated with the socio-economic level of the students. Quality of scholing may, of course, also be expected to be correlated with both socio-economic level and achievement, so the present data do not allow any strong conclusions about causality.

A Two-Level Model with a Group-level Manifest Variable

So far the two-level models have only included aggregated individual variables at the group level. As has already been mentioned it is, however, also possible to include manifest variables which are only observable at the group level, such as characteristics of schools and teachers. Such group-level manifest variables may be invoked as observed independent or dependent variables, or they may be used to define group-level latent variables.

Hox (1994) used the DENOM variable as a group-level manifest variable, and we will do that here too, even though this variable with its three categories is perhaps not optimal as an independent variable. We will start with the simplest possible model in which DENOM is used as a group-level manifest variable to predict the latent *Ach* variable. Because there are only two indicators of *Ach* (i. e., GALO and ADVICE) it is necessary to impose equality constraints on the relations between *Ach* and the two manifest variables. A path diagram is shown below:

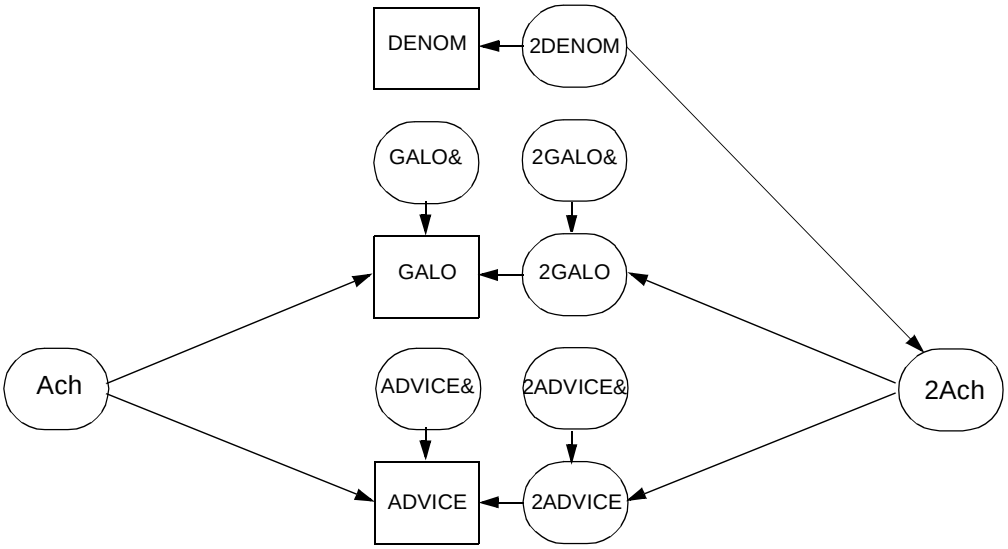


Figure 7. *A path diagram for a model with an observed group-level variable as independent variable*

The MB instructions for this model are:

MB instructions for the two-level regression model with a group-level manifest variable

```
MVR GALO ADVICE
LVR Ach
LV2 2Ach
MV2 DENOM
REL Ach -> (GALO ADVICE)
REL 2Ach -> (2GALO 2ADVICE)
REL 2DENOM -> 2Ach
```

It should be observed that the DENOM variable in the MB statements is referred to as 2DENOM, which is because this variable is only observed at the group level. This model

(*slr*) fits excellently:

Goodness-of-fit statistics for the two-level regression model with a group-level manifest variable

```
Goodness of Fit Test:

Chi-square = 1.67, df = 1.

RMSEA = .022, p-value for RMSEA < 0.05 = .71

Fit Indices: GFI = 1.00, NFI = 1.00, NNFI = 1.00, CFI = 1.00
```

When the model is estimated with Amos a marginally higher test-statistic is, as usual, obtained (χ^2 (1) = 1.70), which value is also arrived at by Mplus. The estimate of the relation between 2DENOM and *2Ach* is .30, with a t-value of 2.41. This result thus indicates that there is a difference in level of performance of schools of different denominations, with Catholic schools achieving the best results. The following standardized estimates are obtained:

Standardized estimates for the two-level regression model with a group-level manifest variable

```
Standardized estimates:

GALO     =    +0.87*Ach         +0.30*GALO&
ADVICE   =    +0.79*Ach         +0.47*ADVICE&

2Ach     =    +0.33*2DENOM      +0.94*2Ach&
2GALO    =    +0.40*2Ach        +0.00*2GALO&
2ADVICE  =    +0.36*2Ach        +0.13*2ADVICE&
```

The standardized coefficient for the regression of *2Ach* on 2DENOM is .33, which implies that about 11 % of the variability in school performance is due to denomination.

We may, however, also extend the model to control for socio-economic differences between schools:

MB instructions for the two-level model with a group-level manifest variable and control for Ses

```
MVR FOCC FEDUC MEDUC GALO ADVICE
MV2 DENOM
LVR Ses Ach
LV2 2Ses 2Ach
REL Ses -> FOCC FEDUC MEDUC
REL 2Ses -> 2FOCC 2FEDUC 2MEDUC
REL Ach -> GALO ADVICE
REL 2Ach -> 2GALO 2ADVICE
REL 2Ses -> 2Ach
REL Ses -> Ach
REL 2DENOM -> 2Ach
COV 2Ses 2DENOM
```

This model (*slraa*) too fits reasonably well (χ^2 (11) = 25.27 with LISREL, χ^2 (11) = 25.40 with Amos, and χ^2 (11) = 25.44 with Mplus). In this model, however, the coefficient for the regression of *2Ach* on 2DENOM is lower (.12 unstandadized, .15 standardized) and not significant (t = 1.76). This indicates that it is not the school's denomination itself that is important for achievement but the socio-economic characteristics of the students that attend the different categories of schools.

Some Issues in Two-Level Modeling

The STREAMS implementation of Muthén's solution of the two-level modeling problem makes it possible to address a whole new class of questions in empirical research. This should make for more powerful and interesting analyses of phenomena in many fields, such as education, sociology, organization and political science.

It must be realized, however, that there is only limited experience how to fit and interpret two-level models, and it also must be realized that the estimation techniques are approximate only. Thus, one typical characteristic of two-level models is that there is an assymetric amount of information on the two levels, there often being a large amount of observation at the individual level, and only few observations at the group level. This causes difficulties when goodness-of-fit statistics are interpreted, and there is also the problem that the estimation technique is a large sample technique. Another problem has to do with the assumption that group sizes are equal, and there is as yet only very little information available how robust the estimator is against violations of this assumption. It would seem, however, that the most essential next step is to apply the two-level modeling techniques to a wide range of empirical data in order to gain experience of the possibilities and problems of this new method of analysis.

7 Issues of Efficiency

Whenever the number of variables and/or groups is large in relation to the capacity of the computer that is being used, issues of efficiency do become central. Modeling is typically a highly interactive activity, and when execution times get too long, or the estimation process fails too often, this activity may become more frustrating than rewarding. The present chapter presents some ways in which STREAMS may be used to avoid such frustrations.

The iterative solution of the equations sometimes fails to converge on the correct solution, and this seems at present to be one of the major problems facing users of SEM programs. For large and complex models in particular the user may have to spend many hours trying to obtain a solution. STREAMS offers, however, some facilities for coming to grips with this problem. This is based on the functions for copying start values from previously estimated models into the setup for a new model. However, this is no fail-safe mechanism for obtaining convergence, and to use these facilities it is necessary to be aware of their intended use and ways of operating.

Reasons for Non-Convergence

The iterations may fail to converge for several different reasons, so different types of actions may have to be taken.

Unidentified model

In an unidentified model unique estimates cannot be obtained for all parameters. The estimation programs are able to identify at least some problems of non-identification, and if there is a message about non-identifiability of a certain parameter this problem must be solved through respecifying the model.

Over-parameterized model

When the model is over-parameterized there is, in principle, a unique estimate of each parameter in the model, but the number of parameters is large in relation to the number of elements in the matrix analysed. This makes it difficult to find a proper solution even when good start values are available. In most cases the best solution is to reduce the number of parameters to be estimated by imposing stronger constraints on the model, even at the expense of a somewhat poorer nominal fit. At any rate, models with a large number of parameters should be avoided in the early stages of modeling.

Poor start values

When the start values which are supplied to the estimation program, or which are computed internally in the estimation program, are too far away from the actual parameter values, the program may fail to find a solution within the maximum number of iterations allowed. Increasing the maximum number of iterations may solve the problem, but even with a very large number of iterations the program may be unable to find the solution.

Small sample of cases

Simulation studies by Boomsma (1985) indicate that fitting a model to a small sample of cases (N < 100, say) entails an increased risk of nonconvergence.

Strategies to be Followed

A large number of steps and actions may be taken to increase the likelihood of obtaining a correct solution.

Equalize variances

Observed variables which are measured on widely different scales make it more difficult for the iterations to converge. When data is prepared for analysis it is therefore good practice to ensure that the variances of the observed variables are not too different. Those variables which have a large variance should be rescaled by dividing the score for each person with a constant (*c*, say). Optimally every observed variable should have a variance around 1. Observe that the square of *c* determines the variance of the transformed variable. For example, if an observed variable has a variance of 100, choosing *c* to be 10 will cause the rescaled variable to have a variance of unity.

When matrices are computed from raw data, the most easy way to accomplish the rescaling is to change the number of implied decimal points (see Chapter 3) either in SPSS or in the *Define Variables* form. For example, if a variable has values that range between 100 and 859 it may be a good idea to declare 2 implied decimals. This will cause the value 100 to be interpreted as 1.0 and the value 859 to be interpreted as 8.59. When Mplus is used with raw data as input, the **Define** command on the *Options* form may be used to rescale a variable (e. g., VAR1 = VAR1/100).

Redefine the scale of the latent variable

The manifest variable chosen to define the scale of a latent variable should optimally be the manifest variable with the highest relation to the latent variable. If the scaling variable has a low relation to the latent variable, estimates of the coefficients for the other manifest variables will be high, and difficult to compute. This problem may be identified through inspecting the estimates of the intermediate solution. If high estimates are observed for all free parameters which express relations between manifest and observed variables, another scaling variable should be selected. This is done through clicking the **SCL** button and selecting an appropriate pair of manifest and latent variables. Remember also that STREAMS uses the first manifest variable which is defined to have a relation to a latent variable as the scaling variable. Thus, when the *Relations* form is used, the first manifest variable to be moved to the box for dependent variables should be the one with the highest expected relationship with the latent variable.

Impose equality constraints

One extremely useful strategy is to impose strong constraints on the first model through having equality constraints over variables and groups. This will cause poor model fit, but it will greatly improve chances that iterations will converge. Once a solution has been obtained, the parameter estimates of the constrained model furnish start values for a less constrained model if one of the "copying" options on the **Start Values** tab of the *Model* form has been chosen. Even an extremely poor-fitting model will usually provide start values which are useful when the constraints on the model are relaxed. The copying of start values functions automatically and transparently to the user. It is recommended, however, that once a solution has been obtained, and occasionally thereafter, a new model name is chosen, and that the option **Copy from Previously Estimated Models** is chosen. If a model does not converge, the parameter estimates may get corrupted, and if this happens it is no longer possible to copy start values from this model. In this case the checkbox labelled **Copy from Previous Model with Same Name** should be deselected, and not be clicked again until the proper estimates have been obtained. If this situation should occur it is, of course, good to have one or more other models available from which start values may be copied.

Develop the model incrementally

Another possibility is to add latent and/or manifest variables successively. In this way start values may be copied from one or more smaller models which have been fitted previously. This strategy is not successful, however, unless the same labels are used for corresponding latent variables from one model to another. This is because the pre-processor relies on the labels of latent and manifest variables when the program determines for which parameters start values may be copied. When the pre-processor determines that no start value is available for a parameter it supplies a more or less arbitrary start value. At present .7 is used for relations and variances, and .1 is used for covariances. These guesses may be quite far off, however, and a few poor guesses may cause the iterative solution to fail. If this happens the guessed start values may be edited in the instruction file created by the pre-processor, and a new attempt may be made to run the estimation program. (Remember to turn the pre-processor off before clicking **Run** because otherwise the newly produced file will destroy the file of edited instructions.) The incremental modeling strategy is particularly useful when a large model may be broken down into several sub-

models. When the submodels are put together only a few parameter values for relations between the submodels need to be guessed. Here too it is useful to combine submodels in a step-wise fashion.

Fit the model in one group first

When the model comprises several groups of cases, or if a one-group model is to be replicated in several groups, it is always a good idea to fit the model in a single group first, because a one-group model is smaller and easier to estimate. The estimates for this group are then automatically applied to the other groups, when one of the copying options is selected.

Avoid over-fitting

One of the most important principles of successful structural equation modeling is that models should not be over-fitted. When parameters are added in order to achieve "good model fit" this often causes the modeler to go too far, and introduce parameters which only represent trivial or random sources of variance. Such over-fitting not only causes the model to fail to replicate in a new sample, but it also causes the model to be unstable and difficult to estimate. There is thus a trade-off between the fit and the stability of models, and generally a somewhat poorer model fit is to be preferred to an over-fitted model. Recently a set of very useful statistics has been developed which assist the modeler in making the decision when to stop modeling (see Bollen & Long, 1993; Jöreskog & Sörbom, 1993c, Chapter 4).

Select Another Estimation Program

It is often the case that a model which is difficult to estimate with one estimation program is quite easily estimated with another program. With STREAMS it is trivial to switch between estimation programs, and once a solution has been obtained with one program, that solution may be used as source of start values for another program.

An Example

Some of the recommendations mentioned above will be illustrated with an example. In order to create difficulties for the estimation program a variable is used which has an extremely large variance in comparison with the other observed variables. Raw data on the six sub-tests in the Swedish Scholastic Aptitude Test have been imported into *hpg* project in the *Examples* directory under the categorization variable *Raw* and the group label *Tot*. Along with these six variables (i. e., WORD, DS, READ, DTM, GI and ERC) the measure of mean grade from secondary school has been included (MRK). The MRK variable is measured on a scale between 1 and 5 with 2 decimals. The decimal point is, however, implied in the data values, and here 0 decimals have been assumed. When interpreted this way the variable has a variance which is 10 000 times as large as when 2 decimals are assumed.

Let us assume that we want to fit a model with two orthogonal factors, one general (*Gen*) and one verbal (*Verb*) to the 7 variables. This may be done with the following MB state-

ments:

```
REL Gen -> WORD DS READ DTM GI ERC MRK
REL Verb -> WORD READ GI ERC MRK
```

This model has been estimated with the Mplus 1.0 program. When the default start values supplied by Mplus are relied upon the model does not converge within 5000 iterations. In order to obtain a set of start values which may be copied into other models, constraints of equality have therefore been imposed on the relations between each of the latent variables and all the manifest variables. In order to impose equality constraints for the relations between both latent variables and all the manifest variables, the following statements may be used:

```
REL Gen -> (WORD DS READ DTM GI ERC MRK)
REL Verb -> (WORD READ GI ERC MRK)
```

The statements for the unconstrained relations are, of course, easily transformed into the statements with equality constraints through clicking the appropriate **Equality** check-box on the *Relations* form. Another possibility is to use the *Set Constraints* form on the **Edit** menu (see “Set Constraints”, page 51). The table presents results from a series of attempts to fit the model with Mplus under the different approaches.

TABLE 1. **Results from comparisons of different methods for computing start values**

Model	No start values Number of iterations	Copying from model 1 Number of iterations	Chi-square	df
1. Equality constraints for *Gen* and *Verb*	907	1	292.8	19
2. Equality constraints for *Verb*	NC	38	124.0	13
3. Equality constraints for *Gen*	NC	38	201.0	15
4. No equality constraints	NC	48	44.11	9

Note. NC means that no convergence of iterations was obtained

As may be seen in the table the iterations converge when equality constraints are imposed on all relations between latent and manifest variables, even though no less than 907 iterations are required. When the model is estimated once again using start values from this solution, convergence is, as may be expected, immediately obtained. When the equality constraints are relaxed for *Gen* but kept for *Verb* the model does not converge when not given start values, but only 38 iterations are needed when start values are copied from the previous model. When there are equality constraints for *Gen* but not for *Verb* the model also fails converge without start values, which is also true when no equality constraints are imposed. However, with start values from the highly constrained model 1, the model with equality constraints on *Gen* converges in just 38 iterations, while the model without any equality converges in 48 iterations. It is interesting to observe that the model without any constraints fits quite well, which is not true for the other models. However, the constrained and poor-fitting models seem to be extremely useful in the process of obtaining a well-fitting model.

One of the reasons why Mplus fails to converge when it is not supplied with good start values is that a less than optimal scaling variable (WORD) was used for the latent variables. When MRK is taken to be the scaling variable for both latent variables Mplus does indeed converge after 3137 iterations without start values. This demonstrates the impor-

tance of using a manifest variable with a high relation to the latent variable as the scaling variable.

Part 3

Managing Projects and Data

The third part of the User's Guide presents the functions for managing projects and data includes in the STREAMS system.

8 Preparing Data and Creating Projects

STREAMS stores the data to be analyzed in a data base, along with descriptive information about the data. This makes it possible for STREAMS to take care of the practical details of data specification for the different SEM programs.

STREAMS uses the *project dictionary* to store information about:

- Labels of codes values, variables and data sets.
- Missing data codes.
- Number of cases in data sets.
- Types of data (i. e., rawdata or different types of matrices).
- Variables included in a particular matrix or data set.

To store and retrieve the datasets STREAMS uses a simple two-level hierarchical system, with one or more *folders* (e. g., Gender), each of which may contain one or more datasets, which are identified with *dataset labels* (e. g., Males and Females). Addition of data to a STREAMS project thus involves tasks such as assigning labels to folders, datasets and variables.

The SEM programs typically analyze matrices of measures of interrelationships (e. g., covariances, correlations, or polychoric correlations) between the variables, and often they do not require access to rawdata. It is thus often convenient and efficient to prepare these matrices in a first step.

Forms of Data

Unless the estimation program specifically requires rawdata as input (such as, for example, the robust estimation procedures in EQS and Mplus, or the missing data estimation algorithms), it is usually preferable to compute covariance matrices in a first step. This is

more efficient because the matrix needs not be recomputed each time a model is fitted for a set of variables. The matrix to be analyzed may, of course, also already be available, because it has been computed previously.

There are several different ways to compute a matrix from rawdata:

- *Through programs in SPSS, SAS and other systems for statistical analysis.* If this method is used, the matrix to be analyzed must be imported into STREAMS using procedures described in chapter 10. The same is true if the matrix is copied from another source, such as a book or a journal.
- *Through the PRELIS2 program.* This program, which is a part of the LISREL system, may be used to compute a wide range of measures of association between variables (e. g., covariances, correlations, polychoric correlations) and other input which is needed for the different estimators supported by LISREL 8. PRELIS2 may also be used to describe data, control for missing values, transform variables, sum variables, impute missing data, along with a wide range of other data handling tasks. After matrices have been computed with PRELIS2 they must be imported into the STREAMS project.
- *Through STREAMS.* There are facilities in STREAMS which may be used to compute covariance matrices for subsets of cases and variables in an easy fashion and also to compute the special matrices needed for two-level analyses and missing-data models. STREAMS also automatically imports the matrices into the project dictionary so they are immediately accessible for analysis.

Below the different forms of input to STREAMS are described.

External Matrices

Often a correlation or a covariance matrix is available, but not the rawdata. The matrix may thus have been published in a book or journal, or it may have been located in an old computer printout. Such a matrix may can be imported using procedures described in the chapter “Importing Raw Data and Matrices” on page 163.

Rawdata in Text Format

If the rawdata only exists as a text-file (which should have the suffix *.raw*) information about the variables (labels, location in data file, missing data codes, and so on) should be entered. STREAMS offers procedures, which are described below, through which a so called STREAMS data dictionary may be created. The information is stored in a file which has the same prefix as the rawdata file, and which has the suffix *.sdd*. After the data file has been described, several different procedures may be used to compute covariance matrices or other types of matrices. Covariance matrices and matrices for two-level analysis may be computed with the built-in functions which also store the computed matrices in the project dictionary. Alternatively, subsets of cases and/or variables may also be imported as rawdata into the project dictionary. In the project dictionary the data in text format is automatically transformed into the SPSS *.sav* format, and stored in that form. These files are stored externally to the database, and they have a name which consists of the folder label and the dataset label connected with an underscore (e. g., *gender_males.sav*).

Rawdata in SPSS or PRELIS2 Format

If the rawdata have been read into SPSS or PRELIS2, STREAMS can access the SPSS *.sav* file and the PRELIS2 *.psf* file directly. Thus, matrices may be computed and imported into the project dictionary, or rawdata may be imported into the project dictionary, in the same way as is described above.

Rawdata in Other Statistical Systems

If SPSS or PRELIS2 is available but the data is stored in another format it may be possible to transfer the data directly into one of these programs, and from there to STREAMS. SPSS thus reads several different types of files, such as EXCEL and dBASE formats, and through the GET SAS command, SAS tables are automatically transformed to SPSS system files. PRELIS2 allows input from a very large number of different file formats.

If SPSS or PRELIS2 is not available one possible solution is to export the data from the statistical system to a text file (e. g., in SAS the PUT command may be used). Then the procedures described above for “Rawdata in Text Format” are used. Another possibility is to use a conversion program, which transforms data into SPSS format. One such system is the *Stat/Transfer* program (see *http://www.stattransfer.com*) which can read data in a large number of formats (e. g., Access, Excel, Foxpro, Gauss, Matlab, SAS, Systat, and S-Plus)

The Project Dictionary

Below the nature of the project dictionary is described somewhat more closely.

Project Name

Each project has a project name (*projname*), which is given when the project dictionary is first defined. A *projname* may consist of 1-8 characters. When the project is established by STREAMS it is written into a file with the name "*projname.mdp*". This file should never be deleted, moved or changed in any way. For earlier versions of STREAMS the project dictionary was stored in file called "*projname.dct*". When such a dictionary is opened by STREAMS it is automatically converted into an *.mdp* project and for backup purposes the old dictionary and its associated files is compressed into a file called "*projname.zip*".

Each project dictionary must be kept in its own subdirectory in the filesystem. STREAMS produces a considerable amount of files of different types (see Chapter 11) but these are not explicitly associated with a particular project. Thus, if the same model name is used in different projects which reside in the same directory there will be a conflict. It is also much more convenient to clean up unnecessary files, and to back up projects and their associated files if projects have their own directories.

Variable Labels

Variable labels may consist of 1 to 7 characters (or 1 to 6 characters if two-level modeling is to be used). EQS, LISREL and Mx allow variable labels which are 8 characters long,

but to identify a residual variable STREAMS automatically appends an ampersand (”&”) to variable names. It may be noted that Amos supports longer labels, but in the interest of compatibility of model specifications over all four programs the 7-character limit is maintained.

Almost all tools in STREAMS are case sensitive, so lowercase and uppercase letters are interpreted as different characters. Thus, the labels VAR1 and Var1 refer to different variables. It is therefore essential that labels of variables and datasets are chosen in a consistent manner. It is recommended that uppercase characters are used for observed variables (e. g., GENDER, AGE, WORD).

Folders and Dataset Labels

Rawdata and computed matrices must be uniquely identifiable and easily retrieved from the project dictionary. This is done through assigning a label (a so called *dataset label*) to each of the data sets and matrices included in the dictionary. However, because a project dictionary may include a very large number of group labels, it is also possible to subsume logically related group labels under a *project folder*. Thus, each data set or matrix is identified with two labels, one for the folder and one to identify the dataset within the folder.

There are two kinds of folders: *open folders* and *closed folders*. An open folder (for which the icon  is used) allows the user to add more datasets. A closed folder (which is identified by the icon ) does not allow addition of any more datasets after it has been created. Most folders are of the open type, but folders containing several datasets generated by STREAMS in a single run (e. g., matrices for two-level analysis; separate matrices for each code value; covariance matrices for each missing data pattern) typically are closed. STREAMS decides whether a folder will be open or closed.

There are many ways in which one or more datasets may be organized into folders. Suppose that we want to fit a model for a group of subjects and that we also want to compare models fitted separately for males and females. Three covariance matrices are thus computed: one for the total group of subjects, and separate matrices for the two genders. These matrices may be subsumed under a default folder (which could be labeled *None*) for which three groups are identified with the labels *Total*, *Males* and *Females*. Another possibility would be to use two folders (called *None* and *Gender*, for example). The *None* folder would then have one dataset (*Total*), and the *Gender* folder would have two datasets with labels *Males* and *Females*, respectively. The latter approach is the recommended one, because it is more flexible and well-structured.

If, for example, in a later step we want to do a further analysis of gender differences using the polychoric correlation matrices, these matrices may be brought into the project in a new folder (e. g., *GendPCM*, with dataset labels *Males* and *Females*). We may, of course, also create new sets of matrices for subsets of cases selected according to other criteria, such as social background, or combinations of different criteria, such as gender and social background, and put these in one or more new folders.

There is no limitation to the number of folders or the number of datasets in a project.

Labels for folders may contain 1-8 characters. It is recommended that the first letter is in upper-case and the following in lower-case (e. g., *Gender*, *Trtmnt*). Group labels may also consist of 1-8 characters, and it is recommended that these too have the first letter in upper-case and the following in lower-case (e. g., *Boys*, *Girls*, *Grp1*, *Grp2*).

Preparing Data for Analysis

Before starting the process of bringing data into a STREAMS project, it is essential that some preparatory work is done.

Data Exploration

In a first step the distributional characteristics of the variables should be investigated. Characteristics such as skewness and the presence of outliers should thus be investigated. One useful tool for such data screening is the PRELIS2 program and EQS also offers excellent facilities for description and exploration of data. These programs offer good guides for how to explore data, so there is little reason to repeat that information here.

Missing Data

The problem of missing data has already been discussed at length in Chapter 5 so that discussion will not be repeated here. Suffice it to point out that often the best solution is to apply different missing data treatment methods, such as imputation and list-wise deletion, in combination.

Length of Variable Labels

When data is stored in SPSS and is to be accessed directly by STREAMS it is also necessary to make sure that the length of variable labels keeps within the limits. Thus, for ordinary modeling variable names may not be more than 7 characters long, and for two-level modeling the variable labels may only be 6 characters long.

Polarity of Variables

The estimation programs typically have greater problems finding a solution which includes negative parameter estimates. To prevent unnecessary problems it is, therefore, good practice to reflect variables, such as Likert-type questions which mix positive and negative wordings, in such a way that a high score consistently reflects a high level on a hypothesized latent variable. Another way to put this is that negative elements should be avoided in the covariance matrix.

Homogeneity of Variance

Another frequent cause of problems in model estimation is that variables have different variances. As has already been pointed out in Chapter 7 there are simple techniques which may be used to correct for this.

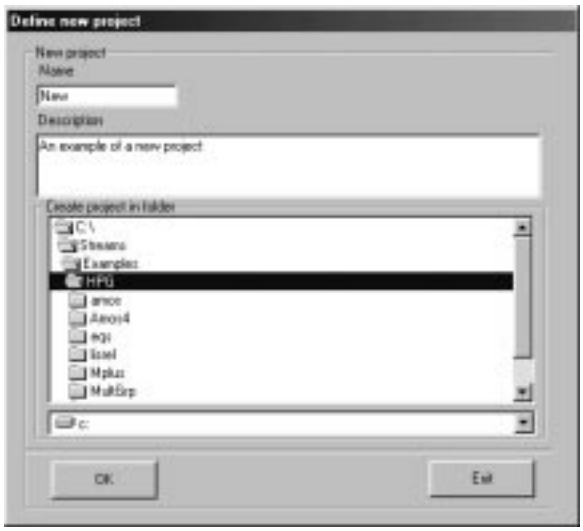
Creating a Project

A new project needs occasionally to be created. It should be observed, however, that new projects should not unnecessarily be introduced, and that it is typically easier to work with a few large projects, rather than with many small projects. A project can keep a large number or datasets, variables, and models, and there is nothing, in principle, which

prevents a user from using one single project for modeling several different sets of data. However, for purposes of overview and manageability it is adviced that project size is kept within reasonable bounds, and that each project is restricted to a particular set of variables.

A new project is created through the function **Create a new project** on the **Data** tab of the *Project window*:

When the button is clicked (or the corresponding menu item under the **Data** menu is selected) a form is presented for entering information about the new project:



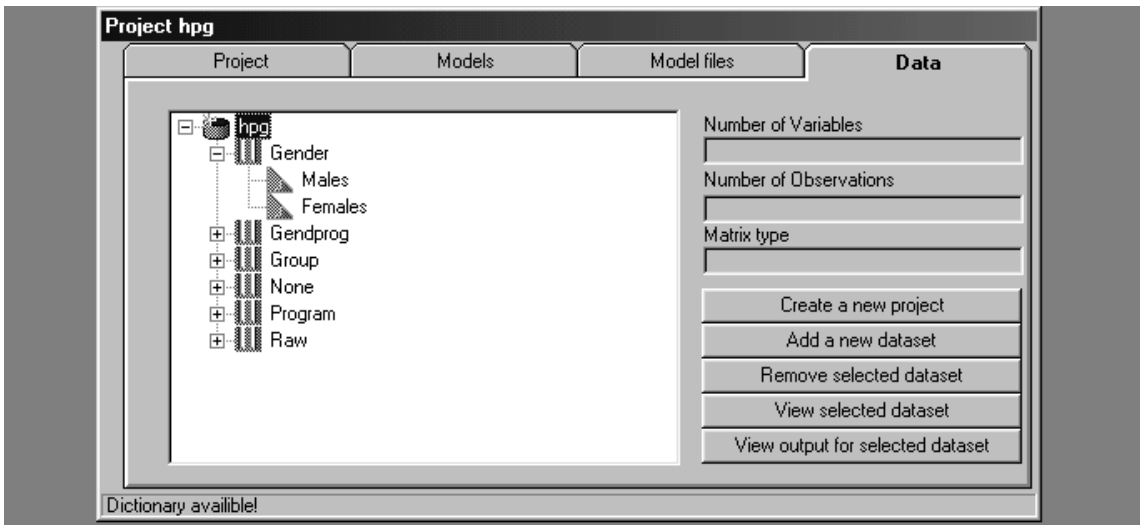
In the **Name** field a name (1-8 characters) of the new project should be entered. Here the name *New* is entered, but users are adviced to use more descriptive names for their own projects. The **Description** field should be used for entering a descriptive text about the project. The new project will be put in a newly created folder which will have the same name as the assigned project name, and which will be put under the selected folder. Here the HPG folder has been selected, so the project will be in a subfolder labeled New under HPG.

When the OK button on the *Define new project* form is clicked, a new, but as yet empty, project is created. After the project has been created the STREAMS *Data Wizard* is automatically started. The *Data Wizard* is used to add data to a project, and if that is not to be done at this point in time, the **Exit** button should be clicked. Use of the *Data Wizard* is described in the next two chapters.

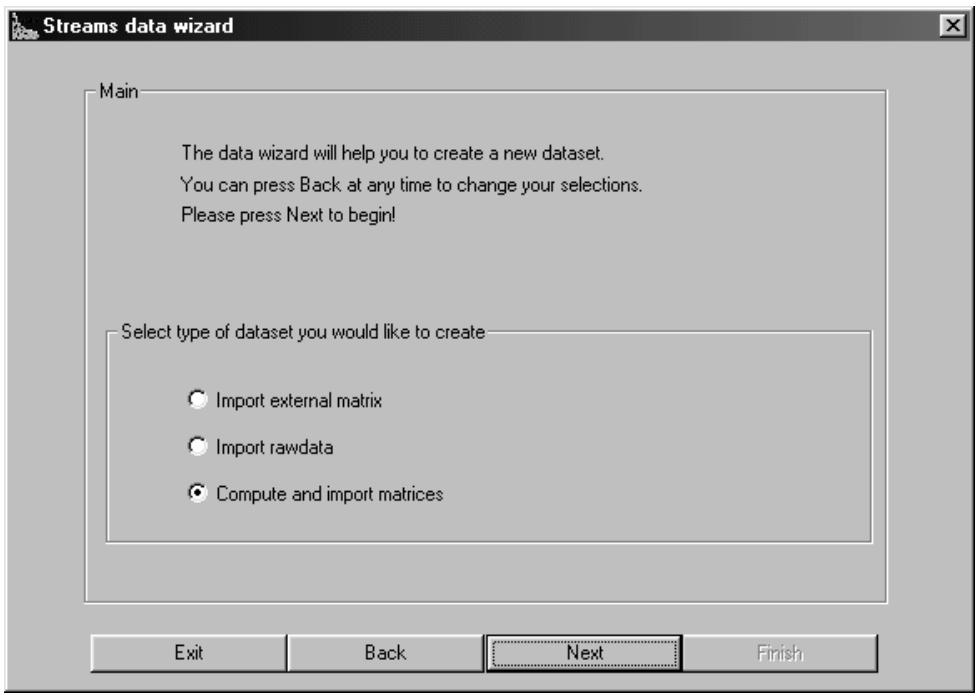
9 Computing Covariance Matrices

It is often advantageous first to compute a covariance matrix from raw data and store it in the project, where STREAMS can access it. In this way the covariance matrix needs to be computed once only, and the additional output associated with the computations is avoided. At other occasions it is necessary to input raw data into the estimation programs, for which STREAMS also offers support.

Addition of datasets to a project is done with the STREAMS *Data Wizard*. The *Wizard* guides the user through the steps necessary for specifying different types of data imports, performs the computations, and adds the new dataset(s) to the project. To start the STREAMS *Data Wizard* the button **Add a new dataset** on the **Data** tab of the *Project window* is clicked. For the example that we will follow it will be assumed that a dataset will be added to the HPG project, so this project should be open when the button is clicked:



This causes the first screen of the *Data Wizard* to be presented:

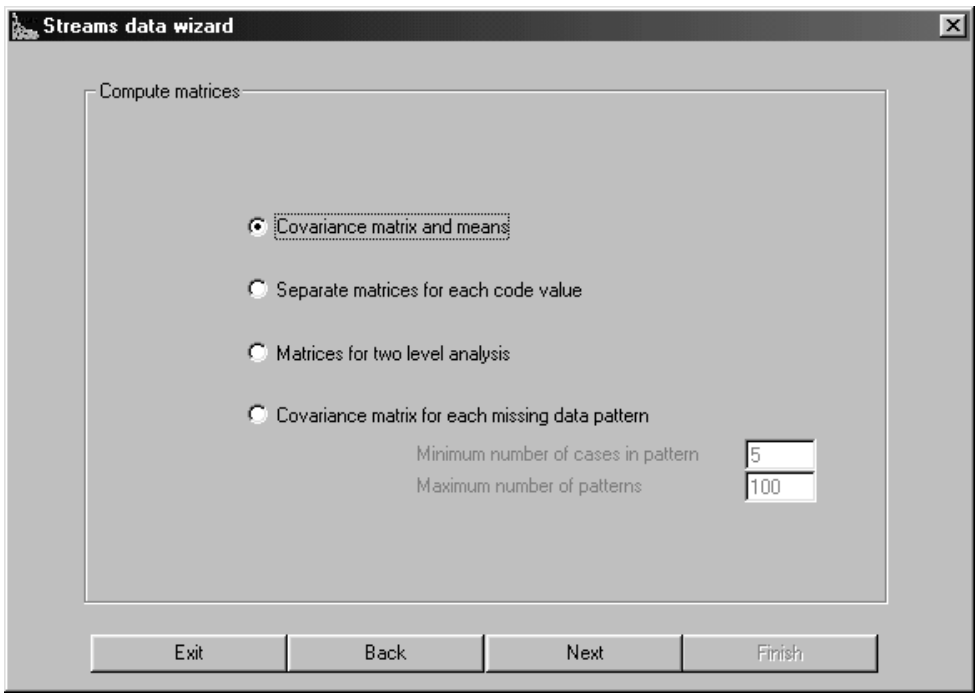


In this step a choice is to be made of what kind of dataset is to be added to the project: an externally computed matrix, raw data, or one or more covariance matrices computed from raw data. The procedures for importing raw data and external matrices are described in greater detail in Chapter 10, while the present chapter describes how to compute matrices from raw data.

First we describe in some detail all the steps involved in computing an ordinary covariance, and after that the other cases are described in less detail.

Here we want to compute a matrix, which is also the default, so we click **Next.**

In this step the *Data Wizard* wants information about what kind of computation is requested, among four different choices:



We will first of all deal with the case when a covariance matrix is to be computed for either all cases or for a subset of cases. At this step we thus select the option **Covariance matrix and means**.

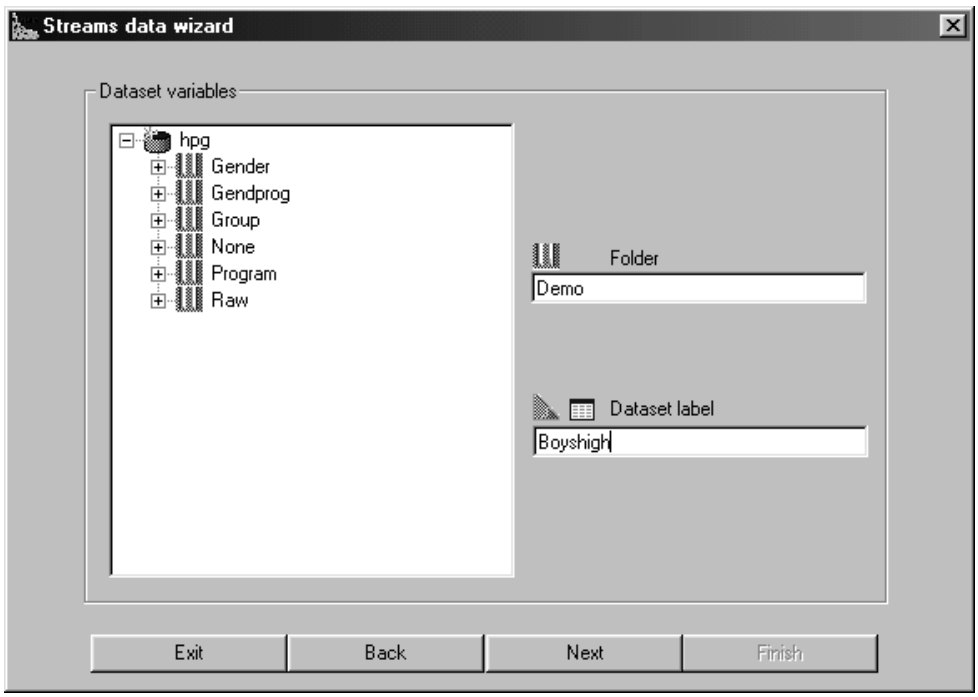
Computing a Covariance Matrix

When the **Next** button is clicked a form for labling the dataset is presented.

Labling the Dataset

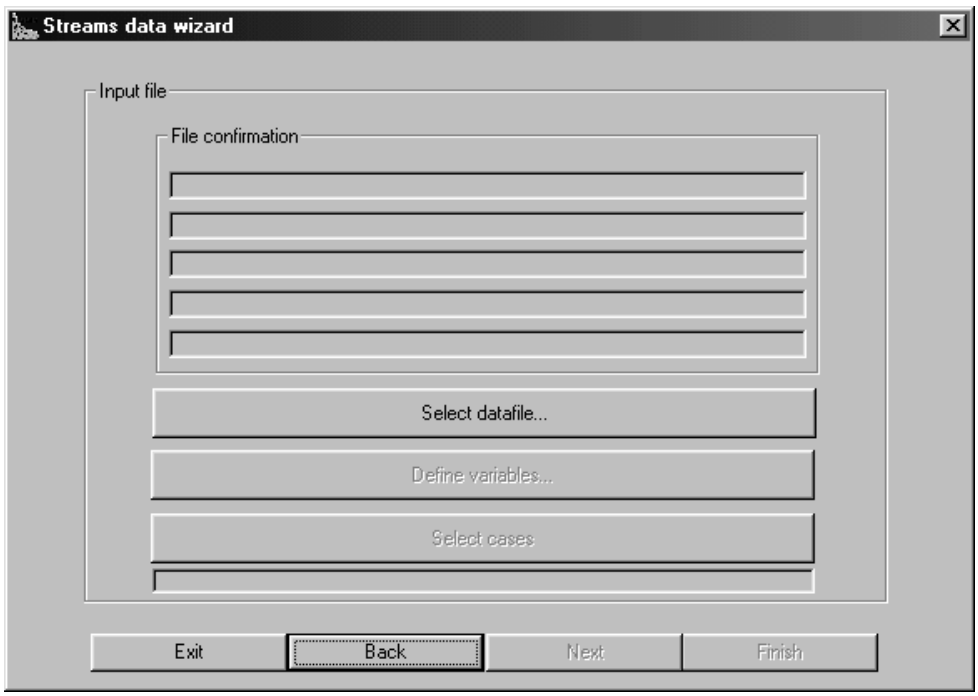
The labling of the dataset is done through assigning one "Folder" label and one "Dataset label". The folders are the top-nodes in the two-level hierarchical system for storing datasets in STREAMS, while the dataset labels are the bottom-level units. The choice of both the name of the folder and the dataset label is completely in the hands of the user, and any label (up to 8 characters) may be assigned. Clicking the icon for an existing folder will present the name in the field for the folder name. This label may be kept, or it may be edited to create a new top-node in the hierarchical system. When the icon for an existing dataset label is clicked its name will also be copied to the field for entering the dataset label. It must be observed, however, that STREAMS will not allow the user to assign the same combination of folder name and dataset label as for an existing data set. If the existing data set is to be replaced with a new one, the old one must first be deleted.

For purposes of demonstration we will compute a covariance matrix for a subsample of boys who have a score of 15 or higher on the DTM test. A suitable new folder may be *Demo*, and we assign *Boyshigh* as the dataset label. After this information has been entered the form thus looks as follows:



Input File

In the next step the data input file is specified:

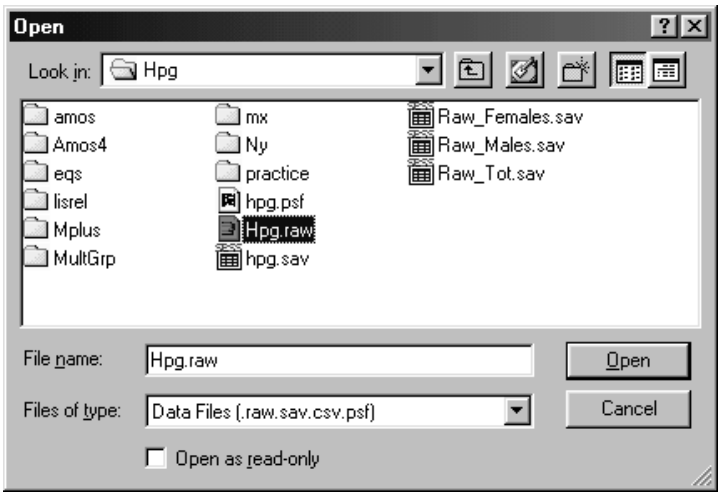


STREAMS reads SPSS files (suffix *.sav*), comma-separated files (suffix *.csv*), PRELIS2 files (suffix *.psf*) as well asci-files (suffix *.raw*) equipped with a special data dictionary (see "The STREAMS Rawdata Format", page 235).

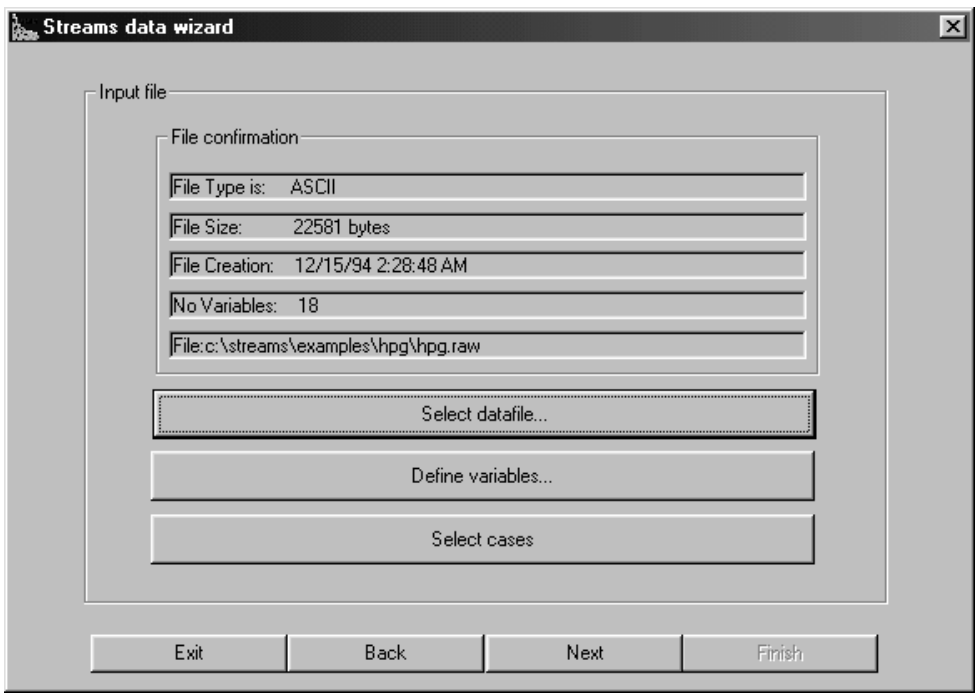
When a *.raw* file is opened the **Define variables...** button is enabled. To add descriptions

of variables in a rawdata file, or to update previously given information, the **Define Variables** button may be clicked, and information entered in ways described in greater detail in Appendix A (page 235).

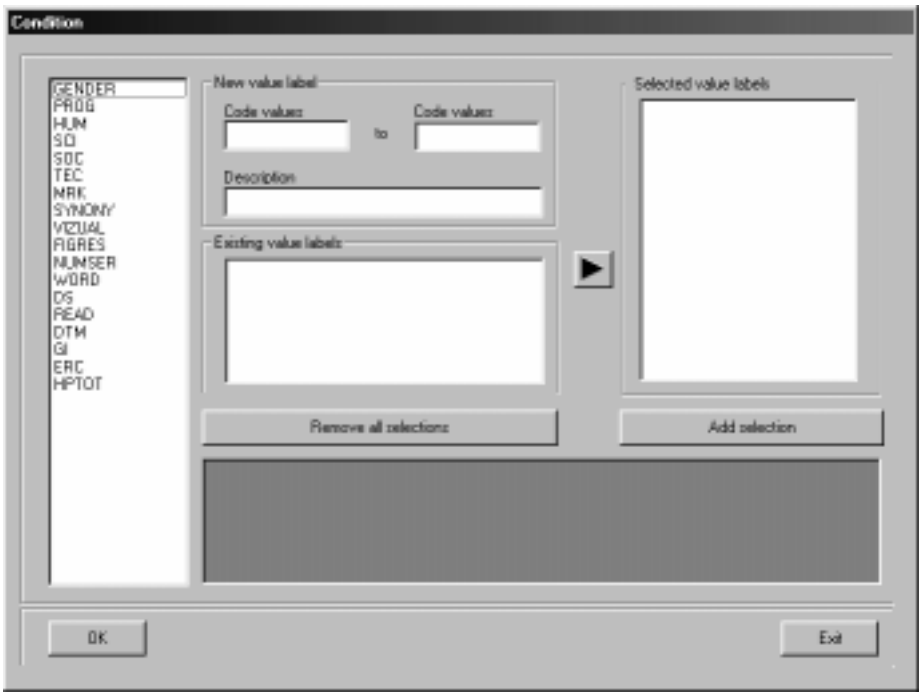
To specify the file the **Select datafile** button is clicked, which presents a standard file open dialogue. The file type must be specified with the drop-down menu labeled **Files of type:**. Here the file *hpg.raw* in the *HPG* directory is selected:



STREAMS reads information about the file and the variables and presents some descriptive information about the file:

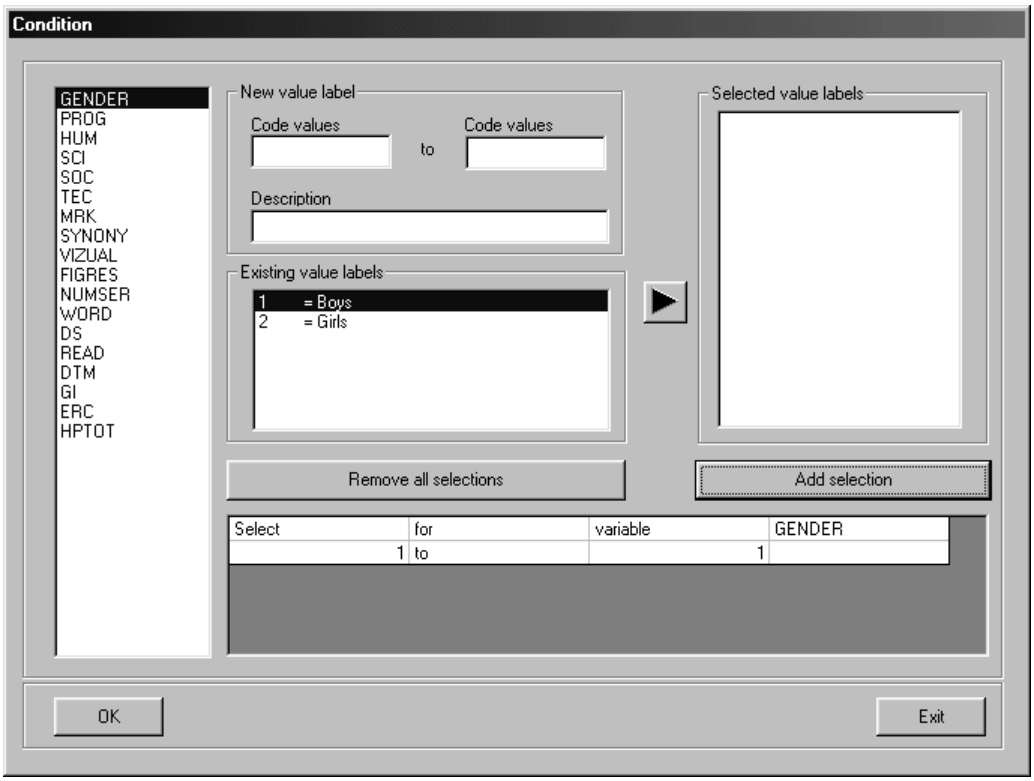


At this step it may also be specified which subset of cases should be included in the computations. It will be assumed that boys with a score higher than 15 on the DTM test are to be included. To specify rules for selection the **Select cases ...** button is clicked, which presents the *Condition* form:

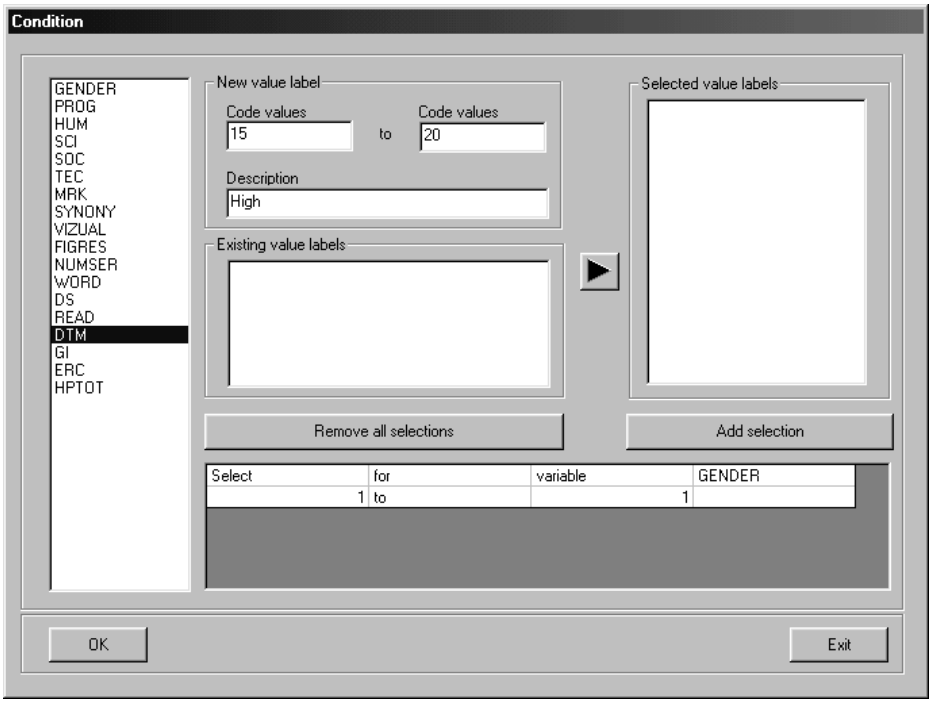


On this form one of the variables in the file may be selected at a time, and one or more code values may be selected for inclusion. To be included in the computations an individual must fulfill the inclusion criteria for all specified variables.

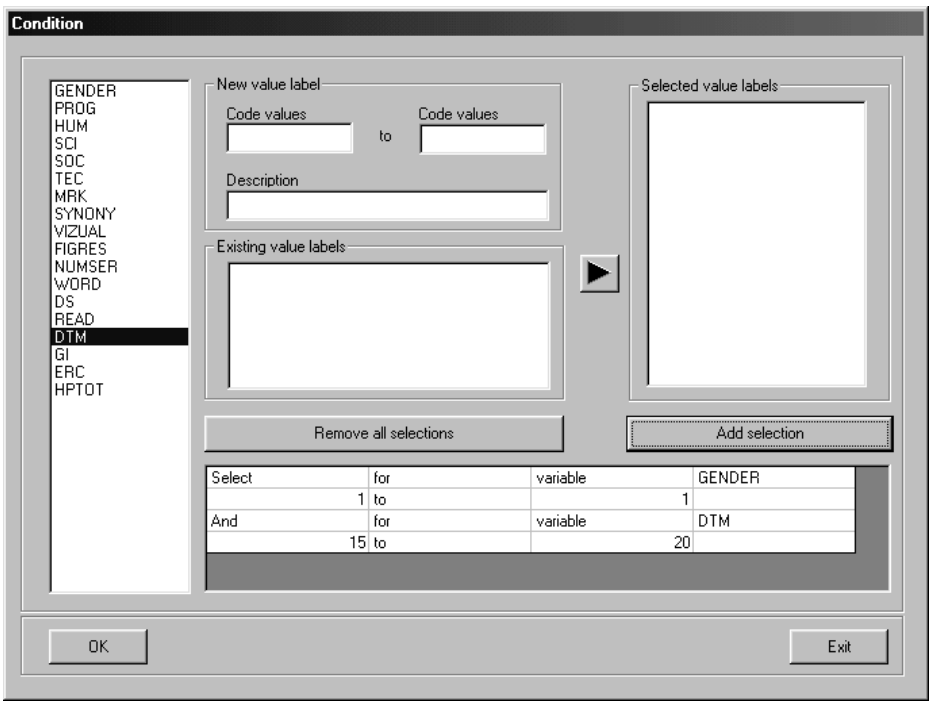
If the GENDER variable is clicked any existing code values and value labels are shown in the **Existing value labels** list-box. Here two value labels have previously been defined, *boys* and *girls*. To select boys this value label is selected and then the arrow is clicked to move the selection to the list-box labeled **Selected value labels**. Should there be other code values to be included for a variable they may also be added to the list of selected value labels. After all the code values for a variable have been selected the **Add selection** button is clicked, which transfers the selection or selections to the grid at the bottom of the form. Thus after *boys* have been selected the form looks as follows:



However, in this hypothetical example we also want to impose the restriction that only boys with a score of 15 or higher on the DTM test are to be included. If the DTM variable is selected it is seen that for this variable no value labels have been defined, so we cannot use the same procedure as before. Instead we introduce a new (temporary) value label, which comprises the scores 15 to 20 (which is the highest possible score on the DTM subtest). This is done through entering the score interval and a label (e. g., High) in the frame labeled **New value label** as shown below:



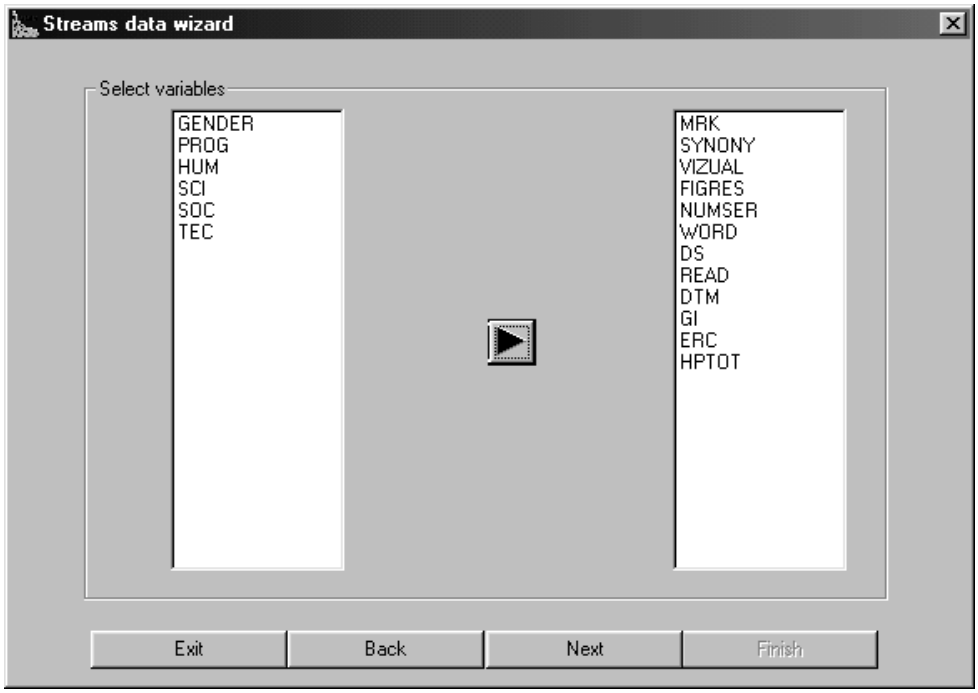
Next the defined range of scores is transfered to the **Selected value labels** list and then the **Add selection** button is clicked. This causes the *Condition* form to take on the following appearance:



If the **OK** button is clicked next, the covariance matrix will be computed only for boys with scores between 15 and 20 on DTM.

Select Variables

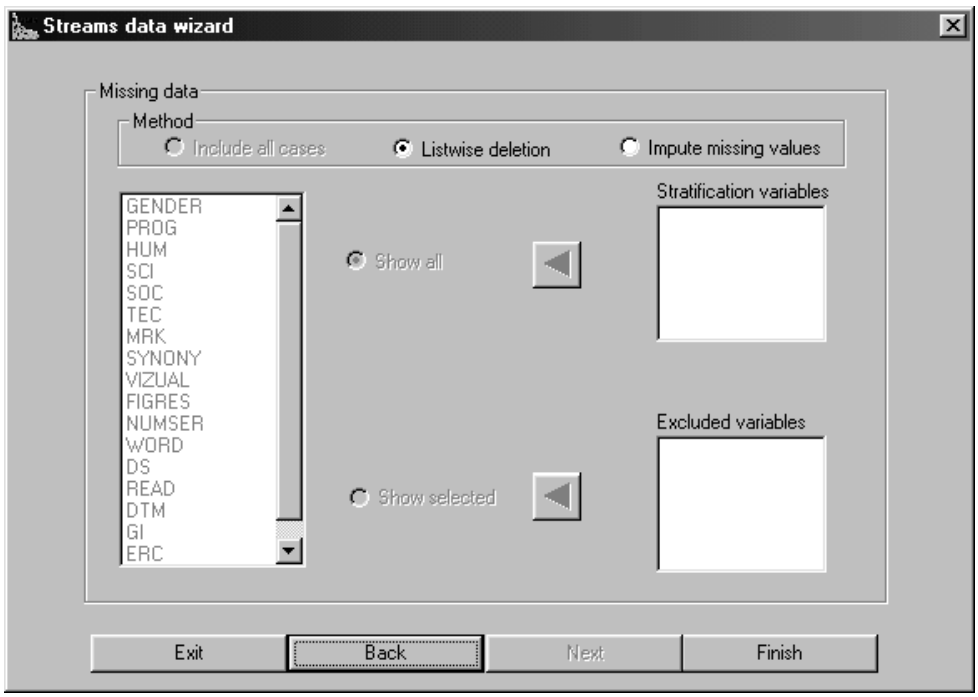
Whe **Next** button on the *Wizard* form is clicked a form for selecting the variables in the matrix is presented:



All the available variables are presented in the list-box to the left, and those to be included in the computed matrix are to be moved to the list-box to the right.

Missing Data

In the last step it is specified how missing data should be dealt with:



This form offers three main methods for dealing with missing data: **Include all cases**, **Listwise deletion** and **Impute missing values**.

The **Include all cases** option implies that all variable values are treated as valid data, and are included in the computations. This option is not available when computing covariance matrices, but it is useful when data are imported for further analysis with the missing data procedures offered by Amos, Mplus and Mx.

The **Listwise deletion** option implies that a case with missing data on one or more variables will be excluded from computations. It should be observed, however, that when there are many variables this option may cause an unacceptably large proportion of the cases to be excluded.

The **Impute missing values** option implies that the missing data codes are replaced with estimates of data values (so called "imputation") when computing covariance matrices. Within STREAMS, missing data may thus be replaced with the mean for the total group, or with the mean of a subgroup of cases identified through one or more *stratification* variables. It should be observed, however, that this option should be used with care, because the procedure will cause systematic disturbance to the covariance matrix (e. g., the variances will be underestimated) when more than a limited number of values is replaced.

Selection of the **Impute missing values** option causes further options to become available. The list-box labeled **Stratification Variables** thus is highlighted, as is the list-box labeled **Excluded variables**.

One or more (up to five) variables may be moved into the **Stratification Variables** list-box. These variables are used to classify the sample into subsets according to combinations of values on the variables, and the substitution of missing values is then based on the mean for the subset to which an individual belongs. When no stratification variable is selected the grand mean is used to replace missing values. Conditioning upon stratification variables causes the replacement of missing observations to be more precise, to the

extent that there is a relationship between the stratification variables and the variables in which missing data are replaced. However, when stratification variables are used, there is a greater risk that the imputation will fail, because it may not have been possible to compute a mean for the particular subset to which the individual belongs.

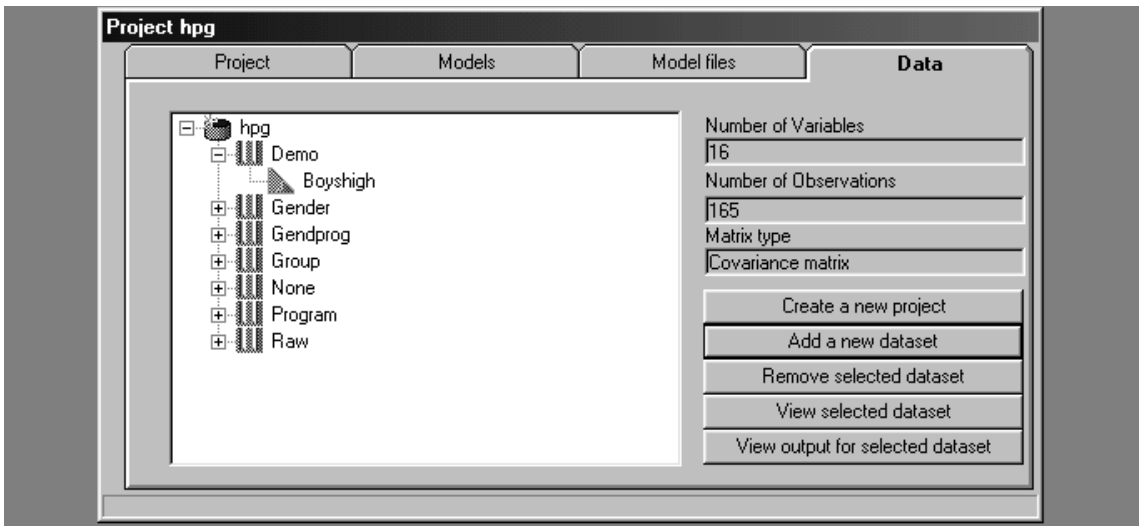
Variables to be used for purposes of stratfication are not so likely to be among those selected to be included in the matrix. When the option labeled **Show All** is selected all the variables included in the data file is therefore displayed.

The **Exclude Variables** list-box is used to exclude single variables from imputation. Variables for which no imputation is to be made should be moved into this list-box.

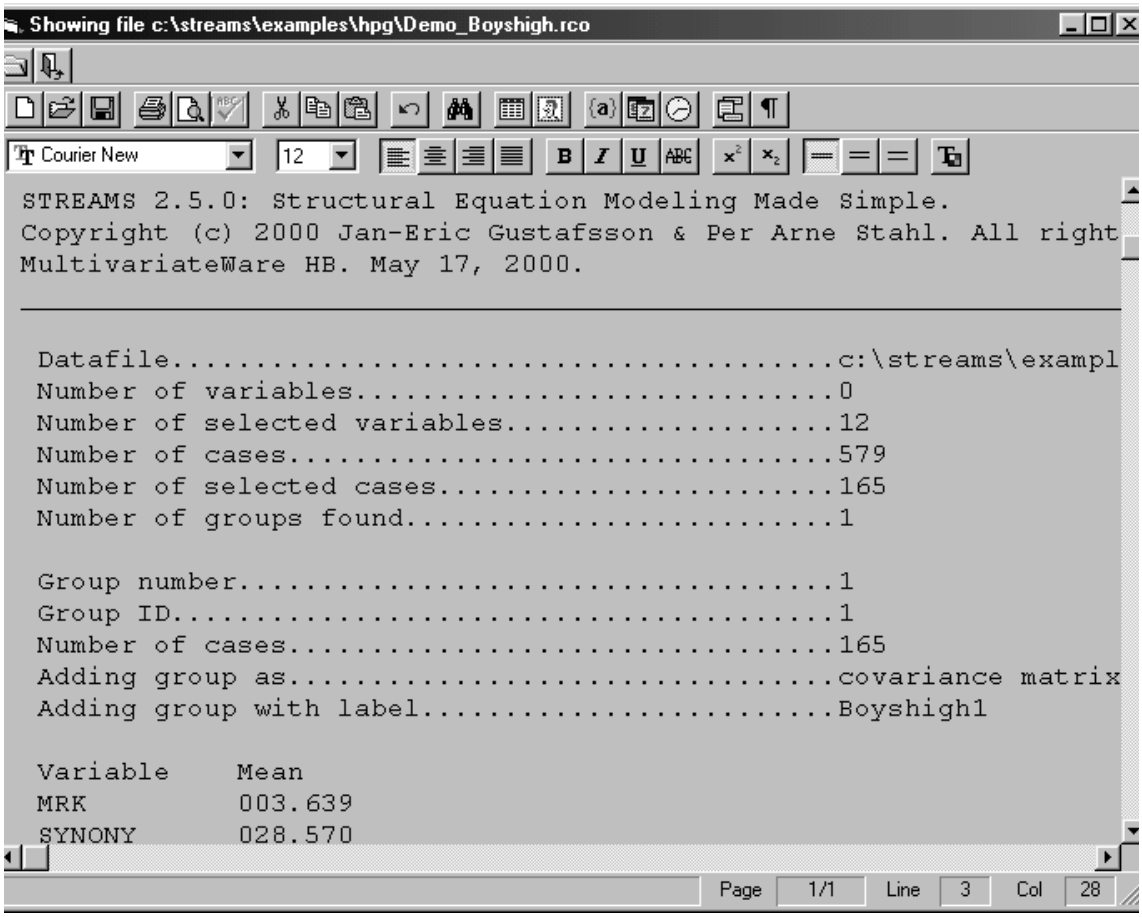
In this case there are no missing data so the option **Listwise deletion** may be used.

Finish

When the **Finish** button is clicked the program performs the computations and when these are completed the updated *Project window* is presented:



A covariance matrix with 12 variables has been computed for 165 cases. If we want to inspect the matrix this can be done with the procedures previously described in this chapter. We also may click the button **View output for selected dataset**, which presents a listing file concerning the computations and some other output such as means:



This file may be closed through clicking the close button, and then modeling may begin. To start modeling, the *Project window* button is first clicked, and then the **Models** tab is selected. The button **New Model** may then be clicked, after which model specification may start, using the procedures described in Chapter 3.

Computing Separate Matrices for each Code Value

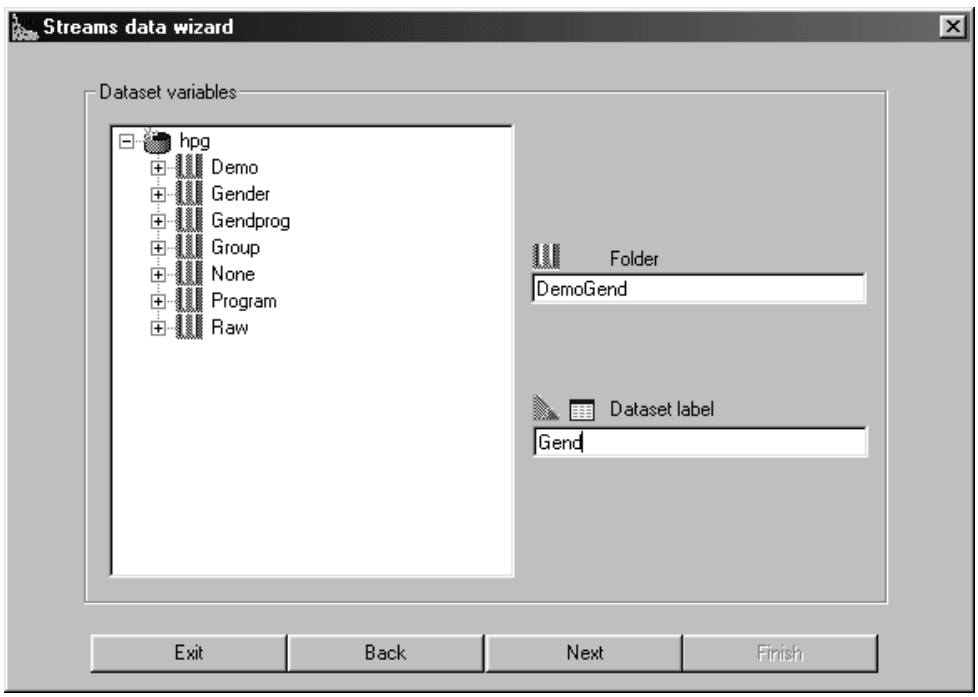
It will be remembered that after the *Data Wizard* has been started and the option **Compute and import matrices** has been selected four options are presented, one of which is **Separate matrices for each code value.** This procedure is highly similar to the one described above, except that multiple matrices are computed and imported. STREAMS thus computes a separate covariance matrix and mean vector for each code value of a classification variable. Thus, if a variable has 300 different code values, 300 covariance matrices will be computed if this variable is used as classification variable. The classification variable must only contain integer values (i. e., alphabetical characters or decimal numbers are not allowed)

We will demonstrate how to use this procedure, assuming that separate matrices will be computed for boys and girls in the HPG data. When the **Next** button is clicked the form for labling the dataset is presented.

Labling the Dataset

The labling of the resulting datasets is done very much in the same way as when a single matrix is computed, except that the Dataset label supplied is used only as a prefix. Each dataset thus is assigned a label which is a combination of this prefix and the code value of the classification variable. The Dataset label must thus be short enough to leave room for this code value. It is strongly recommended that a new project folder is introduced for each new set of matrices that is computed. This keeps the datasets well organized, and because all the datasets in a folder may be removed in a single operation this also makes it easy to maintain the dictionary. After the matrices have been computed the folder will be a closed folder. It should be observed that the groups are introduced in the dictionary in the order they appear in the data. Thus, if a particular order among the groups is required, the data should be sorted in ascending or descending order on the classification variable before the computations.

For our example we will label a new folder *DemoGend*, and the dataset label is *Gend*. This will cause the matrix for boys to be labeled *Gend1* and the matrix for girls to be labeled *Gend2*, because boys have the code value 1 and girls the value 2 on the variable GENDER. After this information has been entered the form thus looks as follows:

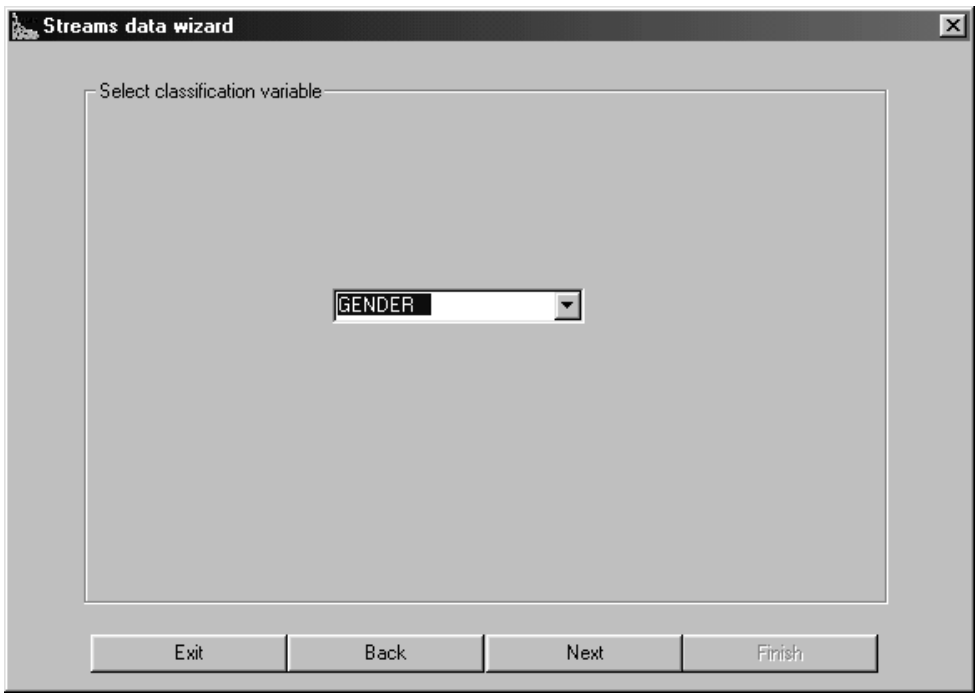


Input File

In the next step the data input file is specified, which is done in the same way as when a single matrix is computed. Selection of a subset of cases may also be specified in the same way as was described above.

Select Classification Variable

When the **Next** button is clicked a form for selecting the classification variable is presented:



The drop down menu is used to identify which variable in the dataset is to be used as classification variable. It must be remembered that only integer values are allowed in the classification variable (i. e., decimal numbers or alphabetical characters are not allowed).

Select Variables

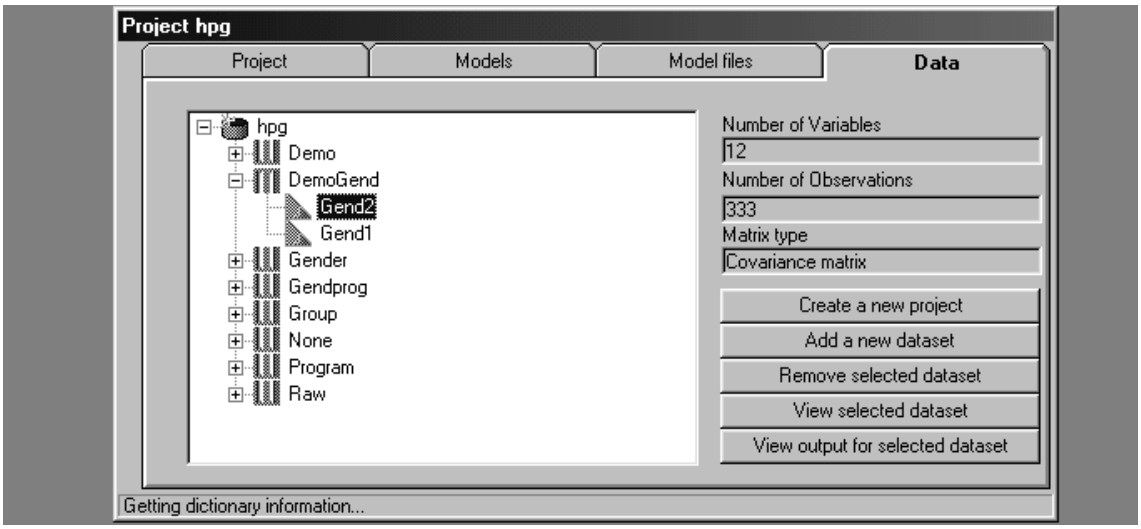
Next the variables to be included in the matrix are selected, which is done in the same way as when a single matrix is computed.

Missing Data

Specification of how missing data should be dealt with is also done in the same way as presented above.

Finish

When the **Finish** button is clicked the program performs the computations and when these are completed the updated *Project window* is presented:



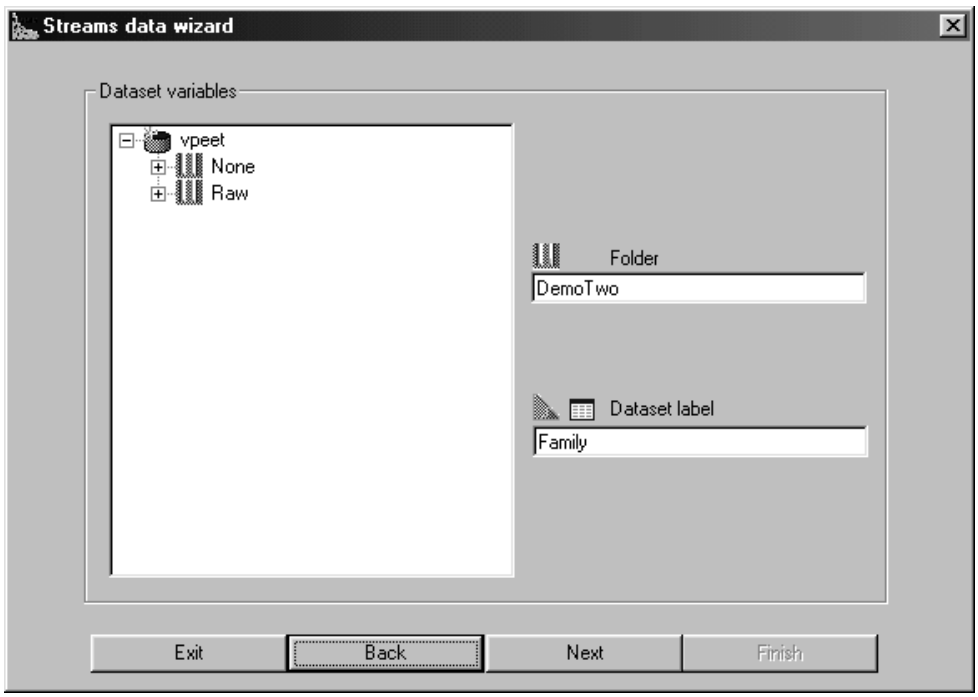
It may be observed that the folder *DemoGend* is a closed folder, which implies that no more dataset may be added to this folder. This is because the folder has been created with option "Separate matrices for each code value" which makes it natural to treat all the matrices in the folder as a unit.

Computing Matrices for Two-Level Analysis

The option **Matrices for two level analysis** prepares two matrices, a pooled-within and a between-group matrix, which are used in two-level modeling (see Chapter 6 for a description of two-level modeling). Except for some minor differences this procedure is identical to the one used in computing separate matrices for each code value of a variable.

Labling the Matrices

When the **Next** button is clicked the form for labling the datasets is displayed. Assignment of the folder name and the dataset labels is done in the same way as when ordinary matrices are computed. However, when matrices for two-level analysis have been requested, two matrices are computed, one pooled-within covariance matrix and one between-group covariance matrix. To separate these matrices, the program uses the supplied dataset label as a prefix, the suffix W for the pooled-within matrix, and the suffix B for the between-group matrix. Here the dataset label *Family* is assigned, which implies that the two matrices will be labeled *FamilyW* and *FamilyB*. For folder the label *DemoTwo* is assigned:



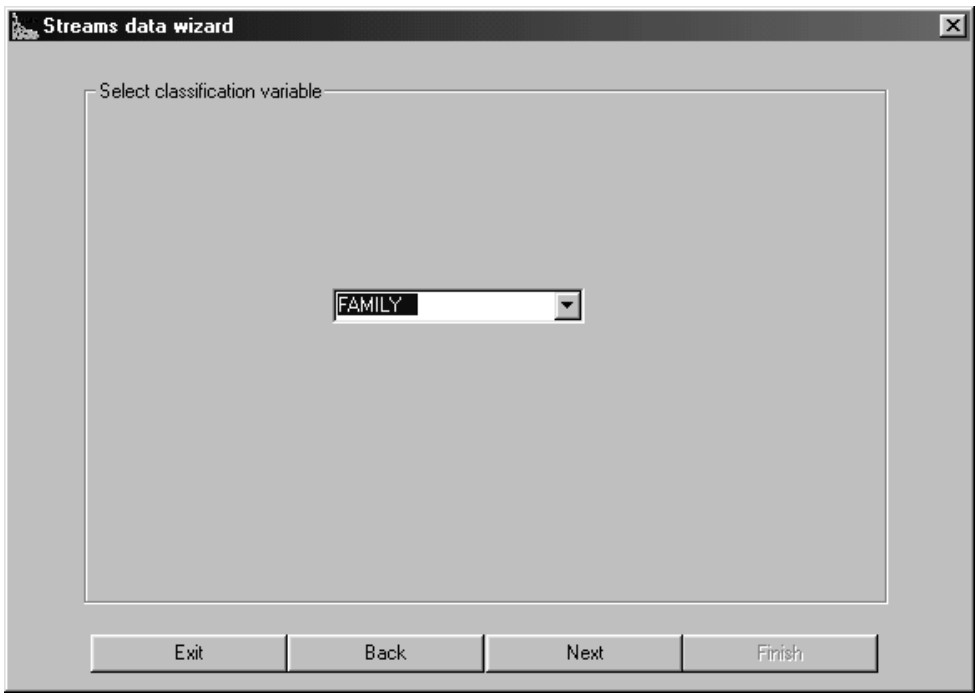
Input File

Next the input file is selected, which is done in the same way as in the cases described above. Observe that all the data must reside in one file, which has as many rows as the number of individual observations. All the group level variables must be copied down to the individual level, and must be repeated for all individuals belonging to the same group.

In the present example the file *vpeet.raw* is selected.

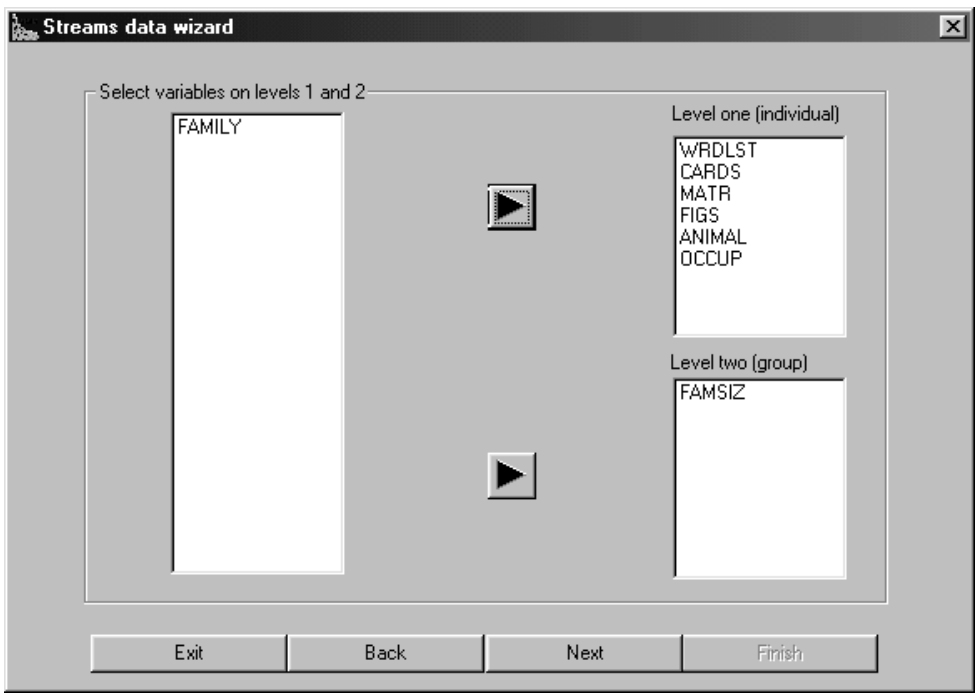
Classification Variable

On the next screen the drop-down box labeled **Classification Variable** is shown. This is used to identify the observed variable which contains information about group membership. The classification variable must only contain integer values (i. e., alphabetical characters or decimal numbers are not allowed). In the present example the FAMILY variable is selected:



Select Variables

Next the variables to be included are to be specified. When matrices are computed for two level analysis it is necessary to specify both which individual- and which group-level variables are to be included. All available variables are shown in the left-most list-box, and those variables which are to be included as individual (level one) variables are moved to the top-most list-box on the right, while those variables which are measured at the group level are moved to the bottom list-box on the right:

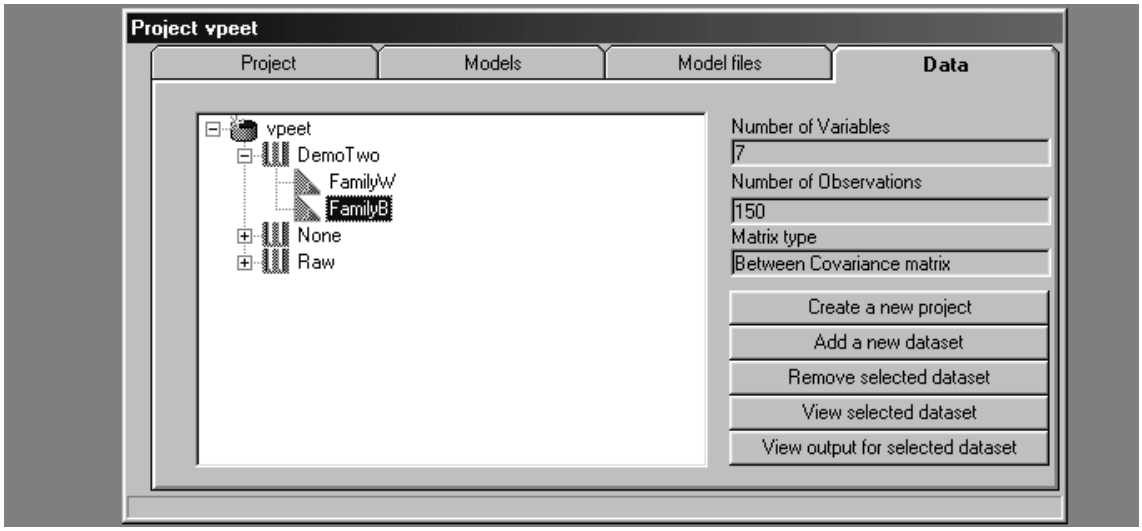


Missing Data

Specification of how missing data should be dealt with is also done in the same way as presented above.

Finish

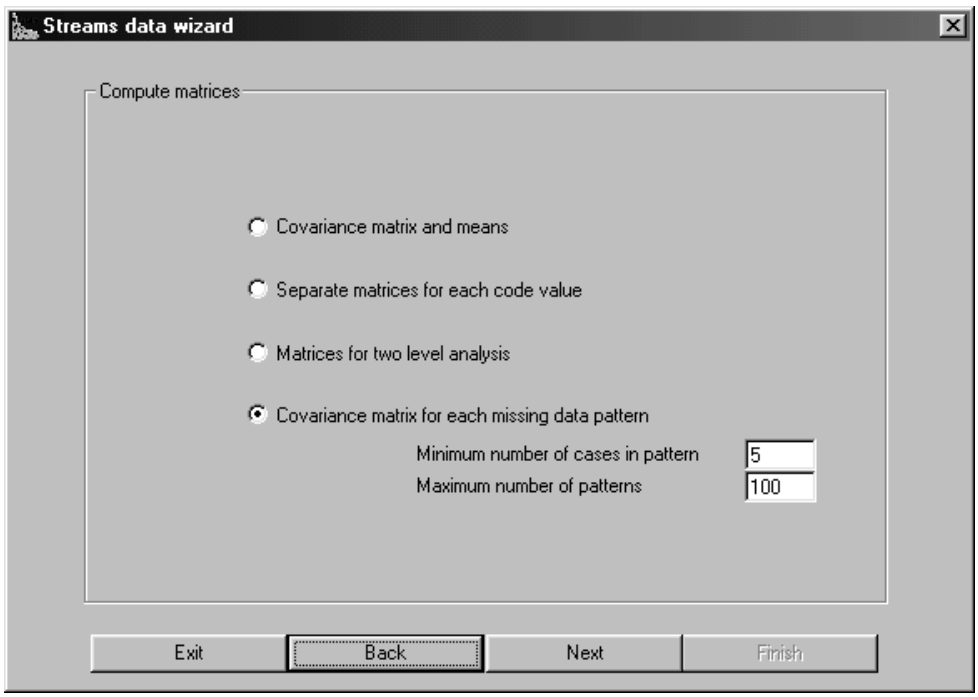
When the **Finish** button is clicked the program performs the computations and when these are completed the updated *Project window* is presented:



Computing Covariance Matrices for Each Missing Data Pattern

Yet another choice among the four types of computations is to generate one **Covariance matrix for each missing data pattern.** When incomplete data is modeled with the procedures described in Chapter 5, input of a separate covariance matrice for each subset of cases with a particular combination of missing data is required. For any particular set of variables and cases, STREAMS can sort the cases into the different combinations and produce a covariance matrix and mean vector for each such group which includes a certain minimum number of cases. It must be observed, however, that when the number of variables is large and there is much accidental missingness, a very large number of missing-data patterns will result, and many of them will be represented by one case only. It is, therefore, strongly recommended that most of the accidental missingness is first taken care of through imputation of missing values, so that a limited number of missing-data patterns remain.

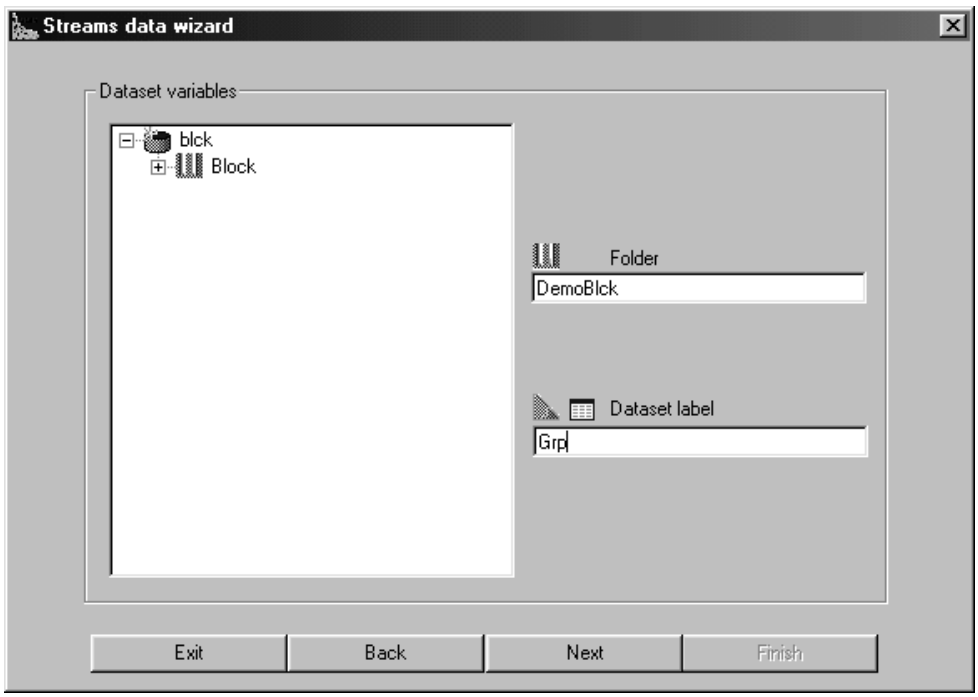
We will illustrate this procedure by an example in which matrices are computed for a matrix-sampling design in which different subsets of a sample have been given 3 blocks of reading tasks out of a total of 7 blocks. The data are in the BLCK project, so this project should first of all be opened. After the option **Compute and import matrices** has been selected the *Compute matrices* form is presented:



When the option **Covariance matrix for each missing data pattern** has been selected a minimum number of cases may be selected for each group, which is done in the field labeled **Minimum number of cases in each pattern**. Only patterns which include at least this number of cases (with a default value of 5) are retained. It is also possible to specify a maximum number of missing-data patterns, which is done in the field labeled **Maximum number of patterns**. When the number specified here (the default value is 100) is exceeded during the reading and classification of data, program execution is halted, and the project dictionary is left unchanged. It is then necessary to rerun the program, either with a higher value set for this parameter, or with some missing values replaced with imputed values.

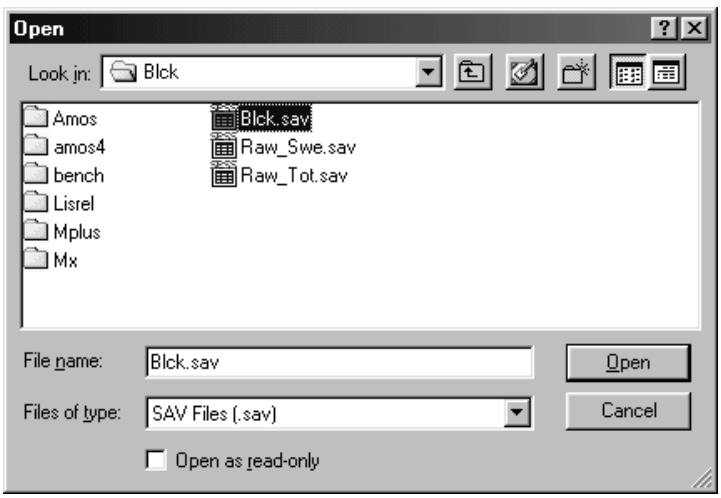
Labling the Dataset

It is strongly recommended that a new folder is assigned whenever this option is used. A very large number of matrices may be produced, and because all the matrices in a certain folder may be deleted in one step this makes it easier to maintain the project. Here the folder *DemoBlck* is assigned. For the Dataset label a prefix is assigned. Each matrix is assigned a label which is a combination of this prefix and the ordinal number of the missing-data pattern. The Dataset label must thus be kept sufficiently short so that there is room to affix the ordinal number. Here the Dataset label *Grp* is assigned.



Input File

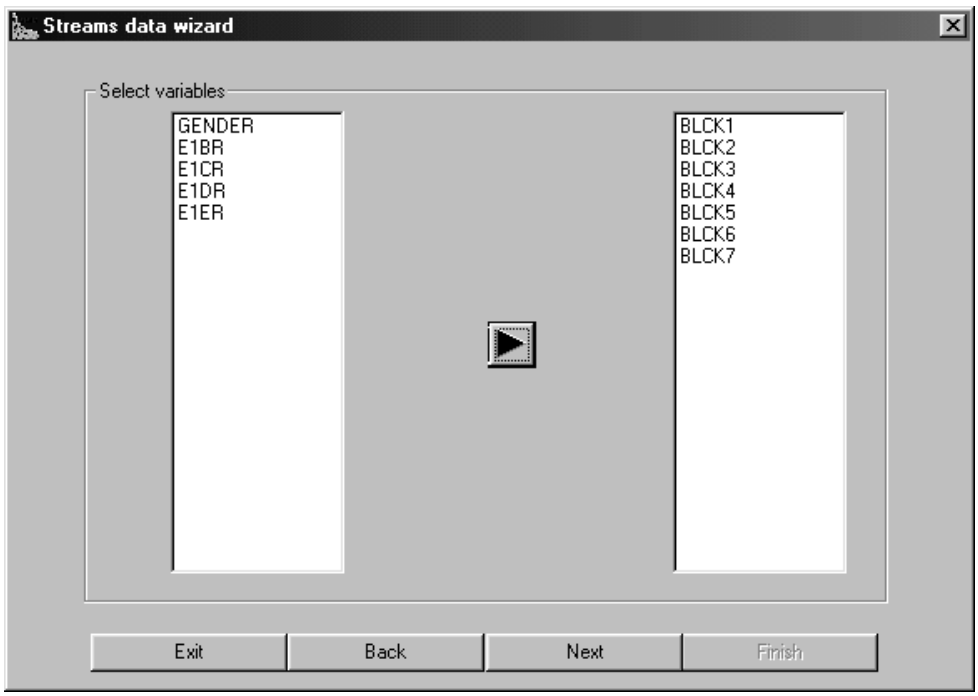
In the next step the data input file is specified, which is done in the same way as when a single matrix is computed. Here an SPSS-file (*Blck.sav*) is to be read, so the file type must be changed:



Selection of a subset of cases may also be specified in the same way as was described above.

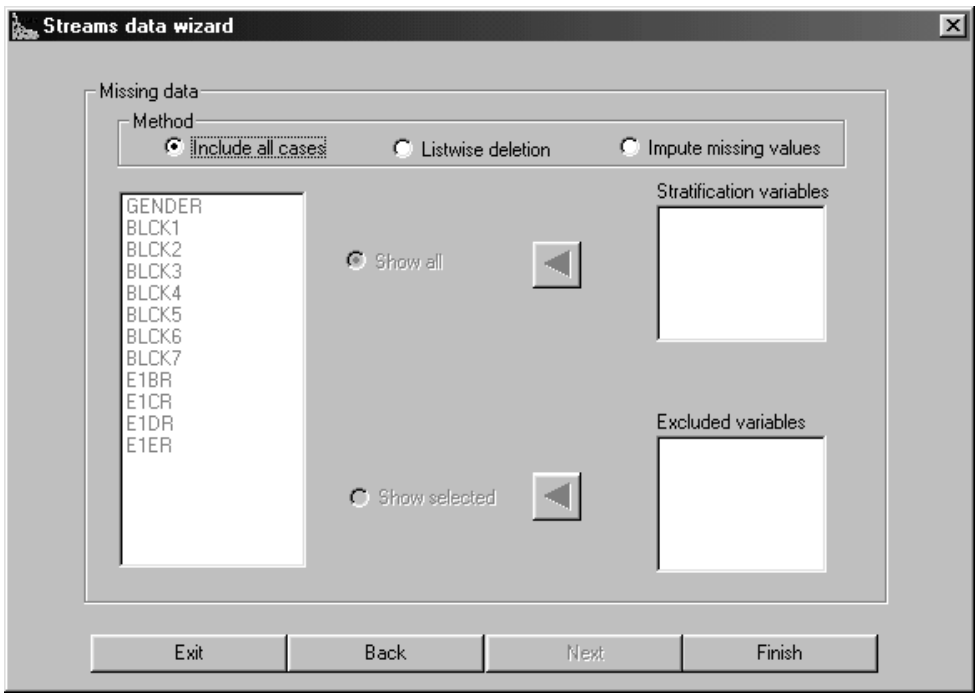
Select Variables

Next the variables to be included in the matrix are selected, which is done in the same way as when a single matrix is computed. All the variables to be included in the computations are specified, and STREAMS keeps track of which variables are present in which group:



Missing Data

On the *Missing data* form the **Include all cases** option is selected:



Finish

When the **Finish** button is clicked the program performs the computations and when these are completed the updated *Project window* is presented.

10 Importing Raw Data and Matrices

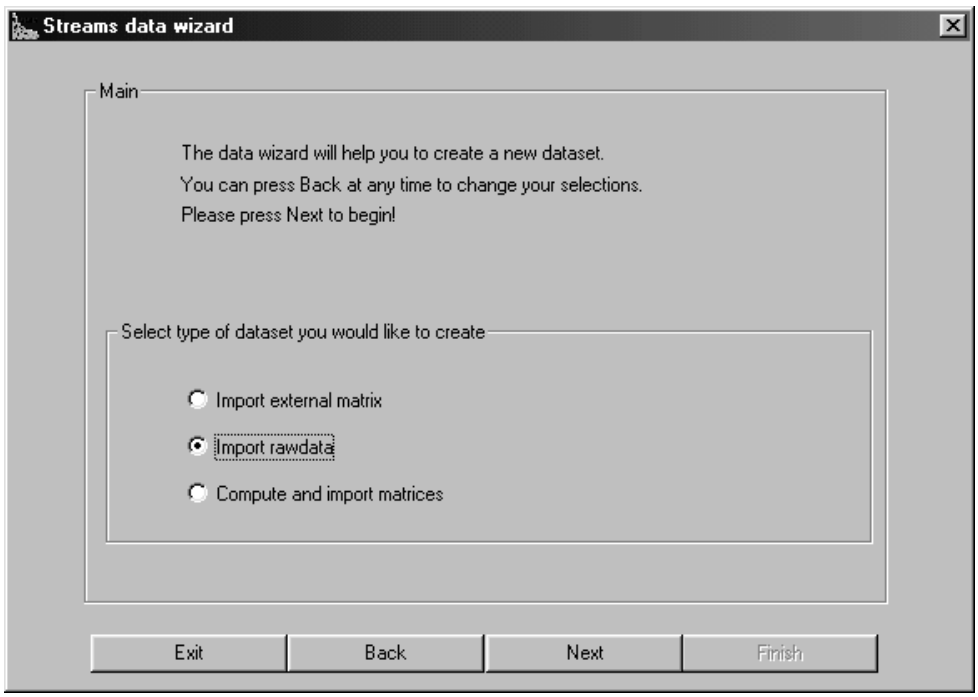
There are several situations when STREAMS needs other data than covariance matrices computed from raw data, such as:

- Raw data, to be input to an algoritm which requires individual observations (e. g., the robust estimation procedures in EQS and Mplus, or the missing-data estimation algoritm in Mplus).
- A covariance matrix with a weight matrix computed by PRELIS to be used in WLS estimation.
- A matrix of polychoric correlations among ordinal level variables computed by PRELIS which also are to be used in WLS estimation.
- A previously computed matrix (e. g., a published covariance matrix) for which raw data is not available.

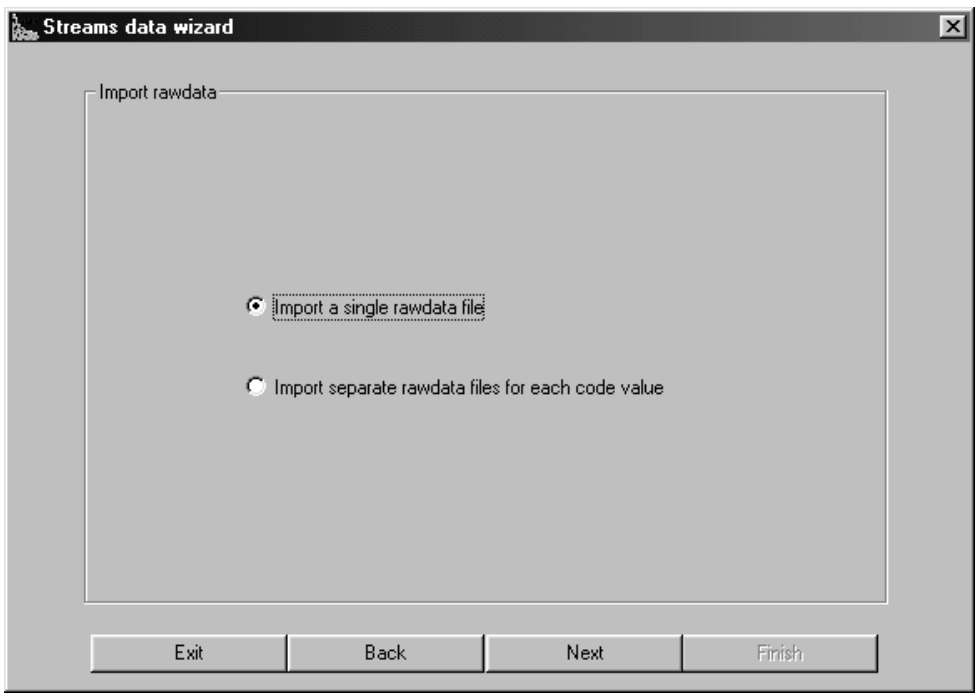
This chapter describes how to import such data into a STREAMS project.

Importing Raw Data

To specify that raw data is to be imported the option **Import rawdata** on the main form of the STREAMS *Data Wizard* is specified:



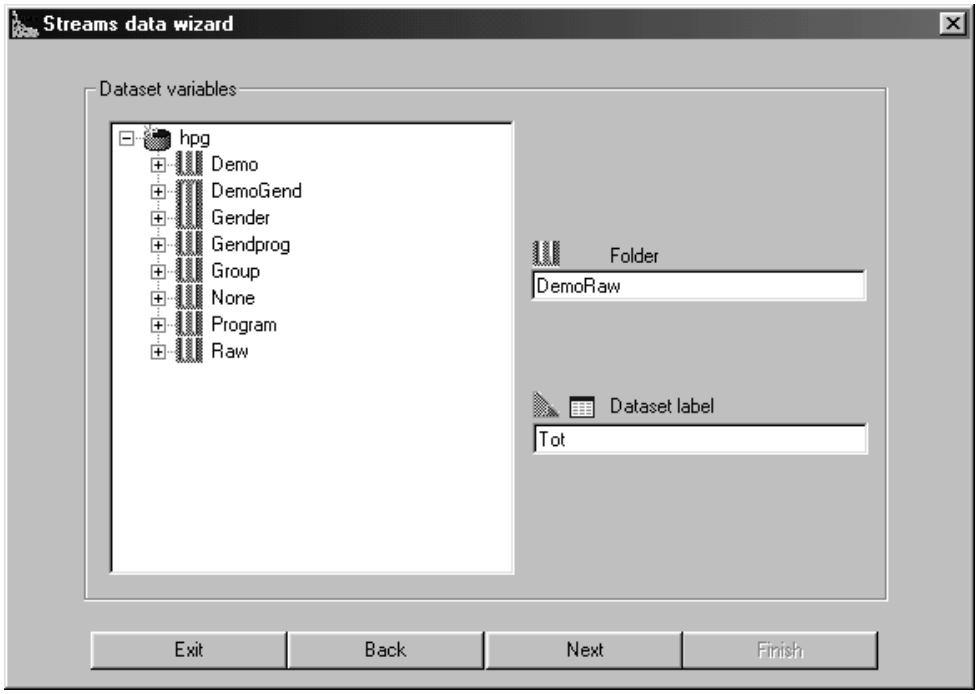
The next form presents two choices: either to import a single file, or to import one subset of data for each code value of a classification variable:



These two options are highly similar, and the only difference is that when separate rawdata files are imported a classification variable is asked for (see the procedure for computing one covariance matrix for each code value of a classification variable in Chapter 9). We will, therefore, only illustrate how to import a single rawdata file. It is assumed that rawdata is to be imported into the HPG project, so this project needs to be opened first of all.

Labling the Dataset

Next the form for labling the imported data set is presented:



Here the label *DemoRaw* is assigned as the categorization variable, and *Tot* as the Group label.

Input File

Next the input file is identified, which is done in the same way as is described in the section “Input File” on page 144 in Chapter 9. Should only a subset of cases be imported, conditions for selection may also be specified.

Here the *Hpg.raw* file is opened.

Select Variables

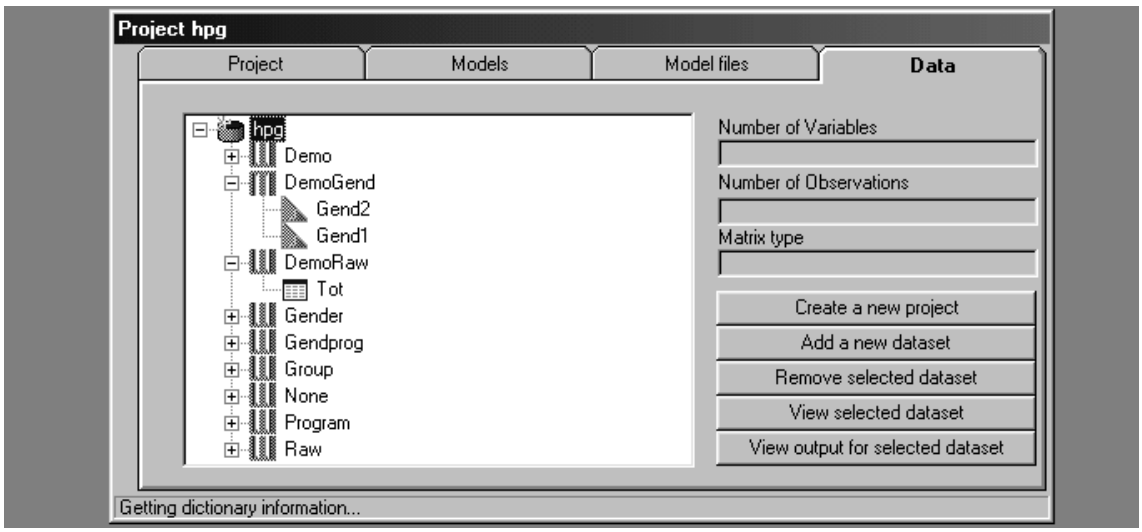
Next the variables to be imported are specified. Here all the variables in the input file are selected.

Missing Data

The same options for dealing with missing data are offered here as when a covariance matrix is computed. In the present example the option **Include all cases** is selected.

Finish

When the **Finish** button is clicked the data is read from the input file, and after any selection of cases the data is stored in the project. Imported data is stored in SPSS-format, and the project dictionary contains information about which variables are included in the data set.



Importing an External Matrix

The procedure for importing an external matrix offers functions both for entering the matrix elements manually into a grid, and for reading a matrix file into the grid. It is, however, recommended that a large matrix which only exists in non-computer readable form (e. g., in a published paper) is first entered into a file. Any text editor can be used for doing that but it must be observed that the file must be saved in text (or asci) format. Thus, if the Word program is used, for example, the **Save as ...** function must be used, and the file format specified to be *.txt*.

Matrices to be imported must be in lower triangular form, and must include the diagonal. However, the program reads the values in free format so the actual physical layout of the file is unimportant. For example, the correlation matrix:

```
1.00
0.10 1.00
0.20 0.30 1.00
0.40 0.50 0.60 1.00
```

may be entered as is shown above. But the matrix may also be entered as a string of values, separated by blanks, in the following order:

```
1.00 0.10 1.00 0.20 0.30 1.00 0.40 0.50 0.60 1.00
```

The elements may also be written on any number of lines, as long as the order among the elements is preserved. When the matrix is read the program checks that the file includes

the correct number of elements. A matrix with k variables consists of k(k+1)/2 elements, and if there are fewer or more elements in the file STREAMS produces an error message.

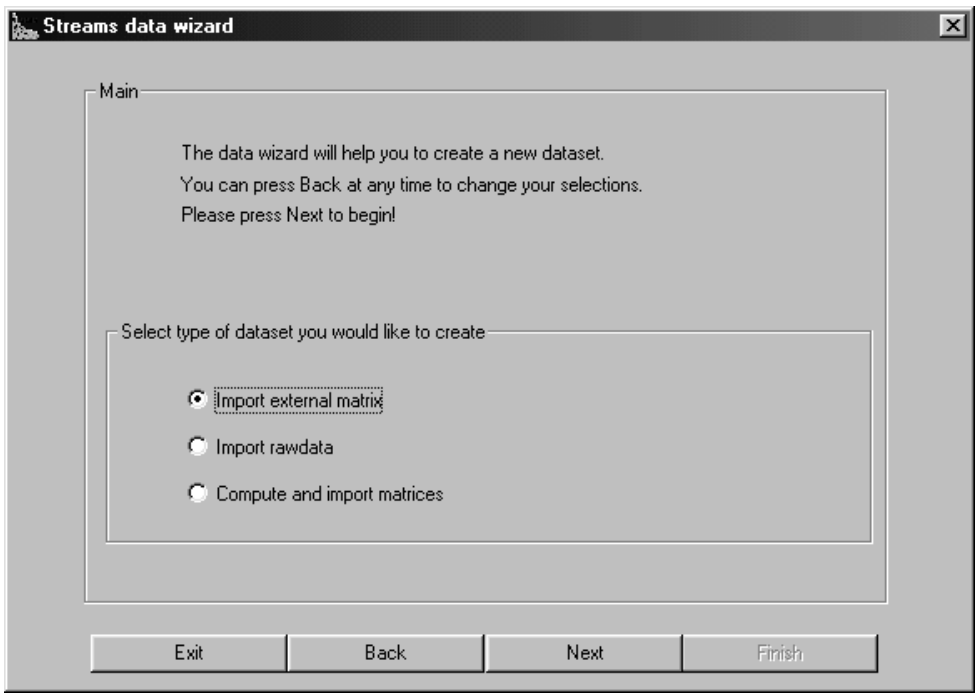
Suppose that we want to import an externally computed covariance matrix for the 6 subtests of the SweSAT (in the order WORD READ GI ERC DS DTM)

```
22.975
10.7517 13.7564
10.4216 6.52787 13.9127
12.9611 9.28106 8.06925 17.1139
6.98265 5.70548 5.46148 6.76535 15.6176
5.4772 4.6835 3.50274 5.31043 7.67652 10.378
```

This matrix is stored under the name *swesub.cov* in the NEW subdirectory under the *STREAMS\EXAMPLES\HPG* directory. There is also a vector of means, which is stored under the name *swesub.mea* in the same directory.

```
17.8791 15.5786 17.7668 17.4076 12.6425 14.8117
```

To import the covariance matrix and the means into the *new* project this project should be opened. On the *Data Wizard* form the option **Import external matrix** should be selected.

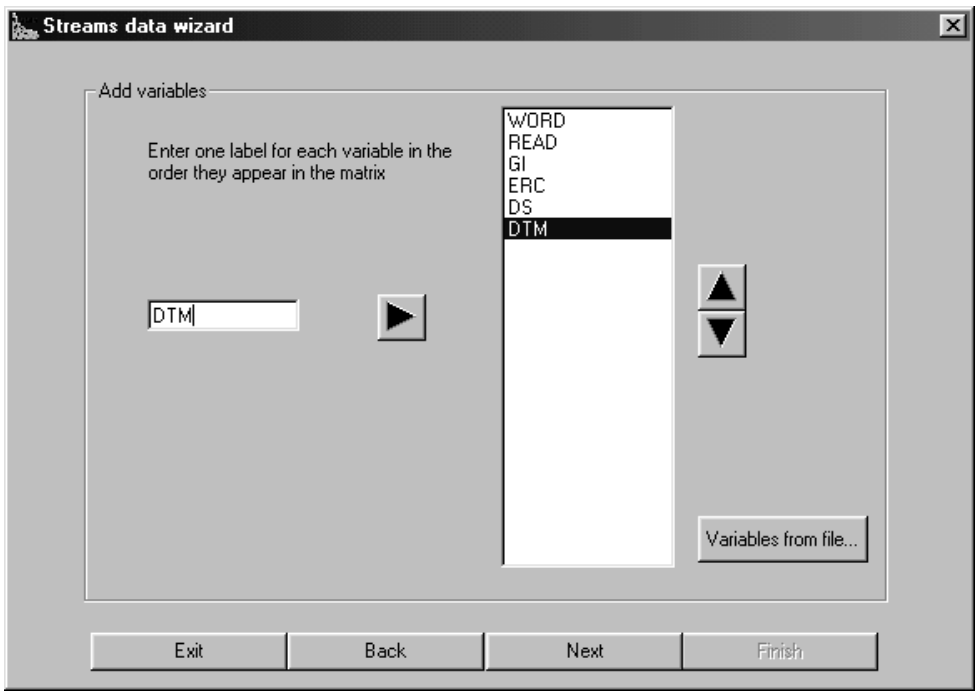


Labling the Matrix

Next the form for labling the dataset is presented, and it is completed in the same way as when raw data is imported, or a covariance matrix is computed. Here the categorization variable *DemoImp* is assigned, and the Group label is *Subtst*.

Add Variables

The variable labels are then to be supplied, which is done through entering them on the next form:

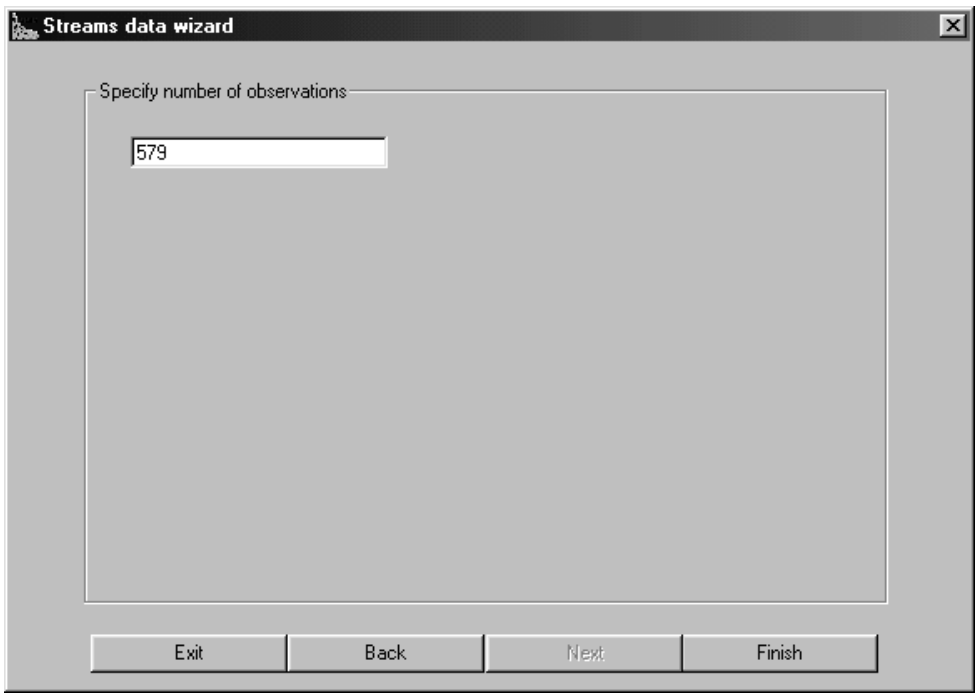


The variable labels may be entered in the text field, and are then transfered with the arrow into the list-box. The order of variables in the list-box must match the order of variables in the matrix. Should there be a need to change the order of variable labels, this may be done with the up and down arrows.

Variable labels may also be read from a file (which should have the suffix *.lab*). This is done through clicking the button **Variables from file ...** . The labels in the file should be entered one label on each line.

Number of Observations

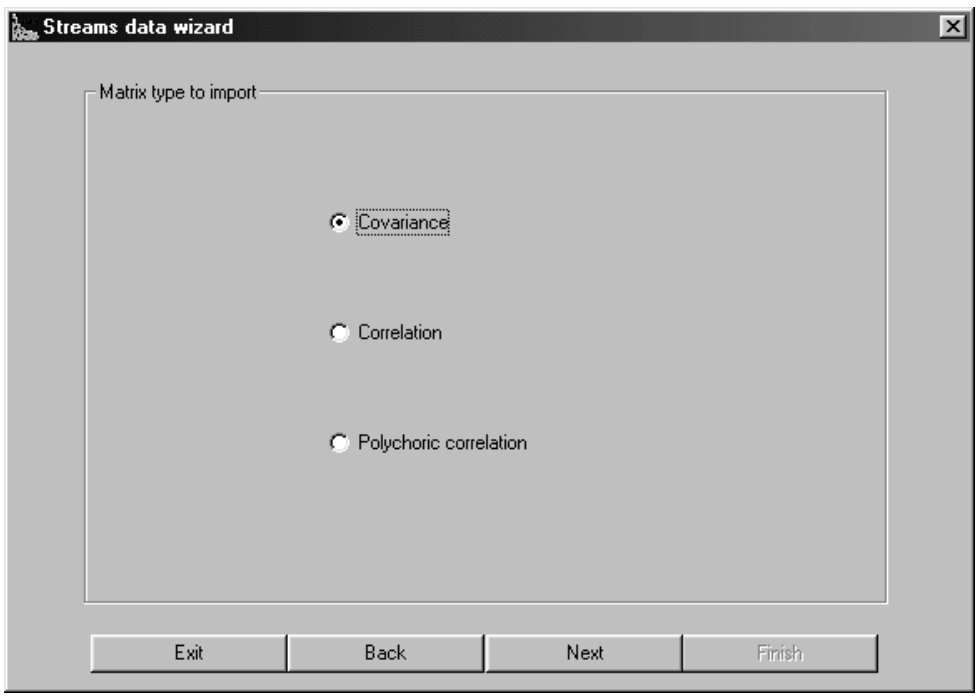
On the next screen the number of observations is entered:



The number of observations is entered in the field (here 579).

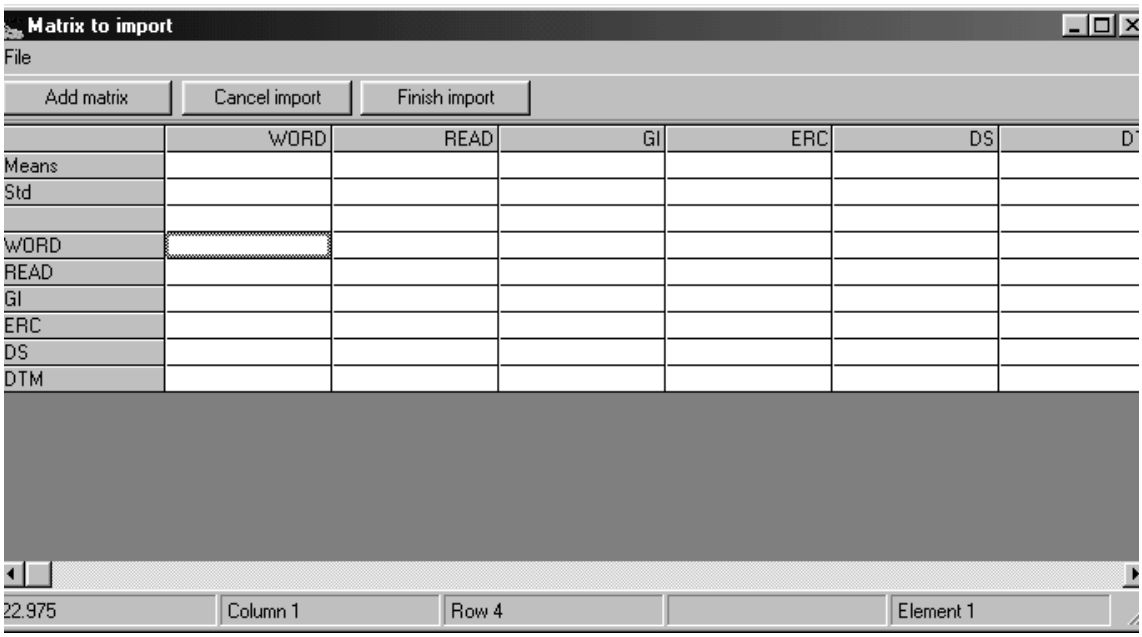
Type of Matrix

Next the type of matrix should be identified:

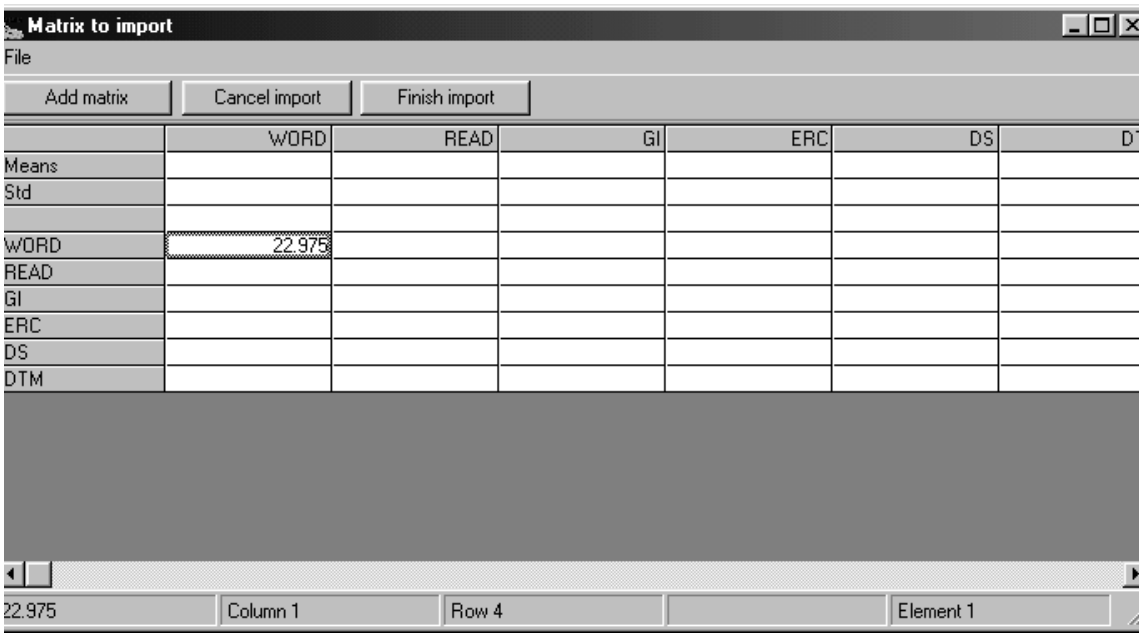


Matrix to Import

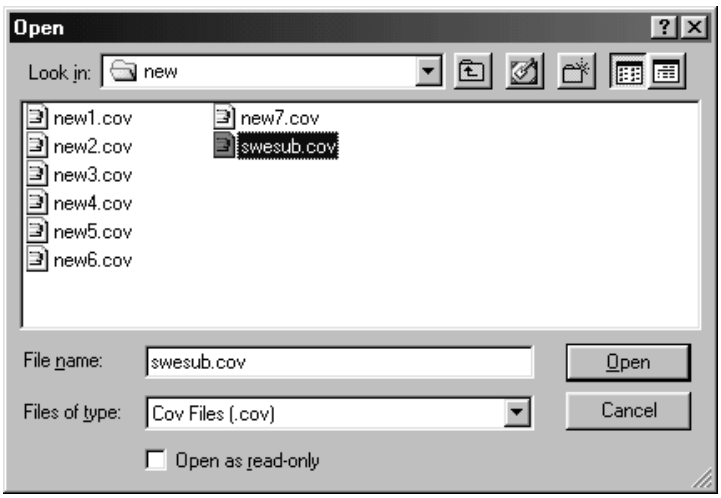
When the **Next** button is clicked, an empty form with variable labels on rows and columns is shown:



This grid may be used to enter matrix elements. The WORD variance element is first selected, and the matrix element is entered. After carriage return or the tab key is hit, the insertion point moves to the next element, which here is the READ WORD covariance element. In this way all 21 elements of the covariance matrix may be entered.

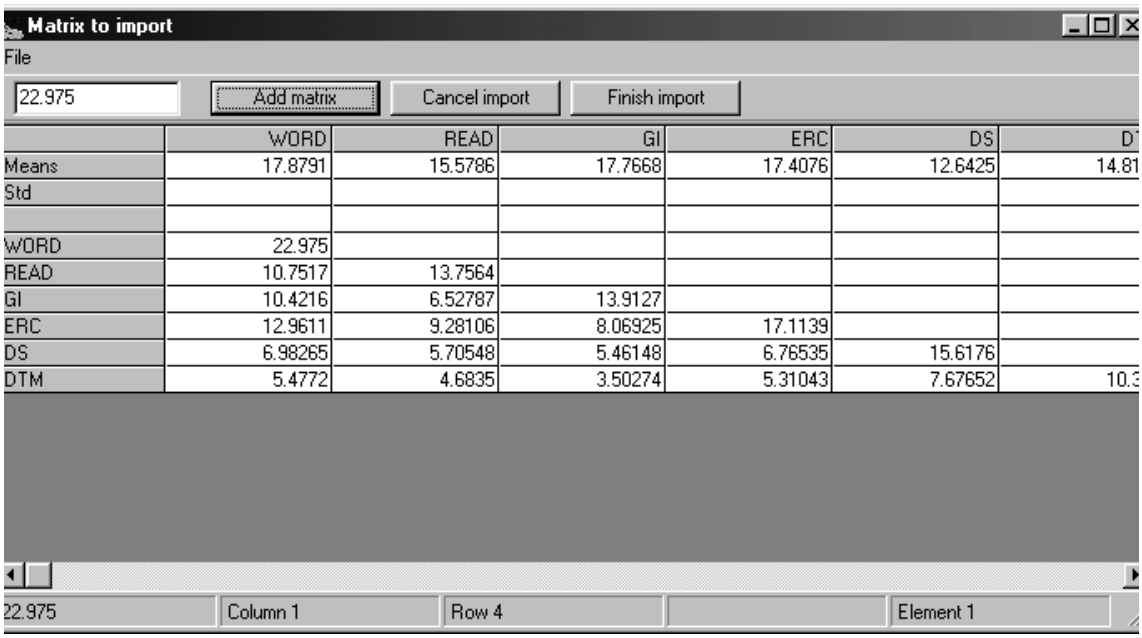


Another possibility is to read the matrix from file. This is done through clicking the button **Add matrix**, which presents a standard file-open dialogue:



There is a choice of two types of files: .cov and .cor files.

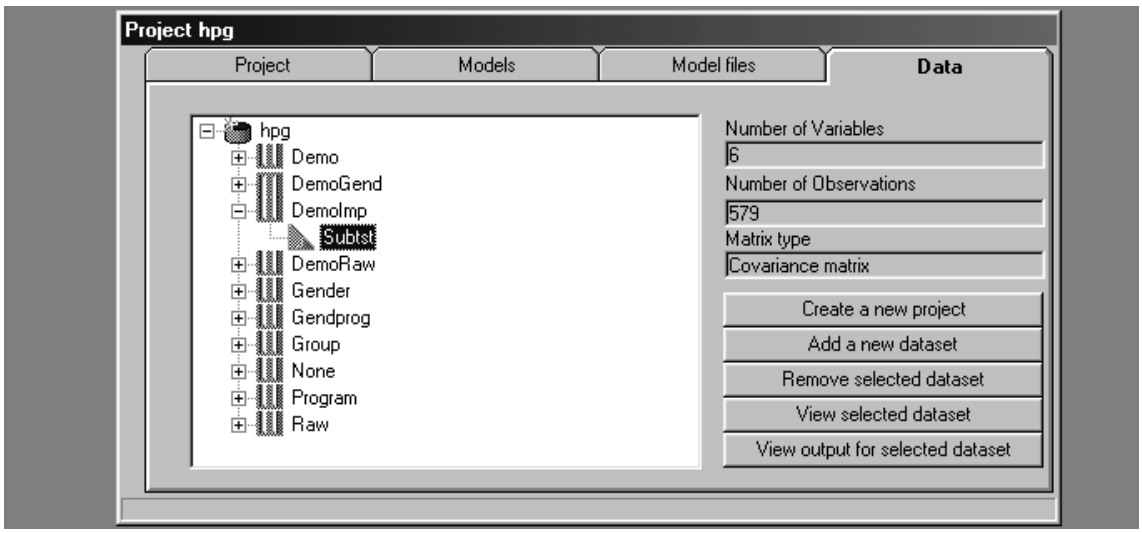
When the *swesub.cov* file is opened, the grid takes on the following appearance:



It should be observed that means are entered as well. This is because these are available in a file with the name *swesub.mea* which is in the same directory as the *swesub.cov* file. It is, of course, possible to enter the means directly into the grid as well. Should there be a need to enter standard deviations along with correlations this is also done automatically, if the standard deviations are in a file with the same name as the corelation matrix, but with the suffix *.std*. When the matrix is imported STREAMS computes a covariance matrix from the correlations and the standard deviations.

Finish

When the **Finish import** button is clicked, the matrix is added to the dictionary:



The matrix is now ready to be used for modeling purposes.

Importing Weight Matrices

Some estimators require more information than is contained in the covariance matrix. Thus, when the Weighted Least Squares (WLS) estimator in LISREL is to be used an **Asymptotic Covariance Matrix** (ACM) must also be computed; and when the Diagonally Weighted Least Squares (DWLS) estimator is to be used an **Asymptotic Variance Matrix** (AVM) must be computed. When polychoric correlations are analysed it is recommended that either the WLS or the DWLS estimator is used. These matrices may be computed with the PRELIS2 program (see, e. g., Jöreskog & Sörbom, 1998).

When PRELIS2 is used to compute the matrices, one or both of the asymptotic matrices may be requested. In order for STREAMS to be able to import these matrices they should be given the same name as the covariance matrix, but assigned a different suffix. Thus, the asymptotic covariance matrix should be given the suffix *.acm*, and the asymptotic variance matrix should be given the suffix *.avm*. If these files are present is the same directory as the matrix to be imported, they are automatically imported into the project, and when an estimation method is selected which requires the weight matrices, they are included in the model specification. It should be observed, however, that when the WLS or DWLS estimator is used it is not possible to make a model for a subset of the variables in the matrix, but a new matrix must be computed for each particular selection of observed variables to be modeled.

It should also be noted that computation of the asymptotic matrices (and particularly the ACM) is associated with some problems:

- When the number of variables is large the ACM matrix is tedious to compute and requires considerable space for storage.
- The WLS estimates can only be computed when the number of cases is large.

Thus, even though maximum likelihood estimates computed for an ordinary covariance matrix may not be theoretically optimal they do have certain practical advantages.

11 Inspecting and Maintaining Projects

Use of STREAMS tends to result in a large number of models, each of which is associated with several files. There is, therefore, often a need for ’housekeeping’ and STREAMS offers a set of utility routines for such tasks.

The utility routines are accessed via the *Project Window* and several will be discussed here:

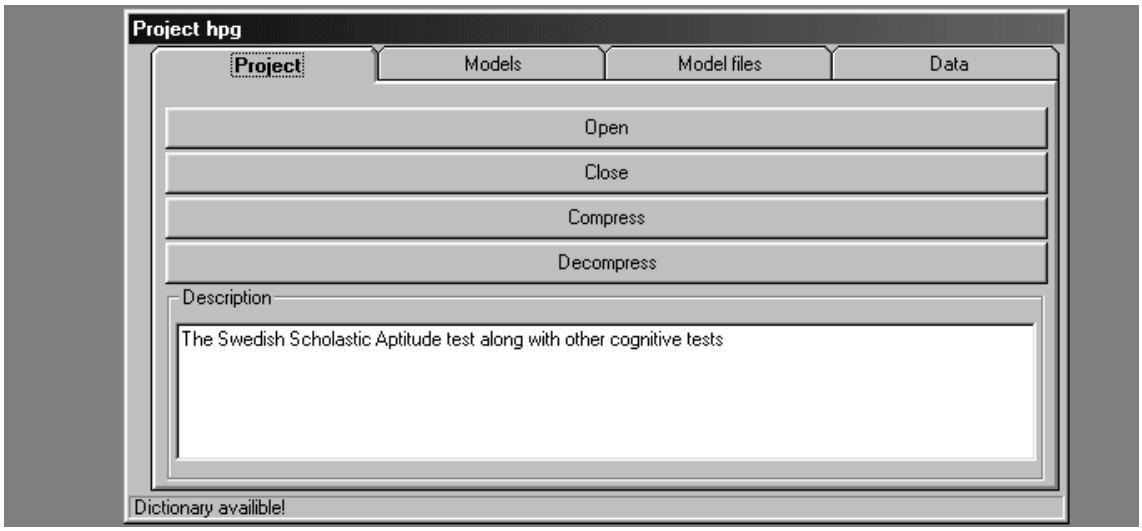
- Compress and Decompress Project
- Model files
- Data

Compressing and Decompressing Projects

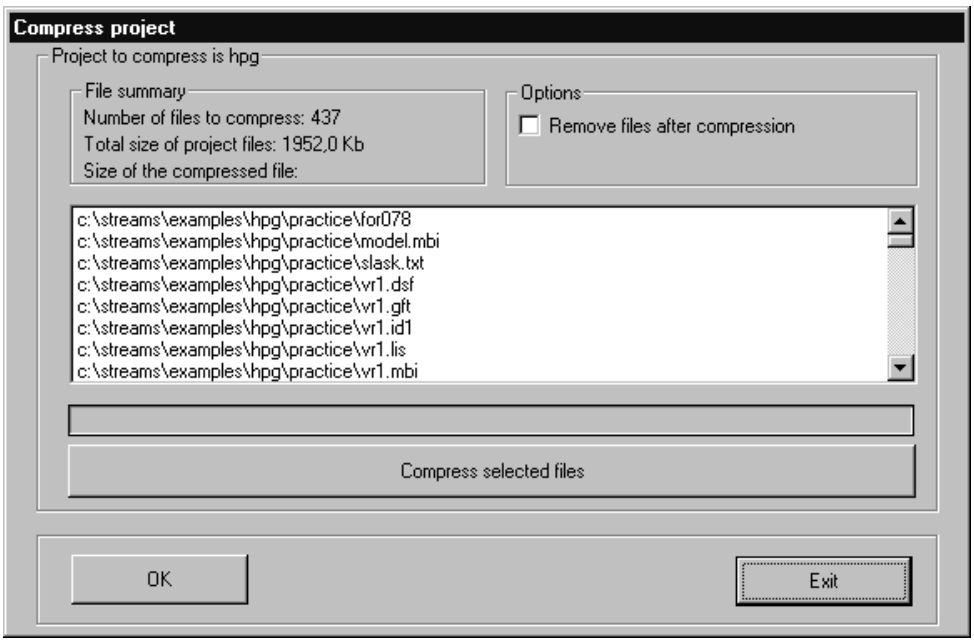
A STREAMS project consists of a large number of files, which makes it space-consuming and unwieldy to transport from one computer to another. In order to solve these problems STREAMS 2.5 offers a built-in function for compressing an entire project into a single (zip 2.04g compatible) file, as well as a function for decompressing such a file.

Compress

The compress function is accessed from the **Project** tab of the *Project Window.* When no project is open the **Compress** button is disabled, but as soon as a project is open, the button is enabled:



When the **Compress** button is clicked, the *Compress project* form is presented:

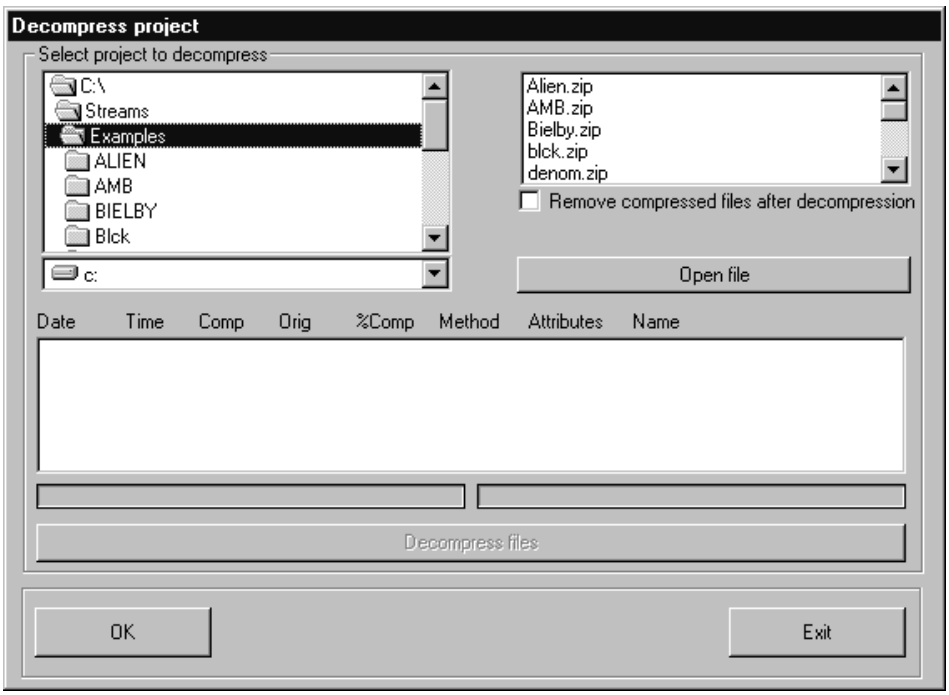


The form presents all the files (e. g., data and model files) associated with the open project. Information is also presented about the total size of the files. When the **Compress files** button is clicked a compressed file is created which includes all the files in compressed form. The compressed file is assigned the name of the project, with *.zip* as suffix (e. g., *hpg.zip*). This file is located in the folder above the *.mdp* file. Thus, when the *hpg.dct* project in *STREAMS\EXAMPLES\HPG* is compressed, the *hpg.zip* file is put in the *STREAMS\EXAMPLES* folder.

There is a choice whether the original files are to be kept or be deleted. To delete files the check-box labeled **Remove files after compression** should be clicked. This alternative is, of course, useful when a project is to be moved or put in an archive, while the alternative to keep the files may be useful when a copy is to be made of a project for purposes of distribution.

Decompress

The **Decompress** button on the **Project** tab is always enabled. When this button is clicked the *Decompress project* window is presented:



A dialogue box for selecting folders is presented, and in the list-box to the right all files with the suffix *.zip* is presented. One of these may be selected, and when the **Open** button is clicked the content in the compressed file is presented in the list box. In order to copy the files from the zip-file, the **Decompress files** button is clicked.

After decompression the files will be assigned the same pathnames as they had when the compression was done, and if the folders in the pathname do not exist they will be created. Thus, the project *hpg.zip* in the example above will be restored into the folder *STREAMS\EXAMPLES\HPG*. This is true also if the *hpg.zip* file is transferred to another computer where the STREAMS program has been installed in another folder. Decompression may also be done with the WinZip program, in which case there are better possibilities for controlling the placement of the project in the file structure.

The *Decompress project* form also presents a check-box labeled **Remove compressed file after decompression**.

Model Files

For each new model which is estimated STREAMS and the estimation programs jointly produce several files. In this section we describe tools, such as the **Model files** tab on the *Project window*, for managing models and their associated files.

Table 2 presents a summary of the types of files and their characteristics.It should be noted

TABLE 2. **Model Files in STREAMS**

Suffix	Description	Recreateable	Size
MBI	MB instructions	No	Small
STR	Binary file with estimates and model description	No	Small
LOG	Log file for error messages	Yes	Small
MOD	Description of the model produced by the pre-processor	Yes	Small
MLD	Description of the previous model which is used for backup purposes if no estimates are obtained for the new model.	Yes	Small
COV	All matrices (of all types) for all groups to be analyzed. The matrices only include the observed variables included in the model.	Yes	Small to medium
LIS	LISREL 8 instructions created by the pre-processor	Yes	Small to medium
EQS	EQS instructions created by the pre-processor	Yes	Small to medium
MPI	Mplus instructions created by the pre-processor	Yes	Small to medium
MXI	Mx instructions created by the pre-processor	Yes	Small to medium
OUT	LISREL 8 output file	Yes	Large
EQO	EQS output file	Yes	Large
MXO	MX output file	Yes	Large
MPI	MPlus output file	Yes	Large
AMP	Amos parameter estimates	Yes	Small
DAT	EQS parameter estimates	Yes	Small
EST	LISREL 8 parameter estimates	Yes	Small
GFT	LISREL 8 goodness of fit statistics	Yes	Small
SVT	LISREL 8 standard errors	Yes	Small
PVT	LISREL 8 estimates of free parameters	Yes	Small
SIT	LISREL 8 Sigma matrix	Yes	Small
PRT	Post-processor output	Yes	Medium

that the column labeled **Recreatable** has a *yes* for those files which are so easy to recreate that a simple rerun, but with optimal start values, of the *.mbi* file suffices. If start values are not to be copied from an existing version of the model all files except the *.mbi* file may in fact be erased, and all the other files may still be recreated.

The column labeled **Size** presents a rough estimate of the expected size of each file. The size estimates are made in terms of three rough categories (small, medium and large) and are quite approximate. It nevertheless seems to be quite a general rule that the output files from the estimation program and the post-processor consume most space on the hard disk. The output files from the pre- and post-processors also require quite a lot of space compared to the work files needed for start values and post-processing.

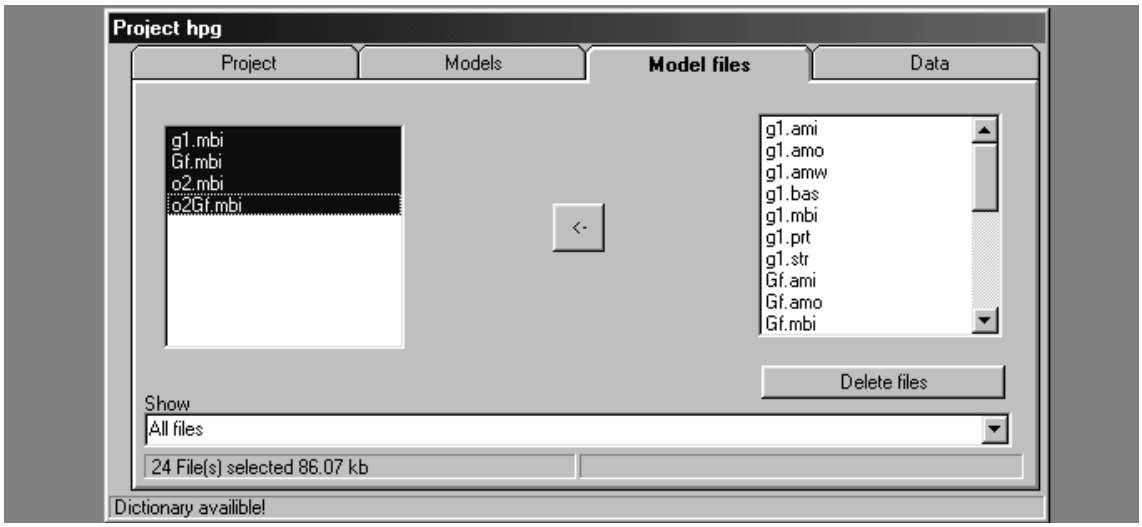
To take a concrete example, all the files for 9 models in one project together required 282 Kb. The 9 *.out* files required the major share of the space (171 Kb) and together the three

most easily recreatable files (*.lis*, *.out*, and *.prt*) took no less than 243 Kb. There is thus a need for tools which control models and file types.

Most of the files which are not needed for future use are automatically removed by STREAMS, but in some situations (such as when a program aborts its operations unexpectedly) files may, nevertheless, be left in the model folder. As is described below the **Model files** tab on the *Project window* may be used to delete such files.

This tab presents three check-boxes, which may be used to specify whether the pre-processor, estimation and post-processor file is to be kept after model estimation is complete. Thus, if none of these check-boxes are checked no output, except for the information in the *.str* file, will be kept.

The **Model files** tab on the *Project Window* may be used to investigate which files are available in a model folder, what amount of space different types of files require, and facilities for removing files are also offered.



The list-box on the left-hand side is used to select one or more models through clicking, shift-clicking, and so on. The files associated with the selected models are presented in the list-box on the right-hand side, and the total amount of space occupied by the files is presented on the status line (e. g., **24 file(s) selected 86.07 kb**).

However, only those files are presented which satisfy the file type criteria which may be imposed with the drop-down menu. The following may be selected:

- **All files**
- **All files but .mbi**
- **Recreatable files**
- **Pre-processor output**
- **Estimation output**
- **Post-processor output**

When the **Remove** button is clicked the selected files are deleted. This is an irreversible process, so it is necessary to be careful.

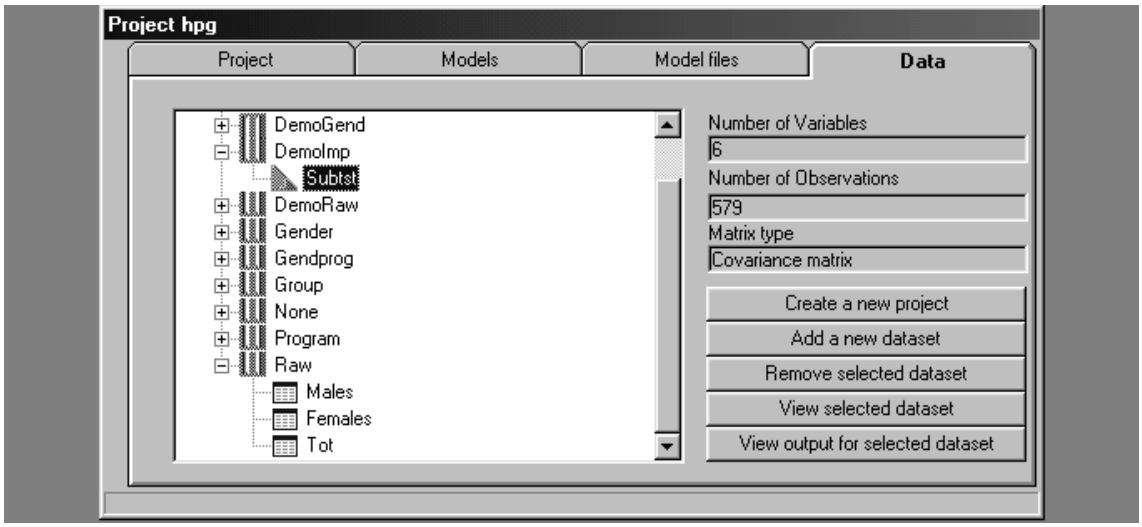
There are, of course, many reasons why models and files should be removed. One useful strategy is to start by deleting those models which should be taken away entirely. Any

modeling activity typically yields a considerable number of intermediate models which serve no function after the model generating phase has been completed. If there is a need to inspect the MB code before a model is selected for removal the **Model** tab may be clicked, and the model opened with the check-box **Preview model before opening** checked.

After all files associated with the unwanted models have been removed, it may be a good idea to take away the output files from the estimation programs, which tend to be the most bulky files, and which sometimes only contain information which is available in the post-processor output file. If there is a need to recapture more hard disk space, the pre-processor and post-processor files may then be removed. The next step would be to take away all the recreatable files. Then all files, except the *.mbi* file, could be removed. This would make it possible to estimate the model again, but this step will remove the *.str* file, so if the process of estimation relies on better start values than those which may be produced by the estimation program, nonconvergence of the iterations may be a problem. If, finally, the MB instructions in the *.mbi* file are removed the whole model has been removed, and cannot be recovered.

Data

The **Data** tab on the *Project Window* allows inspection of the data imported into the project:



The different folders are shown and when they are expanded the dataset labels are listed along with their icons. When a dataset is selected the form presents the number of variables, number of observations and matrix type. Chapter 2 (see "Opening and Inspecting an Existing Project", page 22) presents more information about how to inspect projects.§

Sometimes there is a need to take away data from the project. This is done when the **Remove selected dataset** button is clicked. The dataset to be removed from the project is identified in the usual manner, through selecting first the folder and then the dataset label. A complete folder with all its datasets may also be removed in the same way. When the **Remove selected dataset** button is clicked, the dataset or folder is removed from the project. There is no undo function for this function, so care needs to be excercised.

Part 4

Principles of Design and Operation

The fourth part of the User's Guide presents the MB language in detail. The interfaces to the different estimation programs are also documented, and it is described how STREAMS is installed.

12 The Model Building Language

Previous chapters have introduced the basic elements of structural equation modeling with the MB language. This chapter presents the MB language in greater detail.

Basic Characteristics of the Model Building Language

The Model Building (MB) language has been constructed to take advantage of the positive characteristics of the other languages for specification of structural equation models (see Chapter 13 “Estimation Program Interfaces” on page 211), and it also aims to improve upon these in certain respects. The following are some characteristics of MB:

- Four types of variables may be used as independent and dependent variables: manifest and latent variables, and residuals in manifest and latent variables.
- Four types of statements about relations, variances, covariances and means of the variables allow easy specification of a wide range of models.
- In multiple group modeling, groups are refered to with labels in the MB statements.
- Constraints of equality over groups and variables are specified through enclosing lists of variable and group names in parentheses.
- A general equality constraining statement may be used to impose constraints of equality on any two free parameters.
- Fixed parameter values are assigned through supplying the parameter value in the statement.
- The MB instructions are translated into statements in the Amos, EQS, LISREL, Mplus or Mx languages, and the model is estimated with one of these programs.
- If one or more previously estimated models are available which involve the same relations, start values may be copied from these.

- Model specification is done with an intelligent editor, offering push buttons for choice of commands, list boxes for choice of variables, and forms for choice of options.

The MB language is described in greater detail below.

Categories of Variables in MB

As has already been mentioned MB differentiates between four categories of variables: manifest variables, latent variables, errors in manifest variables, and residuals in latent variables. These categories are described below.

Manifest variables

Manifest variables, or observed variables, are variables which may be directly observed, such as scores on a vocabulary test and gender, just to take two examples. Variables in this category are referred to with labels of at most 7 letters or digits. It is recommended that capital letters are used, e. g. VOC, GENDER. In path diagrams manifest variables are drawn as rectangles.

Latent variables

Latent variables, or unobserved variables, or factors, are variables which cannot be directly observed. Such variables typically represent abstract aspects, and phenomena which cannot be directly observed. A latent variable may, however, be defined in models if there are enough manifest variables which are indicators of the latent variable. Variables in this category are referred to with labels of at most 7 letters. It is recommended that at least one letter in the label is lower-case (e. g., *Gf* and *Gc*). In path diagrams latent variables are drawn as ellipses.

Residuals in manifest variables

Residuals in manifest variables, or errors, is a category of variables which is created when manifest variables are used as dependent variables, for example through being used as indicators of one or more latent variables. The latent variable is conceived of as an independent variable which accounts for scores in the manifest variables, which thus are dependent variables. However, all variance in the manifest variables is not being accounted for, because in each manifest variable there are additional sources of influence beyond the latent variable. These additional sources of variance thus are the errors in manifest variables, and for each manifest variable which is an indicator of a latent variable, such a residual variable an principle exists. The residual variable may represent pure random error, which is not of any interest for model building purposes, or it may represent systematic sources of variance of potential interest for the model. Variables in this category are referred to by adding an ampersand (‘&’) to the name of the manifest variable, e. g., VOC&. This is done automatically by the MB program, and as soon as a manifest variable has been used as a dependent variable, the residual variable may be referred to. The residuals in manifest variables are drawn as ellipses.

Residuals in latent variables

Residuals in latent variables, or disturbances, exist for latent variables which are used as dependent variables in structural equation models. In almost all dependent variables there are additional sources of influence beyond the independent variables included in the model. It is sometimes desirable to include these residual sources of variance as independent variables (and sometimes also as dependent variables) in structural equation models. Variables in this category are formed in the same way as the residuals in manifest variables, i. e. by adding an ampersand ('&') to the name of the latent variable, e. g. *Gc&*. This category of variables too is automatically created by MB. In path diagrams residuals in latent variables are drawn as ellipses.

The MB Statements

MB offers four types of instructions for making statements about RELations, VARiances, COVariances and MEAns of a set of variables. However, before any of these statements may be issued it is necessary to describe the problem, and to declare variables and groups. This is done through a set of statements which are described below.

The TI Statement

This command provides one line with descriptive information about the problem, the model and the data. The command may be repeated up to 10 times, e. g.

```
TI Three-factor model for girls,
TI cognitive variables
```

The MO Statement

This statement allows specification of several keyword parameters. This is done typically through filling out the forms under the **Model** button.

PR=*pname*	The *pname* (1-8 characters) is the project name, and is normally automatically supplied by STREAMS.
NAME=*mname*	The *mname* (1-60 characters) is the name of the model. If a model with *mname* already exists it will be replaced by the new model.

The MO statements also allow several longer text expressions, which are more or less self-explanatory, to direct the program operation. These text strings are normally generated by STREAMS, but if they are entered with an editor, they must be entered exactly as they are presented here. Different sets of expressions concern different categories of options as described below.

Start values

```
Start values: computed by estimation program
```

Start values may also be copied from previously estimated models (see below)

Type of matrix to be analyzed

```
Matrix: CM(Covariance matrix)
Matrix: KM(Correlation matrix)
Matrix: PM(Polychoric correlation matrix)
```

The default is that the covariance matrix is to be analyzed. If a CM or KM is requested when the matrix referred to is PM, the statement is disregarded. If a CM is requested when the matrix in the project is a correlation matrix the statement is also disregarded.

Means

```
Means not included in model
Means included in model
Always allow latent variable means
```

The latter alternative implies that no restrictions are imposed on estimation of latent variable means. If this option is not given the means and intercepts of latent variables are always constrained to zero in the first groups of cases, which is the normal specification in multiple-group modeling. In certain situations, such as in growth-curve modeling, means of latent variables need, however, to be estimated in all groups, and in these situations the option *Always allow latent variable means* must be supplied.

Means are by default included in the model.

Model type

```
Model Type: Two Level
Model Type: Incomplete Data
Model Type: Incomplete Data H1
```

These options indicate that specialized types of models are to be set up (see Chapters 5 and 6), and when none of these is selected an ordinary model is created.

Multiple groups

```
One-group model
Separate one-group models
Multiple groups without constraints
Multiple groups with constraints
```

The option *Separate one-group models* is only available when LISREL is used as estimation program.

The default is a one-group model, or a multiple-group model with constraints of equality applied to all parameters.

When models for incomplete or two-level data are fitted the term `group` is not adequate, so the term `population` is used instead. In these cases the following options are available:

```
One-population model
Multiple populations without constraints
Multiple populations with constraints
```

The default is a one-population model, or a multiple-population model with constraints of

equality applied to all parameters.

Type of model to be generated

```
Create instructions for: Amos
Create instructions for: EQS
Create instructions for: Mplus
Create instructions for: Mx
Create instructions for: LISREL Y-model
```

The default is that instructions are generated for the LISREL model formulated in terms of Y-variables.

Program version

```
Amos 3.5
Amos 3.6
Amos 4.0
EQS 4
EQS 5
EQS 6
Mplus 1.0
Mx 1.4
LISREL DOS 8.03
LISREL DOS/Extender 8.03
LISREL DOS 8.12
LISREL DOS/Extender 8.12
LISREL DOS/Extender 8.14
LISREL DOS/Extender 8.20
LISREL DOS/Extender 8.30
```

The LISREL DOS Extender 8.30 program is currently the default, but the default estimation program may be changed.

The OP Statement

The OP statement is used to transfer options to the estimation program. When the **Option** button is clicked the appropriate *Option* form is produced and may be filled out. The LISREL OU statement (see Jöreskog & Sörbom, 1993a, b), for example, is constructed by STREAMS and put on the OP statement, from where it is transferred into the LISREL instructions generated by the pre-processor. The OP statement also transfers options concerning estimation method, goodness-of-fit tests, output, and so on to the Amos, EQS, Mplus and Mx programs (see Chapter 13).

The STA Statement

One or more previously estimated models may be referred to in one or more STA statements, from which MB copies starting values. Only one model is referred to in each STA statement. These statements are prepared by STREAMS when the **Start Values** option on the **Model** form is selected. It is possible to use a model estimated with any program as a source of starting values for any other program. Observe, however, that multi-group EQS models do not generate start values.

NAME=*mname* The *mname* is the name of a previously estimated model. The model should, of course, as far as possible include the same manifest and latent variables as the model to be estimated.

The DAT Statement

This statement is used to identify the datasets to be included in the model. At least one DAT statement must be given and it may refer either to a set of raw data, or to a matrix of relations among observed variables:

FOLDER=*label* *Label* is the name of the project folder to be used for selection of the dataset for which the model is to be estimated (e. g., Gender).

DATLAB=*label* *Label* refers to the dataset to be selected (e. g., boys).

For example:

```
DAT FOLDER=Gender DATLAB=Boys
DAT FOLDER=Gender DATLAB=Girls
```

These statements identify two groups of cases, boys and girls, to be included in the model.

The MVR Statement

This command is used to identify the manifest variables to be included in the model. The MVR statements are created by STREAMS when the **MVR** button is clicked. The syntax of this command is:

```
MVR variable list
```

The variables are referred to with the labels given in the dictionary. Each MVR statement may only comprise one line, but any number of single-line MVR statements may be used. The variables will be included in the model in the order listed on the MVR statement. An example is:

```
MVR TEST1 TEST2 TEST5 TEST4
```

This statement selects four observed variables from the dictionary, and specifies the order in which they will appear in the model.

The MV2 Statement

This command is used to identify the group-level manifest variables to be included in a two-level model (see Chapter 6). The MV2 statement is created by STREAMS when the **MV2** button is clicked. The syntax of this command is:

```
MV2 variable list
```

This statement is used to declare variables which have been observed at the group level, but not at the individual level. If there are no such group-level manifest variables no MV2 statement should be used. Observe that the individual level variables are always available at the group level in the form of group means, and these variables are always automatically available in STREAMS. The aggregated variables are at the group level referred to

with a "2" as a prefix to the variable name at the individual level. For example, if an individual variable is called TEST1, the corresponding group level variable is called 2TEST1.

When a manifest variable has been declared as a group-level manifest variable this variable also must have the "2" as a prefix when it is referred to in MB statements. This is because the MV2 statement refers to the manifest variables in the project dictionary where no distinction is being made between variables at different levels of observation, while the MB statements in a two-level model refer to structures at two levels. For example, assume that the following statement has been made to declare teacher gender, teacher age, and teacher experience as group-level manifest variables:

```
MV2 TGEND TAGE TEXP
```

To use these as independent variables to predict class achievement (2ACH, say), the following REL statement must be used:

```
REL 2TGEND 2TAGE 2TEXP -> 2ACH
```

The fact that the character "2" is used as a prefix in variable labels to separate group level variables from individual level variables implies that at most 6 characters should be used in variable labels when two-level modeling will be done.

The LVR Statement

This command is used to declare the latent variables in the model and is prepared by STREAMS when the **LVR** button is clicked. The syntax is:

```
LVR variable list
```

The names of latent variables may be freely chosen, but it is recommended that they should contain at least one lower-case letter. If upper-case letters are used for manifest variables this makes it easier to separate the two categories of variables both in input and in output. An example is:

```
LVR g Gc Gv
```

The LV2 Statement

This command is used to declare the latent variables at the group level in a two-level model (see Chapter 6) and is prepared by STREAMS when the **LV2** button is clicked. The syntax is:

```
LV2 variable list
```

The names of latent variables may be freely chosen, but it is recommended that they should contain at least one lower-case letter, and at least one letter which indicates that this is a group-level latent variable. It may thus be good practice to use the letter W as a suffix in labels for the individual latent variables, and the letter B as a suffix in labels for the latent variables at group level. Another possibility is to use the "2" as a prefix for the group level latent variable labels as well. Thus, two examples could be:

```
LV2 gB GcB GvB
```

or

```
LV2 2g 2Gc 2Gv
```

The REL Statement

The REL statement expresses a relation between one or more independent variables and one or more dependent variables for one or more groups. The general form of the command is:

```
REL [Value] [(Grp1 Grp2 ...)] (Indep1 Indep2 ...) -> (Dep1
        Dep2 ...)
```

Several of the components of the statement are optional:

- If a *Value* is given immediately after the command name, this is interpreted as a fixed parameter value which applies to all the relations implied by the statement. If *Value* is omitted, coefficients for the relations are estimated as free or constrained parameters.
- If one or more group labels are supplied the REL statement is interpreted to apply to these groups of cases. If no group labels are included, the statement is interpreted to apply to all groups of cases included in the model. The **Model** form allows the user to determine whether the default is to have all parameters constrained over groups, to have no constraints over groups, or if separate one-group models are to be generated.
- If variable or group labels are enclosed in parentheses, equality constraints are imposed over these variables or groups of cases. If parentheses are omitted there are no equality constraints.
- The REL statement must include at least one independent variable, and at least one dependent variable.
- By default variances of independent variables and residuals are automatically introduced as free parameters.

The simplest possible REL statement thus identifies a relation between one independent variable, which may belong to any one of the four categories of variables, and one dependent variable, which may be a manifest or a latent variable (residuals may be used as dependent variables as well, but such models are usually equivalent with models that specify the variables themselves to be dependent variables, and they may be difficult to conceptualize). An example of a single REL statement is:

```
REL g -> TEST1
```

Here the independent variable is a latent variable and the dependent variable is a manifest variable. Because latent variables do not have a scale, a manifest variable is typically used to define the scale of the latent variable through a fixed relation of unity. The first manifest variable encountered in a REL statement for an independent latent variable is used to define the scale of the latent variable, through assigning a fixed value of unity for this relation. Thus, TEST1 will be the standardization variable if this is the first statement in which the latent variable g is used. If another manifest variable is preferred as the standardization variable for g we can achieve that through the SCL statement (see below).

Manifest variables may also be used in REL statements, e. g.:

```
REL GENDER -> TEST1
```

In this example both the independent variable and the dependent variable are manifest variables. Because manifest variables may not be directly involved in relations with one another in LISREL the MB program automatically creates one latent variable for each manifest variable.

In the following example the residual in TEST1 is used as an independent variable:

```
REL TEST1& -> TEST2
```

Here too the MB program creates a latent variable to represent the residual.

Latent variables may also be used as independent and dependent variables, e. g.:

```
REL g -> Gc
```

Once a latent variable has been used as a dependent variable, a residual is made available for use in further modeling statements, e. g.:

```
REL Gc& -> TEST2
```

If we want to assign the fixed parameter value of 2.5 to the coefficients for the relations between g and three manifest dependent variables we can do that with the statements:

```
REL 2.5 g -> TEST1
REL 2.5 g -> TEST2
REL 2.5 g -> TEST3
```

A more efficient way to accomplish this would be to list the three dependent variables in the same statement, i. e.:

```
REL 2.5 g -> TEST1 TEST2 TEST3
```

When more than one independent variable and more than one dependent variable is specified there will be one relation for each combination of variables. For example, if GENDER and AGE are independent variables and TEST1, TEST2 and TEST3 are dependent variables, the following statement specifies six relations:

```
REL GENDER AGE -> TEST1 TEST2 TEST3
```

The coefficients for the six relations are estimated as free parameters. We may, however, wish to impose equality constraints on some of the relations. The following statement imposes constraints of equality over the three dependent variables.

```
REL GENDER AGE -> (TEST1 TEST2 TEST3)
```

This statement thus specifies two coefficients to be estimated for the six relations: one for the regression of the three dependent variables on GENDER, and one for the regression of the three dependent variables on AGE. If we want to impose equality constraints for only two dependent variables (TEST2 and TEST3, say), more than one REL statement must be used, because the syntax of the REL statement does not allow for a mixture of constrained and unconstrained relations. We could thus write:

```
REL GENDER AGE -> TEST1
REL GENDER AGE -> (TEST2 TEST3)
```

These two statements will specify four different coefficients to be estimated for the six relations.

Constraints of equality may also be imposed on the independent variables. Thus, the following statement specifies six relations to be estimated, but with constraints of equality imposed on the coefficients for the two independent varibles:

```
REL (GENDER AGE) -> TEST1 TEST2 TEST3
```

If constraints of equality are imposed both on dependent and on independent variables the six relations will have one common estimated coefficient.

So far it has been assumed that there is only one group of cases but some examples of multiple group modeling will now be considered. Assume that two groups of cases have been selected from the dictionary through the statements:

```
DAT FOLDER=Gender DATLAB=Boys
DAT FOLDER=Gender DATLAB=Girls
```

If the REL statement does not include any group label the relations will apply in both groups, and the parameters will be estimated under constraints of equality over the two groups if that has been selected as the default, or without any constraints of equality if that option has been selected as the default.. The following statement also estimates the parameters without any equality constraints over groups:

```
REL Boys Girls g -> TEST1 TEST2 TEST3
```

Another way to express this is to use the following two statements:

```
REL Boys g -> TEST1 TEST2 TEST3
REL Girls g -> TEST1 TEST2 TEST3
```

It should be observed that these alternative statements do not produce exactly the same model. This is because in the latter case there are no equality constraints imposed on the residuals of the manifest variables, while such are imposed in the former specification.

If we want to impose constraints of equality for an identified (sub)set of groups, this can be done through enclosing the list of group labels in parentheses, e. g.:

```
REL (Boys Girls) g -> TEST1 TEST2 TEST3
```

Observe that it is not possible to use more than one set of parentheses for identifying group constraints in the same statement; nor is it possible to mix different types of expressions in the same statement. Thus, more complex patterns of constraints over different subsets of groups must be expressed on several different REL statements.

The VAR Statement

This command identifies a list of variables the variances of which are to be estimated as free, fixed or constrained parameters in the model. The statement is constructed by STREAMS when the **VAR** button is clicked. The syntax is:

```
VAR [Value] [(Grp1 Grp2 ...)] (variable list)
```

The optional constant is used to assign *Value* to one or more fixed parameters in the same way as in the REL statement. The rules described for reference to multiple groups in the REL statement also apply to the VAR statement. As is true for all the other statements, the parentheses around lists of variable or group labels indicate constraints of equality and are optional.

For example, the following two statements imply that the variances of errors in manifest variables are constrained to be equal over grades 8 and 9, but unconstrained in grade 7:

```
VAR Grade7 TEST1& TEST2& TEST3&
VAR (Grade8 Grade9) TEST1& TEST2& TEST3&
```

As has already been mentioned each independent variable referred to in a REL statement is by default assumed to have a variance to be estimated as a free model parameter, and each dependent variable is assumed to have a residual variance to be estimated. Thus in many cases the VAR statement may be omitted.

The COV Statement

This command identifies a list of variables the covariances of which are to be estimated as free, fixed or constrained parameters in the model. The command is constructed by STREAMS when the **COV** button is clicked. The syntax is:

```
COV [Value] [(Grp1 Grp2 ...)] (variable list)
```

The rules for identifying fixed parameters and constraints over groups which have been described for the REL and VAR commands apply here too. For example:

```
COV Males Females Gf Gc Gv
```

specifies the model to estimate covariances among the independent latent variables Gf and Gc, Gc and Gv, and Gf and Gv, in such a way that the two identified groups obtain unconstrained estimates.

The statement

```
COV TEST1& TEST3&
```

specifies a covariance to be estimated between the errors of the manifest variables TEST1 and TEST3.

It should be observed that the COV statement does not imply any effect with respect to the variances of the variables.

The MEA Statement

This statement identifies a list of variables the means of which are to be estimated as free, fixed or constrained parameters in the model, and it is constructed by STREAMS when the **MEA** button is clicked. The syntax is:

```
MEA [Value] [(Grp1 Grp2 ...)] (variable list)
```

For example, the statement:

```
MEA Gf Gc Gv
```

specifies the model to estimate means of the latent variables Gf, Gc, and Gv. It must be observed, however, that normally means of latent variables may only be estimated when the analysis comprises two or more groups of cases. In a one-group analysis where means are available the means on observed variables are estimated as free parameters, and means on latent variables are constrained to be zero (unless the option "Always allow latent variable means" has been selected, see page 184). As in the other MB statements the optional constant is used to identify fixed parameters, and the rules for identifying constraints over groups are the same as in the other MB statements.

The EQ Statement

The EQ statement is used to constrain any two free parameters to be equal. The syntax is:

```
EQ Statement1 AND Statement2
```

where the two statements identify one free parameter each. To impose constraints of equality over more than two parameters the EQ statement is repeated the necessary number of times with the same Statement1 and another Statement2.

It should be observed that the parentheses which are available to impose constraints of equality in all statements generally is the most simple and convenient method to impose constraints of equality. This method is not general, however, and there are many situations where it is not possible to express the needed constraints with the parentheses. If, for example, we need to express constraints of equality on the regressions of two different manifest variables on two different latent variables this cannot be done with the parentheses. Or if, to take another example, we want to constrain a variance in a latent variable to be equal to regression coefficient this cannot be done with the parenthesis method.

The EQ statement is general, however, and may be used to express any kind of equality constraint. Suppose, for example, that we want to constrain a series of regressions between six variables (y1-y6) in a simplex model to be equal. This may be done with the following set of EQ statements:

```
EQ REL y1 -> y2  AND  REL y2 -> y3
EQ REL y1 -> y2  AND  REL y3 -> y4
EQ REL y1 -> y2  AND  REL y4 -> y5
EQ REL y1 -> y2  AND  REL y5 -> y6
```

An EQ statement to constrain the error variance of the manifest variable X1 to be equal to the regression coefficient for the regression of X1 on x would be:

```
EQ VAR X1&  AND  REL x -> X1
```

Defining Scales

The SCL statement may be used to identify one variable as the scaling variable for a latent variable. This thus implies that there will be a fixed relation of unity between these variables. The SCL command is issued when the **SCL** button is clicked. The syntax is:

```
SCL variable1 variable2
```

where *variable2* is used to define the scale of *variable1*. For example, the command:

```
SCL g NUMSER
```

implies that the latent variable *g* will have its scale defined by the observed variable NUMSER. The command

```
SCL Gv Vz
```

causes the latent variable *Gv* to have its scale defined by the latent variable *Vz*.

Limitations of the MB language

Even though the MB language allows a wide range of models to be formulated, not all

types of model may be expressed. Thus, the general non-linear constraints which are supported both by LISREL and EQS cannot be expressed in MB.

It should also be observed that the automatic assignment of scales to latent variables causes some order dependency among MB statements. Thus, in a full structural equation model the measurement models must be specified first, because otherwise the latent variables will be used as scaling variables.

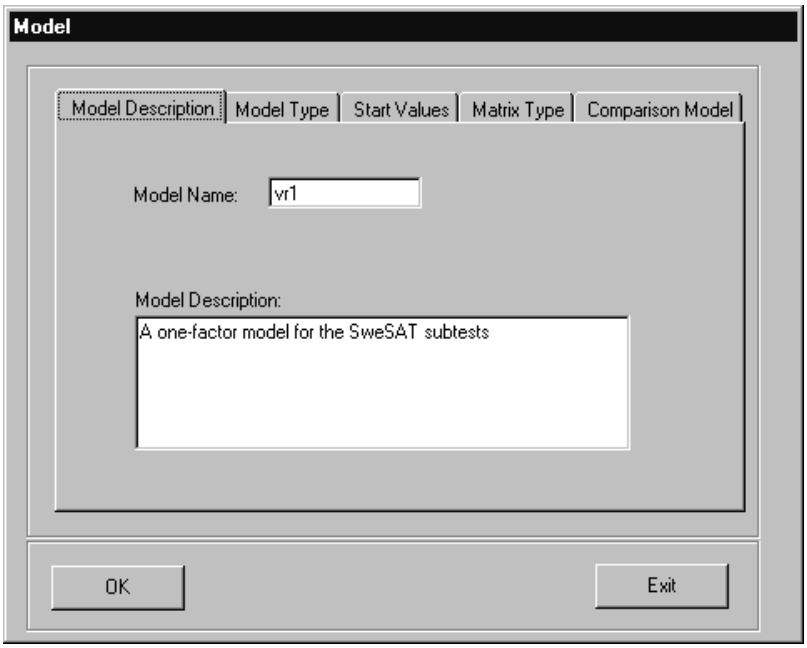
To formulate models which are not supported by MB, the language may be used to specify a model which is as close as possible to the intended one. After the instructions have been generated, they may be edited into the final model.

Constructing the MB instructions

As has already been described the actual production of the MB statements is typically done by STREAMS, and it has also been described how the user directs the process through interaction with an intelligent editor and/or through drawing path diagrams. Here more information is given about how to use the forms of the graphical user interface.

The Model form

When the **Model** button on the Modeling toolbar is clicked the *Model* form is presented:



The *Model* form has five tabs:

- **Model Description**
- **Model Type**
- **Start Values**

- **Matrix Type**
- **Comparison Model**

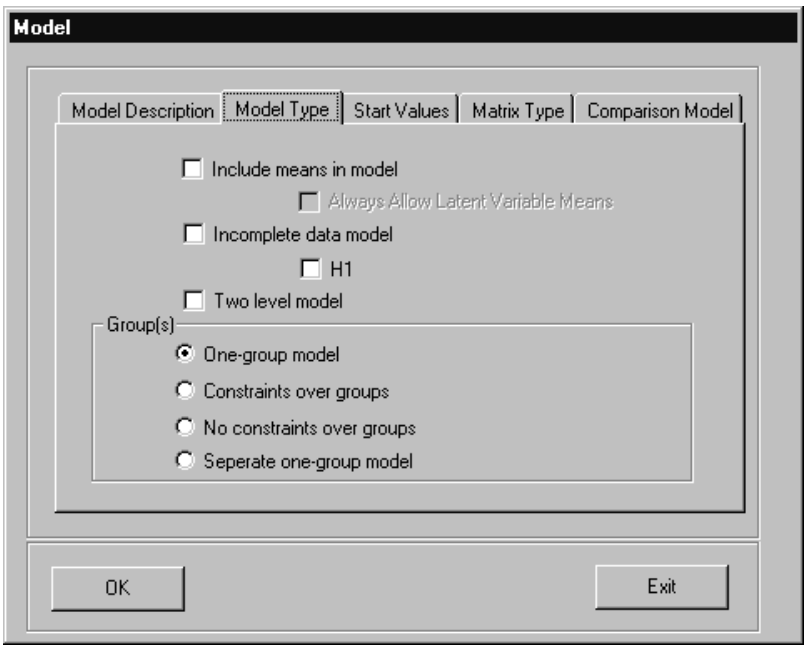
Model Description

The **Model Description** tab allows the user to give a name to a new model by simply writing the name into the **Model Name** field (1-8 characters). If an old model is edited, a name that has previously been entered into this field may be changed. This corresponds to the **Save Model as ...** function under the **Model** menu.

A description of the model must also be provided in the **Model Description** field. This information is used by STREAMS to construct the TI statements.

Model Type

When the **Model Type** tab is selected the *Model* form presents options concerning the basic structure of the model.



There is a choice of including or not including means on manifest and latent variables in the model. The default is that means will be included if means are available in the project.

There is also an option labelled **Always Allow Latent Variable Means**. When this option is not checked, which is the default, the program prevents estimation of models with latent variable means in one-group models. This is because in most cases such models are not identified. However, in some types of models, such as growth curve models (e. g., Willett & Sayer, 1994), latent variable means must be estimated, so for this type of models this option must be used.

In a one-group model inclusion of means does not affect the results in any way if constraints of equality are not imposed over the means, because the means of the manifest variables are treated as free parameters to be estimated, and there are as many parameters

as there are manifest variables. If the model in a later step is to be developed into a multiple-group model these estimates may, however, be useful as a source of start values. When the model comprises multiple groups different kinds of models are created depending on whether means are included or not. The decision whether to include means or not must thus be made on the basis of the nature of the substantive problem that is being studied.

This form also offers check-boxes for identifying an **Incomplete Data Model**, **Two Level Model** and **H1 model**. These options may be used to specify certain types of models for complex observational data. Detailed information about these advanced models is given in Chapters 5 and 6.

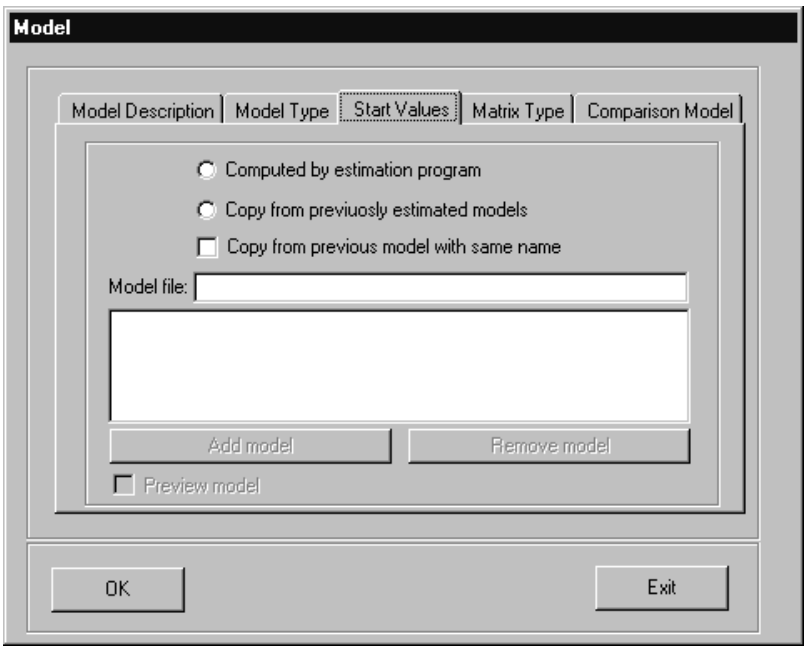
In the **Group(s)** frame four options are offered, three of which are of any relevance only when a multiple group model is to be specifed:

- **One-group model**. The model comprises one group of cases only.
- **Constraints over Groups**. This option implies that by default parameters will be constrained to be equal over groups. Thus if no explicit reference is made to group labels in REL, VAR, COV or MEA statements, equivalence restrictions will be imposed.
- **No Constraints over Groups**. When this option is selected the default is to have no constraints of equality over groups. To constrain parameters over all groups, or over a subset of groups, the group labels must be explicitly listed in the MB statements and enclosed in parentheses.
- **Separate One-Group Models**. This option implies that as many one-group models will be generated as there are groups, and there are, of course, no constraints over groups. This option is available only when LISREL is used as the estimation program.

When an **Incomplete Data Model** or a **Model for Two Level Data** is fitted the model always includes more than one group of cases, even though they may represent one population only. However, such models may be fitted to multiple populations as well, so when such models are requested the term **population** is used instead of the term **group** (e. g., **Constraints over Populations**).

Start Values

When the **Start Values** tab is selected the *Model* form offers several options about procedures for determining start value for free parameters.



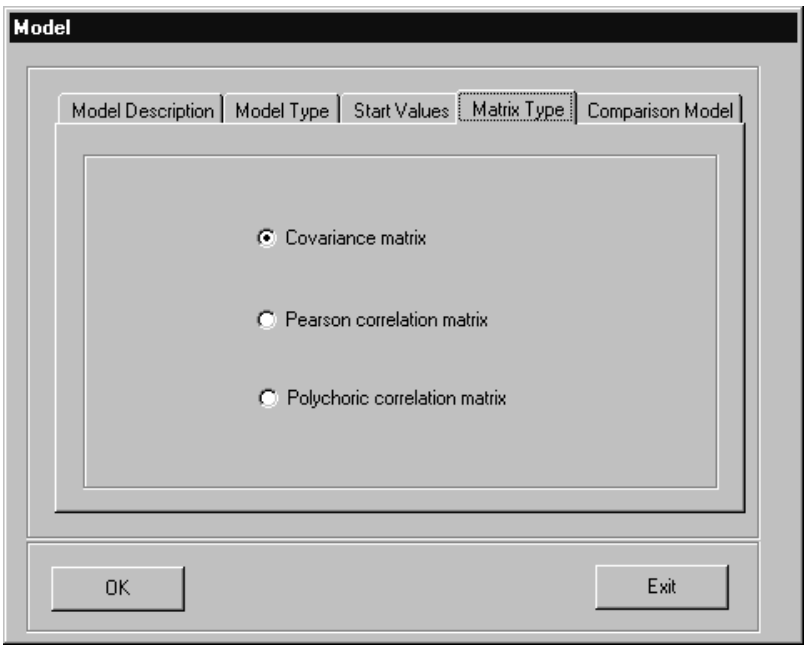
The three options are:

- **Computed by Estimation Program**. This option implies that STREAMS will rely on the procedures in Amos, EQS, LISREL and Mx for computing start values and will make no attempt to copy or guess start values. LISREL uses an elaborate procedure for determining start values (or "initial estimates", as these are called in the LISREL output), and for many models these are very close to the final estimates (see Jöreskog & Sörbom, 1989b, pp. 17-18 for a description of the procedure used to compute the initial estimates).
- **Copy from Previously Estimated Models.** When this option is selected the **Add Model** button is enabled, and when this button is clicked a file open dialogue allows a model to be identified. One or more of models may identified this way. On the basis of names of manifest and latent variables STREAMS determines which sections of the previous model overlap with the current model, and copies start values accordingly. When manifest and/or latent variables are added, or new relations are introduced, start values will of course not be available in the previous model and in these cases STREAMS guesses start values (at present the guessed value is .7 for all parameters except for covariances which are guessed to be .1). If no estimates are available STREAMS will rely on the estimation program's start value procedures instead as described below (see Chapter 7 for an extended discussion about how to run STREAMS optimally).
- **Copy from Previous Model with Same Name**. When this option is selected, which is done through clicking the check-box, an instruction is generated to take start values from a previous version of the current model. This instruction may be used in combination with the other two. If **Copy from Previously Estimated Models** and **Copy from Previous Model with Same Name** are both used, the order of the statements is such that start values from the current model, when available, will override start values from other models. It is recommended that this option is

regularly used. Sometimes, however, a previous model may have generated parameter estimates which will make it difficult to find the correct solution, and in this case this option must be disabled. Because this may happen it is important that a new model name is chosen now and then, and a reference to the previous model is included.

Matrix Type

When the **Matrix Type** tab is clicked, three options concerning the type of matrix to be analyzed is presented: **Covariance Matrix**, **Pearson Correlation Matrix**, or **Polychoric Correlation Matrix** are offered.



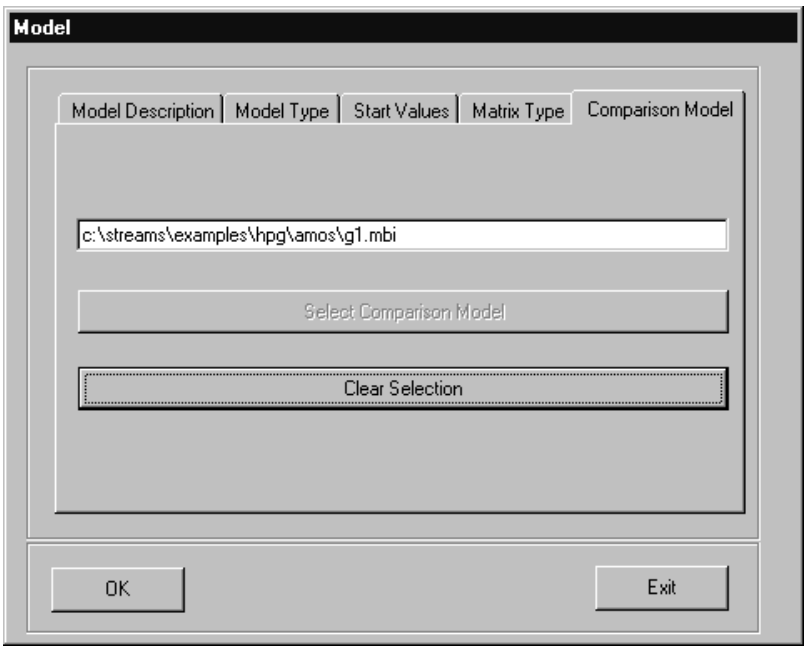
The default is **Covariance Matrix**.

These matrices and the implications for estimation of choice of the different matrices is discussed at length by Jöreskog & Sörbom (1989a, 1993b), so this discussion will not be repeated here. It must be emphasized, though, that different estimators are differentially appropriate for the different types of matrices.

With maximum-likelihood estimation the covariance matrix is the recommended choice, and when a polychoric correlation matrix is used Generally Weighted Least squares estimation is recommended. In the latter case a weight matrix must be available as well (see Chapter 10).

Comparison Model

The **Comparison Model** tab may be used to select a model with which the goodness-of-fit of the model to be estimated will be compared.



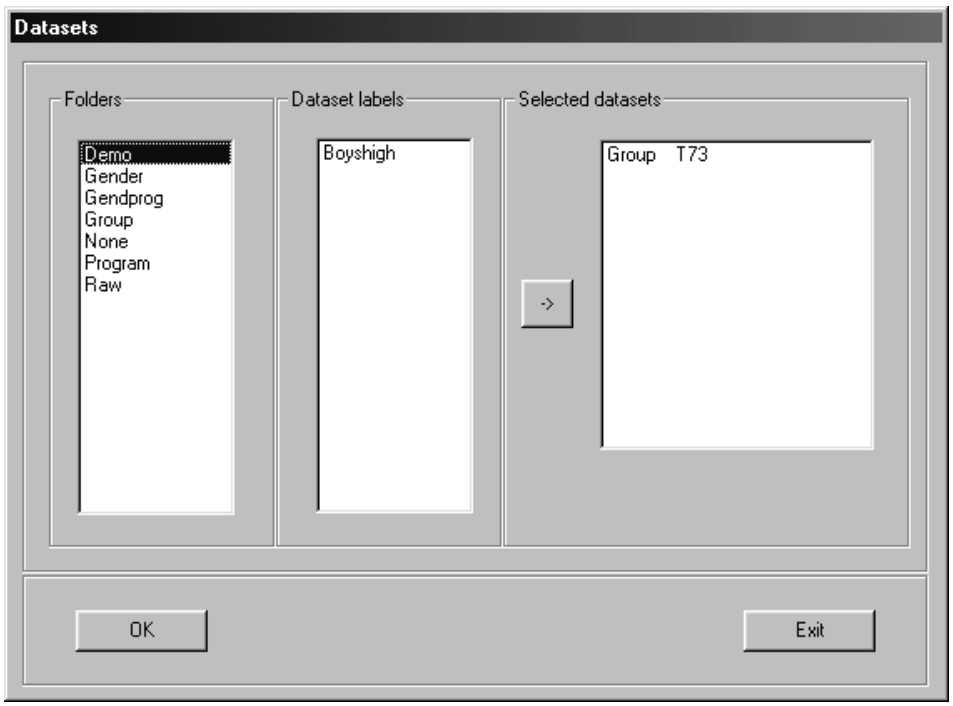
Use of a comparison model is of the greatest value when missing-data models are estimated (see Chapter 5). However, a comparison model may also be used when ordinary models are fitted, in order to test if there is a significant difference between a more constrained and a less constrained model.

The Options form

When the **OK** button is clicked on the *Model* form a set of MB instructions is written into the editor area if MB mode is used. If PD mode is used the same statements are constructed, but are not displayed. One of the statements is the OP statement which specifies options for the selected estimation program. Thus, if LISREL is used the OP statement includes a standard LISREL OU statement (Jöreskog & Sörbom, 1993b). When the program is run this statement is transferred to LISREL, with some further information added to it. It is possible to change the default OU statement, however. One way to do this is to double-click on the OP statement, and another way is to press the **Options** button. Both these methods bring forward the *Options* form which is appropriate for the selected estimation program: *Amos Options*, *EQS Options*, *LISREL 8 Options*, *Mplus Options* or *Mx Options*. Detailed information about these options is given in Chapter 13 “Estimation Program Interfaces” on page 211.

The Datasets form

The groups to be included in the model are identified with the *Datasets* form. No default is available, so a choice has to be made. This is done through clicking the **DAT** button. When this is done the *Datasets* form is shown.



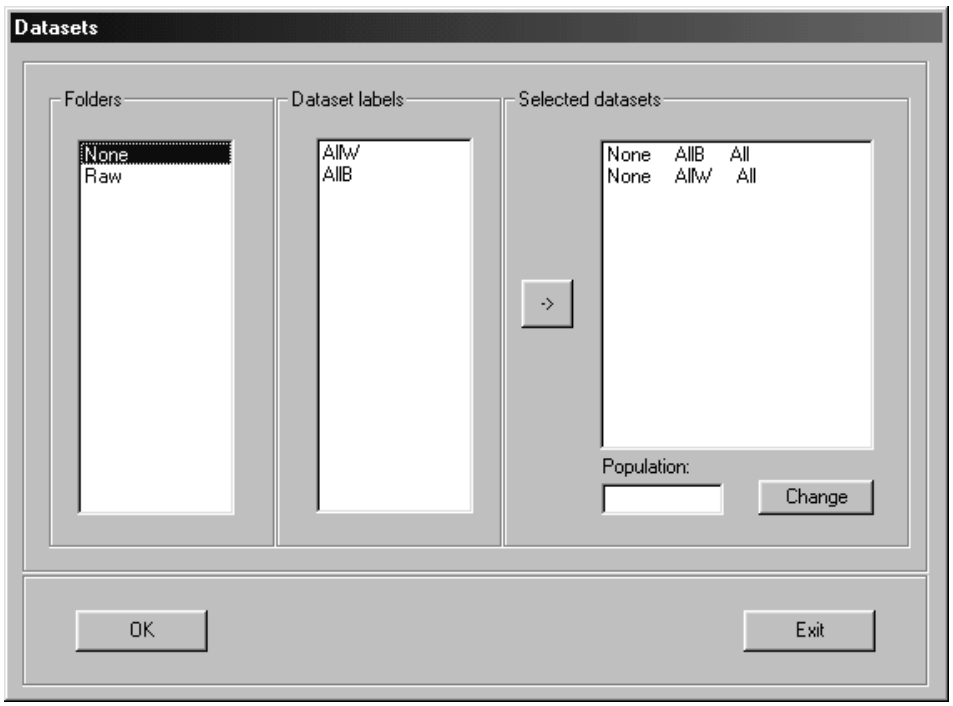
The list-box on the left-hand side presents the folders that have been defined for the project, and when one of these is selected the dataset labels which have been defined for this folder are displayed. To identify a dataset both the folder and the dataset label must be selected. When this has been done for one or more datasets, the button marked with an arrow may be clicked, which causes the selected datasets to be moved to the list-box on the right-hand side. Datasets may also be deselected through moving them back again.

When the **OK** button is clicked DAT statements corresponding to the selections made are constructed. For example, if the folder Gender has been chosen and the two datasets with labels Males and Females have been selected, STREAMS inserts the following two statements into the edit area:

```
DAT FOLDER=Gender DATLAB=Males
DAT FOLDER=Gender DATLAB=Females
```

If we want to add or remove datasets the DAT button may be clicked at any time to retrieve the *Datasets* form. Double-clicking on a DAT line will also produce this form.

When a two-level model or an incomplete data model has been requested the *Datasets* form also offers facilities for assigning datasets to populations.

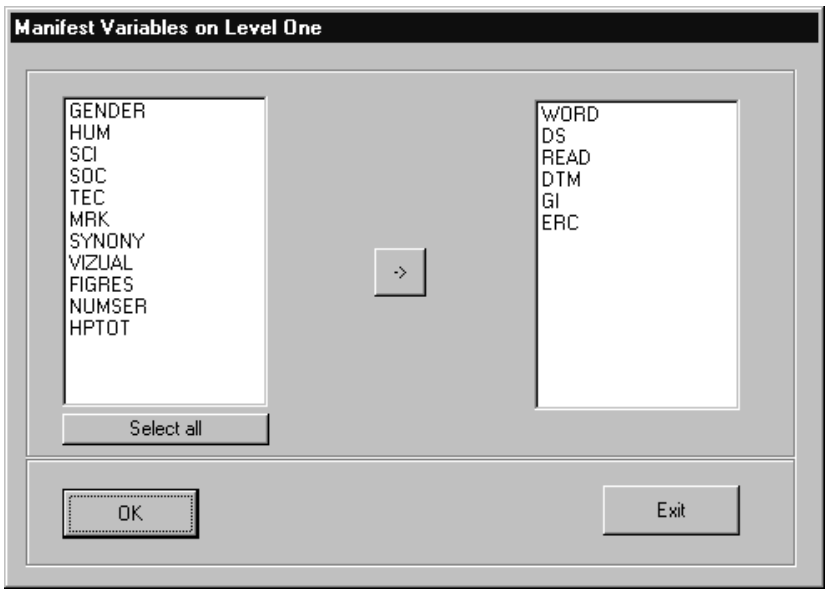


For each group there is a default classification into populations. For two-level models this is done on the basis of the common prefix of the dataset label, and for incomplete data models it is done on the basis of the folder name.

If, however, the default classification is not the correct one the population assignment is easily changed: Select one or more datasets from the **Selected datasets** list, enter a population label in the **Population** field, and click the **Change** button. This process is then repeated until all datasets have been assigned to the correct population.

The Manifest Variables form

The MB language requires that all manifest variables are declared, which is done with the MVR statement. To select all variables or a subset of variables from the project the **MVR** button is clicked. When this is done the *Manifest Variables* form is shown.



Variables are identified through selecting one or more variables in the list on the left-hand side and when the arrow button is clicked these variables are moved to the list on the right-hand side. This process may be repeated any number of times, and the same procedure may also be used to move variables from the right-hand side to the left-hand side.

When the **OK** button is clicked one or more MVR statements are put into the edit area by STREAMS. For example, if the variables WORD, DS, READ, DTM, GI and ERC were selected the MVR statement would be:

```
MVR WORD DS READ DTM GI ERC
```

If there are more variable lables than may be fit into one line, STREAMS uses more than one statement. To change the selection of variables the MVR button may be clicked again, or an MVR statement may be double-clicked.

When the option **Autoremove variables** on the **General** tab of the *Preferences* menu is checked variables which are removed from an MVR statement will also be removed from all the MB statements.

The Latent Variables form

The latent variables must be declared as well, and because these are unknown to STREAMS labels of the latent variables must be supplied. This is done on the *Latent Variables* form, which is presented when the **LVR** button is clicked.

Labels of the latent variables are entered in the top white field, and then the **ADD** button is clicked, which moves the new label to the list of latent variables. This process is repeated as many times as there are labels to be entered. To delete an already entered label, select it in the list and click the **Delete** button. When the list contains the labels for the latent variables to be included in the model, click the **OK** button. This will cause STREAMS to construct an LVR statement, and add it to the edit area.

If, for example, the latent variables Verb and Reas are entered, STREAMS inserts the line:

```
LVR Verb Reas
```

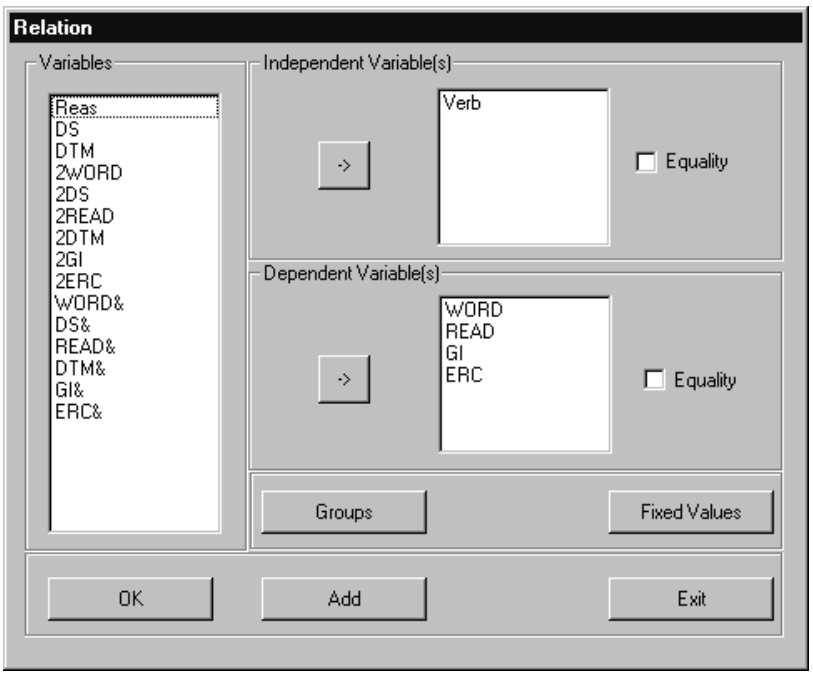
Latent variables may be added and deleted at any time, through double-clicking on the LVR statement, or through clicking the **LVR** button.

When the option **Autoremove variables** on the **General** tab of the *Preferences* menu is checked variables which are removed from an LVR statement will also be removed from all the MB statements in which they appear.

The Relations form

When groups have been selected, and the manifest and latent variables have been declared, the actual model specification may be started. The Model Building toolbar offers a set of buttons (REL, VAR, COV, MEA and SCL) some or all of which may be used in the process of model building. The buttons may be used both in MB mode and in PD mode.

Most models involve one or more relations, and to specify these the *Relations* form is used. Clicking on the **REL** button causes this form to be shown.

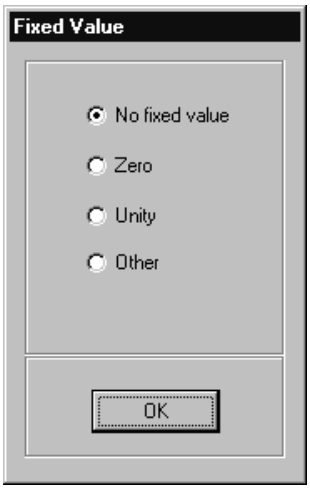


This form displays the variables available for modeling in the **Variables** list on the left hand side. This list includes the declared manifest and latent variables, along with residual variables for those manifest and latent variables that have previously been defined as dependent variables. The residual variables have the same label as the dependent varibles, but with an ampersand (&) added to the label.

The form also presents one list-box for independent variables, and one list-box for dependent variables. Initially these are empty (unless the *Relations* form has been opened by double-clicking on a REL statement) but variables in the **Variables** list may be moved to either of these list boxes. This is done through selecting one or more variables in the list using the standard techniques, and then clicking the appropriate button with an arrow. The same technique may be used to move variables from the lists of independent or dependent variables to the **Variables** list.

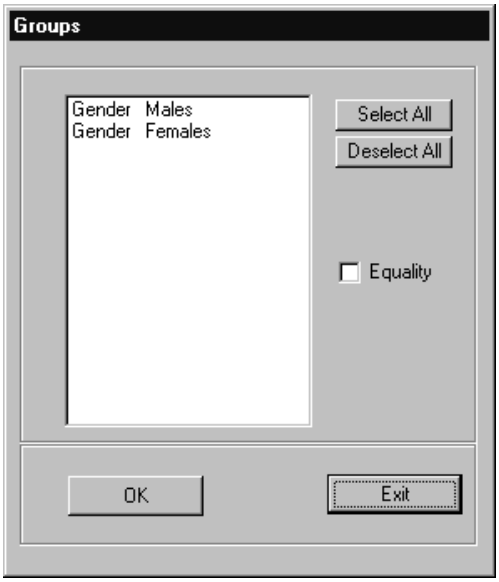
Next to the list boxes for independent and dependent variables are check boxes labeled **Equality**. When these are checked (through clicking) equality constraints are imposed for the marked category of variables as has previously been described.

The *Relations* form also offers two buttons: **Groups** and **Fixed Value**. The **Fixed Value** button is used to assign a common fixed value to all the relations defined by the REL statement. When this button is clicked the *Fixed Value* form is shown:



This form offers the options **Zero**, **Unity** and a user-specified value. If the **Other** option is chosen any number may be entered in the field. Clicking **OK** causes the *Relations* form to appear again.

The **Groups** button causes the *Groups* form to be shown.



This form is used to select one or more groups for which the REL statement is to apply. The default is that the REL statement applies to all the groups identified through GRP statements. However, even when all groups are to be included it is often useful to mention the groups explicitly in the REL statement. This is effected either through the check box labeled **Select All** or through identifying all groups in the list as selected. To select a sub-set of groups these are high-lighted in the list. When the **OK** button is clicked the *Relations* form appears again.

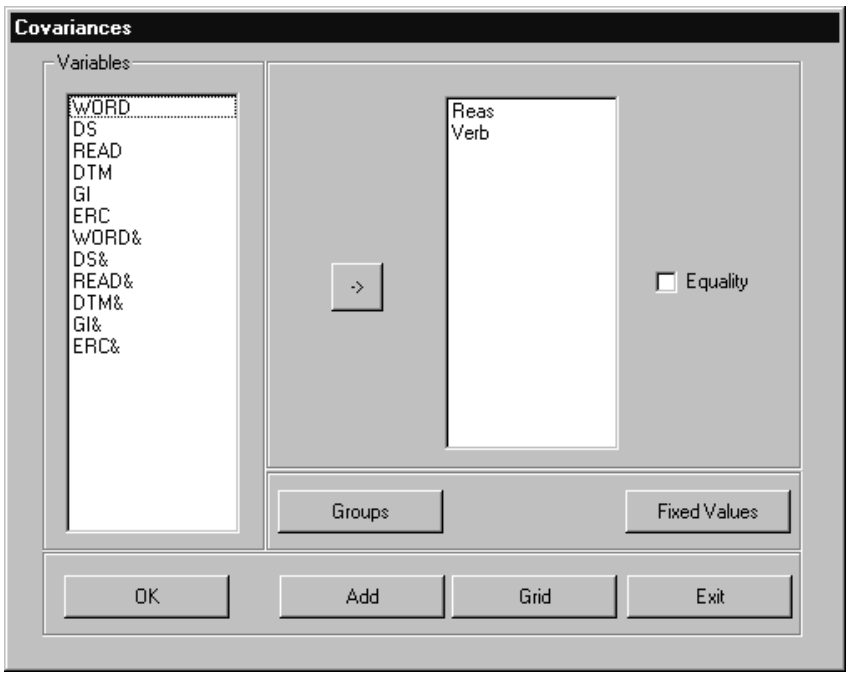
When the relations have been specified as desired the **OK** button or the **Add** button on the *Relations* form is clicked. This causes STREAMS to contruct a REL statement which is inserted with the other statements in the edit area. When **OK** is clicked the *Relations* form is closed, but when **Add** is clicked it stays open. An example is:

```
REL (Males Females) Verb -> WORD READ GI ERC
```

In order to change an existing REL statement it may be double-clicked, or the button **Edit Line** may be clicked, with the insertion point on the line to be edited. To introduce another REL statement the **REL** button should be clicked again.

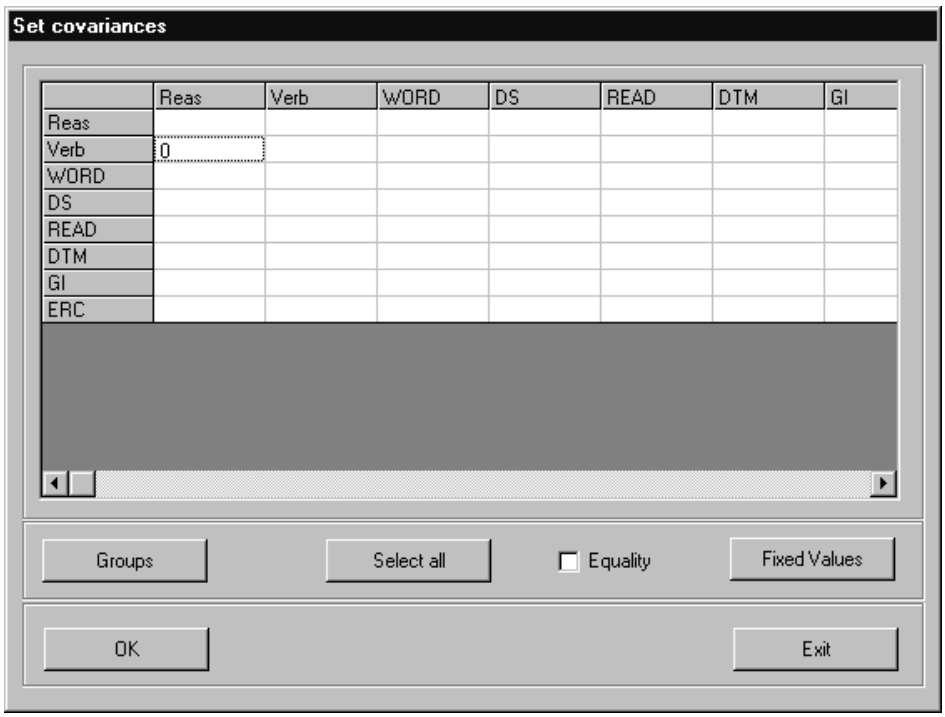
The Covariances form

Covariances among manifest variables, latent variables, residuals in latent variables, or among errors in manifest variables are essential aspects of many structural equation models. When the **COV** button is clicked the *Covariances* form is shown:



This form is used to select a set of variables among which covariances are estimated. In the Variables list box on the left hand side the same set of variables as was described in connection with the *Relations* form are shown. Using the procedure described above two or more of these variables may be moved to the empty list box on the right hand side. The selected variables will be included in a COV statement which is constructed when the **OK** button, or the **Add** button (which leaves the form open), is clicked.

The *Covariances* form also includes a button labeled **Grid**. When this button is clicked, with some or all of the variables in the **Variables** list selected, the *Set Covariances f*orm is shown:



A grid is presented with the selected variables as rows and columns. The cells of the matrix may be clicked, in which case a 0 is entered. Each cell marked in this way will generate a COV statement when the **OK** buttons on the *Set covariances* form and the *Covariances* form are clicked.

The *Covariances* and the *Set covariances* forms also offer **Group** and **Fixed Value** buttons. These call up the *Group* and *Fixed Value* forms described above, and these forms have the same function here.

An example of a COV statement produced by STREAMS may be:

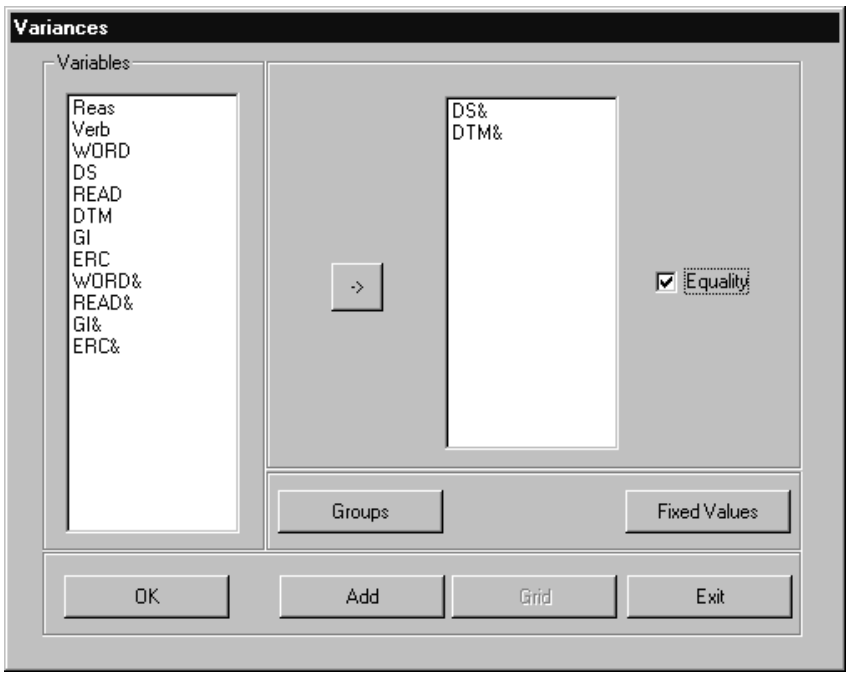
```
COV Verb Reas
```

This statement implies that the two latent variables *Verb* and *Reas* are allowed a covariance, because STREAMS by default assumes all variables to be uncorrelated.

The Variances form

Every independent variable introduced in a REL statement is in STREAMS assumed to have a variance which is estimated as a free parameter, and every dependent variable is assigned a residual, which also has a variance. The VAR statement may thus often be excluded from the MB instructions.

However, there also often are situations in which the VAR statement must be included, because variances are to be constrained to be equal over groups or variables, or because they are to be assigned a fixed value. The *Variances* form, which is presented when the **VAR** button is clicked, is used to accomplish these tasks. It looks exactly like the *Covariances* form, and it is used in the same way.



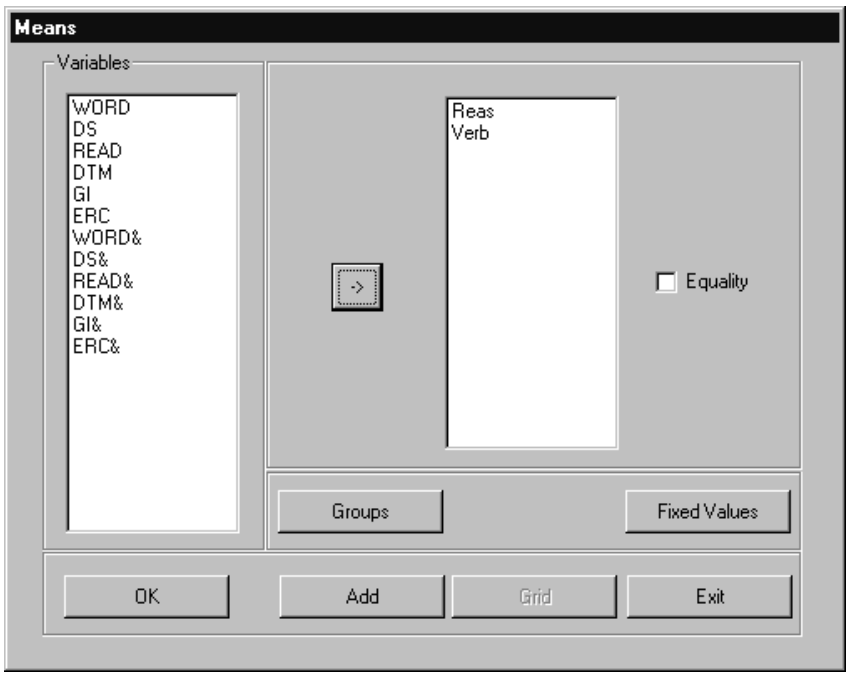
If, for example, we want to constrain the variances of the errors in the manifest variables DS and DTM to be equal, this is accomplished with the statement:

```
VAR (DS& DTM&)
```

The Means form

Means on latent variables are by default assumed to have a mean of zero in all groups when general constraints of equality are imposed over groups. When no constraints of equality are imposed the means on latent variables are fixed to zero in the first group, and are left free to be estimated in the other groups. Means on residuals in latent and manifest variables are by default assumed to be zero in all groups. In a one-group analysis means on manifest variables are by default left free to be estimated, and in multiple-group analyses constraints of equality are imposed on the means of manifest variables.

Statements about means are made with the *Means* form, which is presented when the **MEA** button is clicked. This form has the same appearance as have the *Covariances* and *Variances* forms, and it is operated in the same way as these:

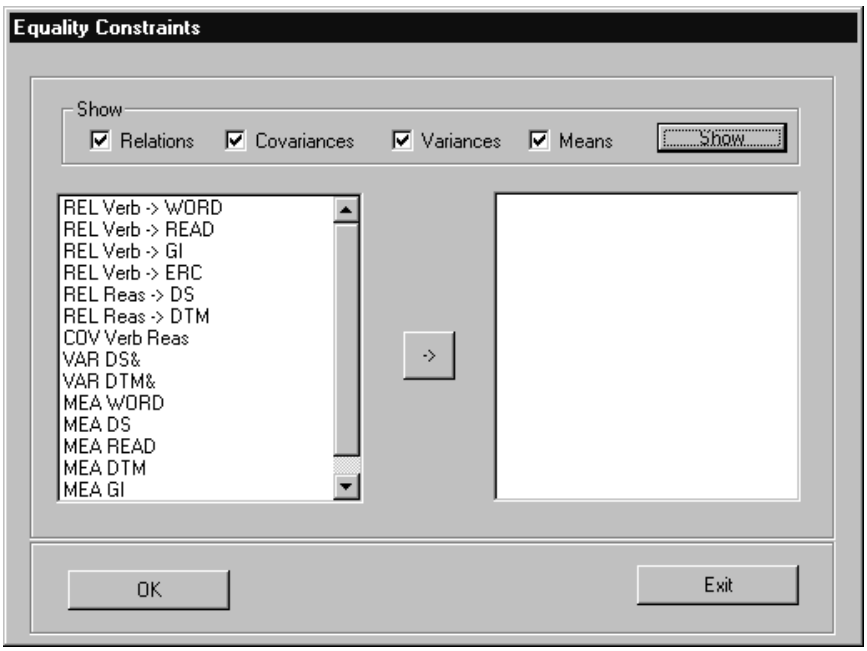


For example, to set the means on the latent variables Verb and Reas free for the group of females the following command would be issed:

```
MEA Females Verb Reas
```

The Equality Constraints Form

As has already been emphasized the EQ command needs only rarely be used, because most of the time the options for equality constraints available in all statements may be used. But in some cases a more general equality constraining statement is needed, which is obtained with the *Equality Constraints* form.

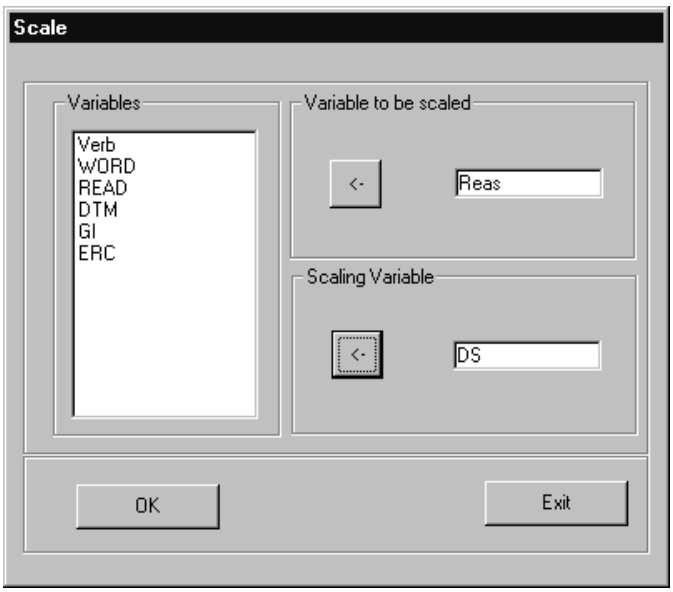


The list-box to the left shows all, or a subset of, the free parameters in the model expressed in terms of an MB statement for one parameter. The check-boxes at the top of the form may be used to restrict the list of statements to comprise only certain categories of statements (relations, covariances, variances and means). The parameters for which the equality constraints are to apply are selected using the ordinary techniques and moved to the list-box to the right. From this information one or more EQ statements are generated when the **OK** button is clicked. Equality constraints for two parameters requires one EQ statement, constraints for three parameters requires two EQ statements, and so on. An example could be:

```
/* Start EQ Block
EQ REL Gsc -> SDQGSC AND REL Esc -> SDQESC
EQ REL Esc -> SDQESC AND REL Msc -> SDQMSC
/* End EQ Block
```

The Scale Form

STREAMS assumes that the first manifest variable to which a latent variable has a relation will be used to set the scale of the latent variable through a fixed relation of unity. In higher-order models the first lower-order factor to which a higher-order factor has a relation is in a similar fashion used to set the scale of the higher-order factor. These choices are not necessarily the best, however, and the Scale form may be used to select another variable for establishing a scale in a latent variable. The *Scale* form is presented when the **SCL** button is clicked:



The variable list is presented in a list-box, and the form allows selection of one **Variable to be scaled** (i. e., the latent variable) and one **Scaling variable** (i. e., the manifest variable or another latent variable). For example, to use READ instead of WORD as the scaling variable for the latent variable Verb the following statement would be issued:

```
SCL Verb READ
```

13 Estimation Program Interfaces

The present chapter provides some further information about the SEM estimation programs with which STREAMS communicates, and describes characteristics of the interface.

Languages for Structural Equation Modeling

Several excellent computer programs are available to specify, estimate, and test structural equation models, such as Amos (Arbuckle, 1997), EQS (Bentler, 1995), LISREL (LInear Structural RELations; Jöreskog & Sörbom, 1989a, b; 1996b, c), Mplus (Muthén & Muthén, 1998) and Mx (Neale, 1995). LISREL offers two languages for model specification. One is the LISREL language (Jöreskog & Sörbom, 1989), and the other is a language with a simplified syntax called SIMPLIS (Jöreskog & Sörbom, 1996c).

Current releases of all SEM program offer a "path diagram" interface, with which models may be specified and respecified, and results are displayed. However, the diagram representation is translated into a text-based language, which is interpreted by the computer. The different programs have their own languages, the characteristics of which are important for what kinds of models may be formulated, and also for the amount of effort needed to specify a model. Because the path diagram interfaces are of limited utility in many situtations, and particularly so when large and complex models are specified, the SEM user needs to learn at least one language.

The LISREL language

The LISREL language offers a set of 13 parameter matrices which are used to specify relations between variables of different kinds (e. g., latent independent variables and manifest independent variables, latent dependent variables and manifest dependent variables; and latent dependent and latent independent variables) and variances and covariances for

different categories of variables (e. g., covariance matrices for latent independent variabbles, and for residuals in manifest and latent variables). This languge allows models to be created with great freedom and precision. The resulting models also tend to be efficiently estimated by the LISREL program. It is, however, a somewhat tedious and error-prone task to specify a large model in the LISREL language, which partly is because variables are refered to by numbers only.

The SIMPLIS language

The SIMPLIS language allows simple specification of relations between observed and latent variables, and statements may be made in a free form about variances and covariances of independent variables. Statements are made in terms of labels of latent and manifest variables, so in most cases the model specification is extremely simple. The SIMPLIS language is somewhat limited, however, because it does not allow relations which involve residuals. Specification of multi-group models also is somewhat tedious. SIMPLIS is not supported by the current version of STREAMS.

The LISREL 8 program translates the SIMPLIS statements into a sequence of statements in the LISREL language. In this way an efficient model specification is obtained without the user having to bother with the complicated details of the LISREL languge.

The EQS Language

The EQS program also offers a relatively simple language with which models may be formulated as a set of equations in terms of the variables. EQS has a strong affinity to the general RAM (Reticular Analysis of Moment structures) framework described by McArdle and McDonald (1984). In contrast to SIMPLIS, EQS makes the residuals in manifest and latent variables available as independent variables. One disadvantage of EQS, however, is that start values for the iterative solution of the equations must be supplied by the user, while these are computed by Amos and LISREL. As in SIMPLIS the specification of models for multiple groups is done through one specification for each group, and through statements which specify constraints across groups.

The Amos Language

The Amos language, which is referred to as Amos Text by Arbuckle (1997), also is based upon the RAM specification. Amos Text, which is available in Amos 3.6, is a simple and general language, which offers the same degree of generality as EQS. However, unlike EQS, Amos does not require specification of start values. The Amos language offers an elegant method for imposing equality constraints on parameters, through assigning the same label to parameters which are constrained to be equal. This feature makes it relatively easy to specify multi-group models with Amos. With Amos 4 the Amos Text language has been replaced with Amos Basic, which has similar structure, but which is more general and powerful because it has the capabilities of a programming langtuage.

The Mplus Language

The Mplus language also is a simple and general language, which is somewhat similar to Amos text. Thus, a smilar method for imposing equality constraints on parameters is used in Mplus as in Amos. The Mplus language also offers built-in support for specification of

two-level models and for several other complex type of models. The Mplus program has limited capabilities for computing start values, so these must often be supplied in the language statements. In the Mplus language no clear distinction is made between residuals and other variables, which implies some restrictions on the generality of the type of models which may be formulated (for an example, see page 71).

The Mx Language

The Mx program (Neale, 1995) is a structural equation modeling package, which offers almost complete generality and flexibility. The program is based on a matrix algebra processor, and a general optimizer, and these tools make it possible to specify and estimate all standard SEM models, and also complex non-standard models, in terms of matrix algebra expressions. The flexibility and power of Mx is, however, bought at the price of a relatively high level of complexity, which makes the Mx language inaccessible to many substantively oriented SEM users.

Advantages and Disadvantages of SEM Programs

There are both advantages and disadvantages associated with each of the systems. Nevertheless, modelers tend to commit themselves to use of one program or language only, which probably has to do with the fact that it is quite tedious and confusing to prepare data, and to specify and respecify models for different systems. But with STREAMS it is easy to change from one system to another, which makes it quite feasible to use more than one program in the same modeling project. This makes it possible to exploit the complementary strengths of the programs. In many cases these advantages probably are so great that users who only have access to one program should acquire one or more of the other programs as well.

For example, LISREL 8 offers excellent functions for determining start values for the iterative solution of the system of non-linear equations involved in the estimation process and it generally is fast and efficient. An EQS or Mplus model may be supplied with start values which have been determined from parameter estimates computed by LISREL in a previous modeling step (or vice versa). EQS and Mplus, on the other hand, offers estimators, and statistics for evaluating goodness of fit which are not available in Amos or LISREL. As is demonstrated in Chapter 7 the programs are differentially effective for different kinds of models, EQS being very good at one-group models, but quite inefficient when estimating multiple-group models. Other examples of advantages and disadvantages of the estimation programs could easily be cited, but it should also be stressed that user experiences and judgement differ greatly (see, e. g., Byrne, 1995, for a comparison between EQS and LISREL).

Types of Models Supported for Different SEM Programs

There are certain restrictions on which types of models are supported for different lan-

guages. The table below presents an overview:

TABLE 3. **Types of models supported for different languages**

	LANGUAGE				
Model Type	**Amos**	**EQS**	**Mplus**	**Mx**	**LISREL**
One-group model (pre- and post-processing)	X	X	X	X	X
Multiple-group model (pre-processing)	X	X	X	X	X
Multiple-group model (post-processing)	X		X	X	X
Models with residuals as independent variables	X	X		X	X
Models for incomplete data	X		X	X	X
Models for two-level data	X		X	X	X

All the different model types are supported when Amos, LISREL or Mx are used. However, because EQS does not write estimation output to files when multiple group models are estimated it is not possible to support post-processing of output from such models. Models for incomplete data and two-level models are multiple-group models, so such model types cannot be supported for EQS.

For Mplus models with residuals as independent variables are not supported, because the Mplus language does not make a distinction between residuals and other kinds of variables. There also are some restrictions on the types of two-level models supported (see “Specifying Models for Two-Level Data”, page 99), and only a subset of the advanced features offered by Mplus are supported by STREAMS.

The Options Forms

For each SEM program there is an *Options* form. These are described below.

The Amos Options form

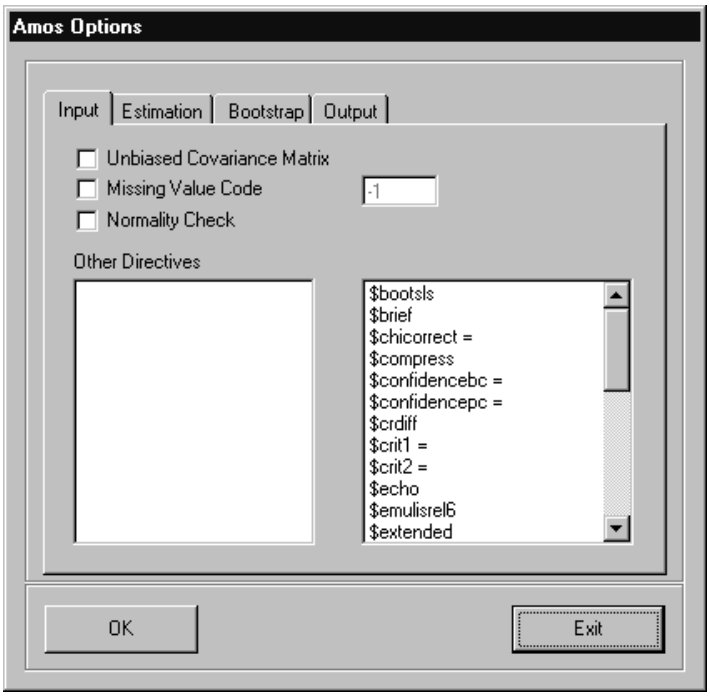
This form offers four main categories of options on four tabs:

- **Input**
- **Estimation**
- **Bootstrap**
- **Output**

The Amos (Text) program offers a very large number of options, which are specified through directives, the first character of which is a dollar sign. Most of these directives may be specified through prespecified choices on the *Amos Options* form, but some are specified through selecting directives from the list supplied in the **Input** category.

Input

The **Input** category offers check boxes to select some options concerning the nature of the data.

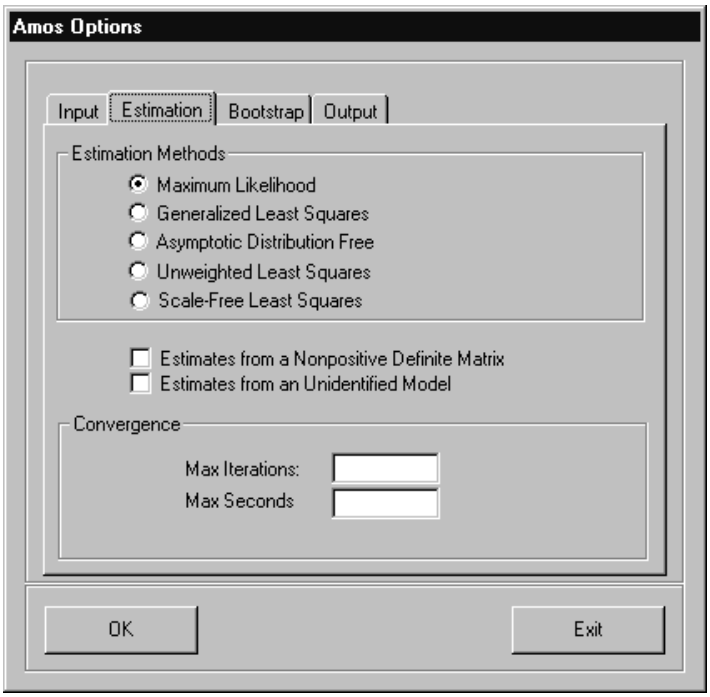


The **Unbiased Covariance Matrix** option generates the *$unbiased* directive, which should be selected when a covariance matrix has been computed with N-1 as the number of observations. STREAMS computes such matrices, so this option should be used when matrices computed by STREAMS are analyzed.

When the **Missing Value Code** option is checked a value (default -1) may be supplied which indicates a missing value in raw data. The same missing data code must thus be used for all variables. When this option is selected Amos uses a special missing-data estimation technique (see Chapter 5 for further information). The **Normality Check** option may also be used with rawdata as input, and causes a test of multivariate normality to be computed.

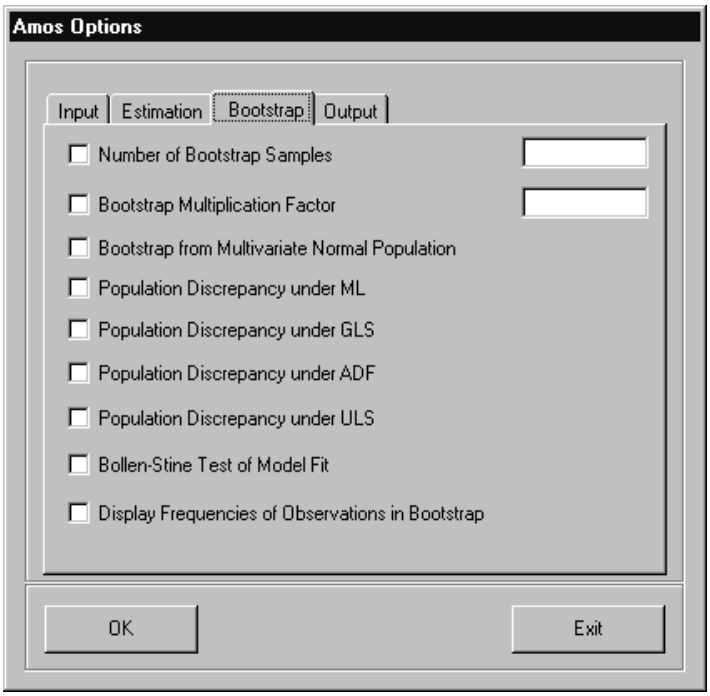
The **Input** tab also offers a list of directives, which may not be selected in any other way. Clicking one of these causes it to be moved into the box labeled **Other Directives**. Some of the directives must also be supplied with a numerical parameter, which is simply done through writing the number. To remove a directive from the list, the ordinary text editing techniques are used.

Estimation



The **Estimation** tab allows choice of different estimation methods, and of some other options which govern the process of estimation. It is also possible to restrict the number of iterations and the time for estimation; the default is that there are no such restrictions.

Bootstrap

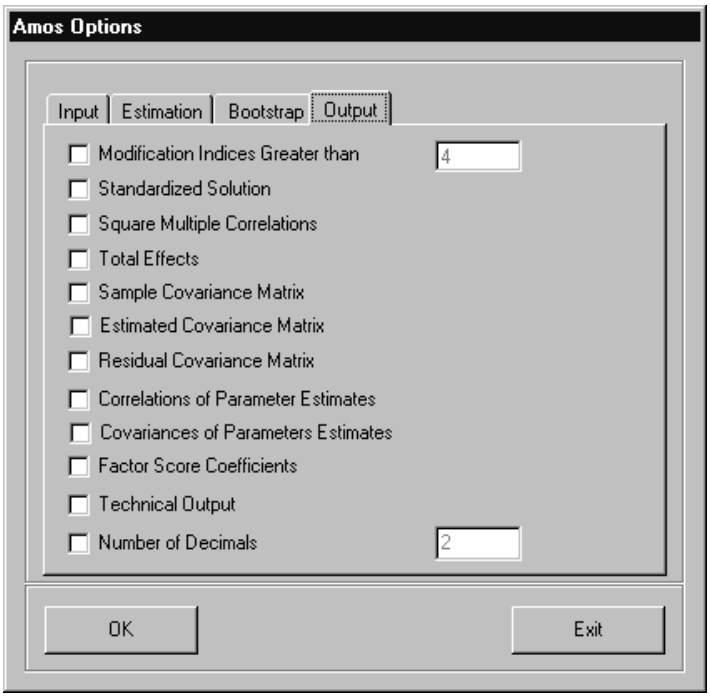


The **Bootstrap** tab is used to ask Amos to perform boostrap estimation of standard errors, and other statistics.

When the check-box labeled Number of Boostrap Samples is clicked, the number of samples may be supplied. When this box is not checked, no bootstrapping is performed.

This form also offers several options concerning the statistics to be generated. For an explanation of the meaning of these options, the Amos manual (Arbuckle, 1997) is referred to.

Output



The **Output** category may be used to select different output to be printed. In most cases the output requested is also written to a file, which has the model name as suffix, and *.amp* as suffix. The options should be self-explanatory and full information is give in the Amos manual (Arbuckle, 1997).

It should be observed, however, that when the **Factor Score Coefficients** option is selected, the post-processor retrieves the factor score weights from the Amos output listing, and constructs the SPSS instructions needed to actually obtain the factor scores. The SPSS syntax is written to a file with the model name as prefix, and *.sps* as suffix, and which is put in the dictionary directory. In order to achieve sufficient precision in the computation of factor scores the number of decimal places is always set to 6 when this option is selected.

If, for example, the **Factor Score Coefficients** option is selected for the *vr1.mbi* model in the HPG project, the following SPSS code is written to the file *vr1.sps*:

SPSS instructions for computation of factor scores constructed by the post-processor

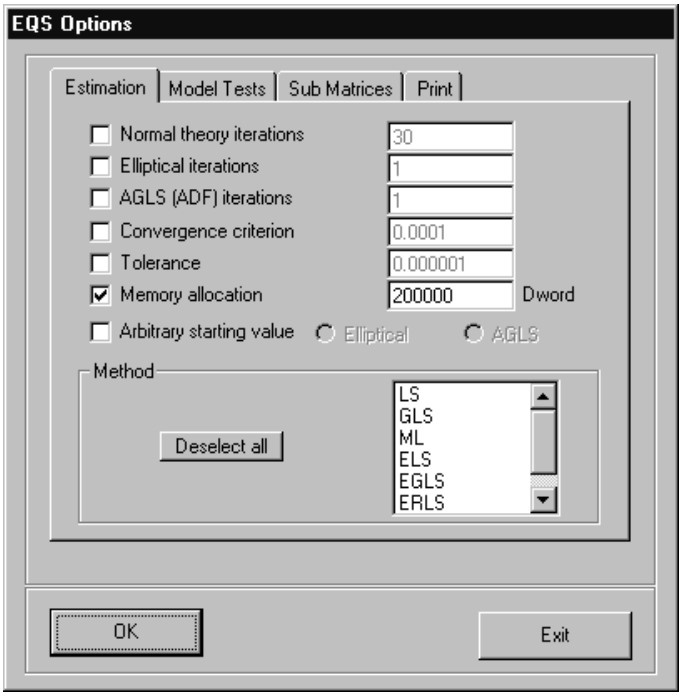
```
COMPUTE Verb = WORD*.37456 +
               READ*.383831 +
                GI*.051307 +
               ERC*.031045 +
                DS*.042173 +
               DTM*.048222 .
COMPUTE Reas = WORD*.06358 +
               READ*.065153 +
                GI*.292412 +
               ERC*.176936 +
                DS*.240358 +
               DTM*.274834 .
EXECUTE.
```

The EQS Options form

This form offers four main categories of options on four tabs:

- **Estimation**
- **Model tests**
- **Sub-matrices**
- **Print**

Estimation



The EQS program offers several estimation methods, more than one of which may be obtained in the same run (see Bentler, 1995, pp. 46-47). The available estimation methods are shown in the list, and through clicking and shift-clicking in the list, up to three estimation methods may be chosen.

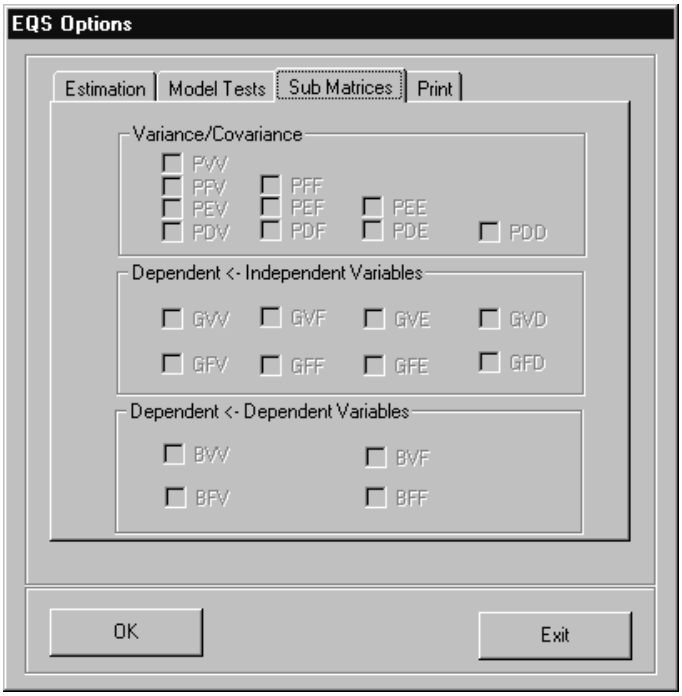
The **Estimation** category tab allows the user to specify different parameters which govern the estimation process. The meaning of these options is described in the EQS manual (Bentler, 1995, Chapter 3).

Model Tests

Under the **Model Tests** tab, the user may select the Wald test, which investigates if one or more free parameters of a given model may be dropped, and the LM (Lagrange Multiplier) test, which investigates if one or more fixed parameters should be treated as free parameters instead. When EQS 5 is used there is also an option labelled **All Fit Statistics**, which by defalt is selected.

Default forms of these tests are obtained through clicking the check-boxes. In EQS it is also possible to specify a large number of alternatives and options for these tests. The EQS instructions for obtaining most of these must, however, be edited into the EQS instructions generated by the pre-processor. However, some of the options concerning the LM test are available in the **Model Tests** category (see Bentler, 1995, for a description of the meaning of these).

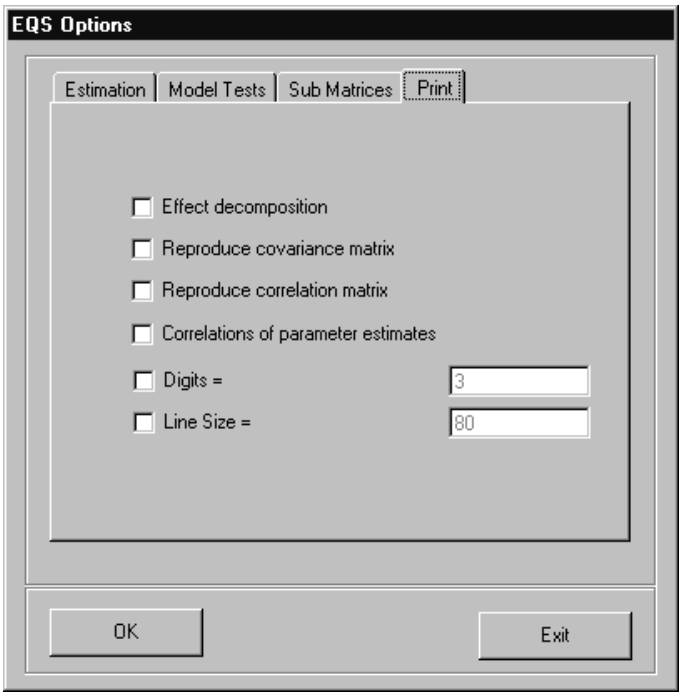
Sub Matrices



The **Sub Matrices** tab allows the user to specify which sub-matrices are to be investigated with the LM test.

The sub-matrices which are to be included in the LM test are checked with the check-boxes.

Print



The **Print** tab, finally, is used to govern the output from EQS.

Most of the options are self-explanatory, and more information may also be obtained from the EQS manual (Bentler, 1995).

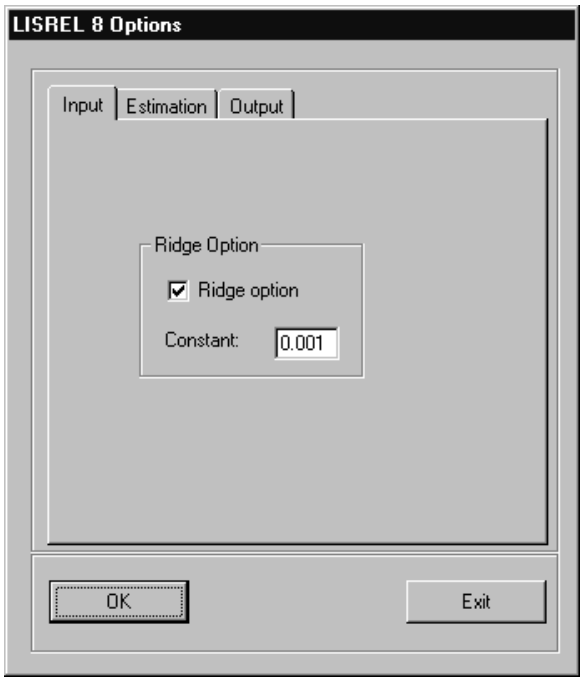
The LISREL 8 Options form

The *LISREL 8 Options* form offers three main categories of options on three tabs:

- **Input**
- **Estimation**

- **Output**

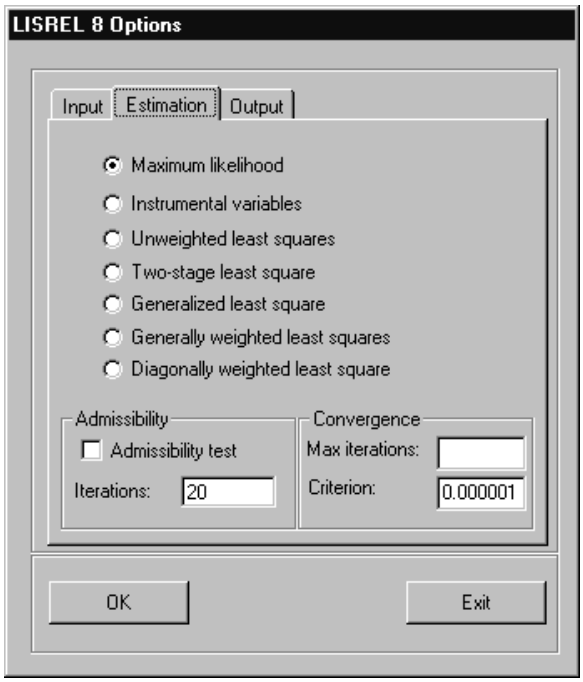
Input



When the **Input** tab is clicked, the following form is presented:

This form offers the user the opportunity to decide whether the Ridge Option (see Jöreskog & Sörbom, 1989b, p. 22) is desired or not, and which constant to apply.

Estimation



The **Estimation** tab offers options governing the estimation of parameters.

This form offers a choice between seven different methods of estimation: Maximum Likelihood (ML), Instrumental Variables (IV), Two-Stage Least Squares (TSLS), Generalized Least Squares (GLS), Generally Weighted Least Squares (WLS), and Diagonally Weighted Least Squares (DWLS).

The estimation methods are described by Jöreskog and Sörbom (1989a, pp. 16-22). As has already been pointed out choice of estimation method should be matched with choice of matrix to be analyzed, and the matrix should, of course, be appropriate for the data.

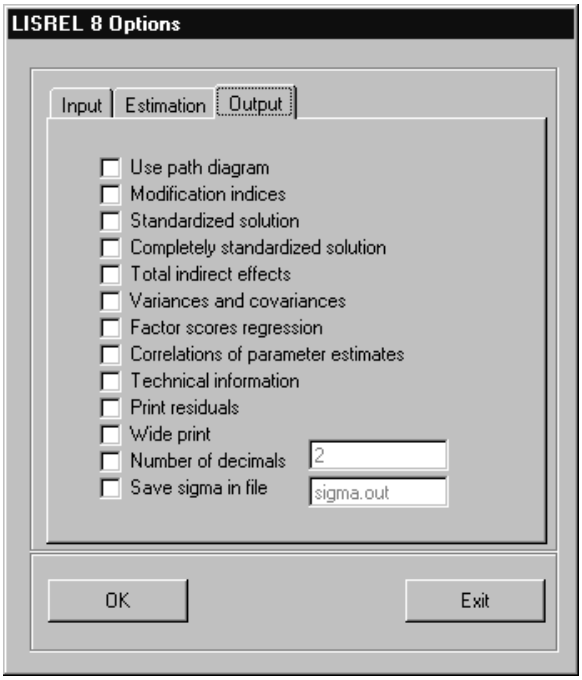
The default estimation method is ML, and in many cases the combination of ML and the covariance matrix is a good choice. However, when ordinal variables are analyzed and the sample is large the recommended procedure is WLS and a polychoric correlation matrix.

When WLS or DWLS is chosen the appropriate asymptotic matrices must have been com-

puted by PRELIS 2, and they must also have been imported into the project. If this has been done STREAMS makes the appropriate references to the asymptotic matrix. However, because subsets of variables cannot be extracted from the asymptotic matrices the model must include all the variables in the matrix. Thus, if a model is to be fitted to another subset of variables it is necessary to compute a new set of matrices, and import these into the project.

The **Estimation** category also allows the user to turn the Admissibility check (Jöreskog & Sörbom, 1989a, p. 23) on and off, and to change the maximum number of iterations from the LISREL default of three times the number of free parameters.

Output



On the **Output** tab of the *LISREL 8 Options* form a large number of check boxes are presented through which the amount and type of output is determined. These options are more or less self-explanatory and detailed information is presented by Jöreskog and Sörbom (1989a, b, 1993b, c).

When the **OK** button on the *LISREL 8 Options* form is clicked the OU line in the edit area is updated. It is, however, always possible to change the OU statement, either by double-clicking on the OU line, or through clicking the **Options** button.

When the **Factor Scores Regressions** option is selected, the post-processor retrieves the factor score weights from the LISREL output listing, and constructs the SPSS instructions needed to actually obtain the factor scores. The SPSS syntax is written to a file with the model name as prefix, and *.sps* as suffix, and which is put in the dictionary directory. In order to achieve sufficient precision in the computation of factor scores the number of decimal places is always set to 6 when this option is selected.

If, for example, the **Factor Scores Regressions** option is selected for the *vr1.mbi* model

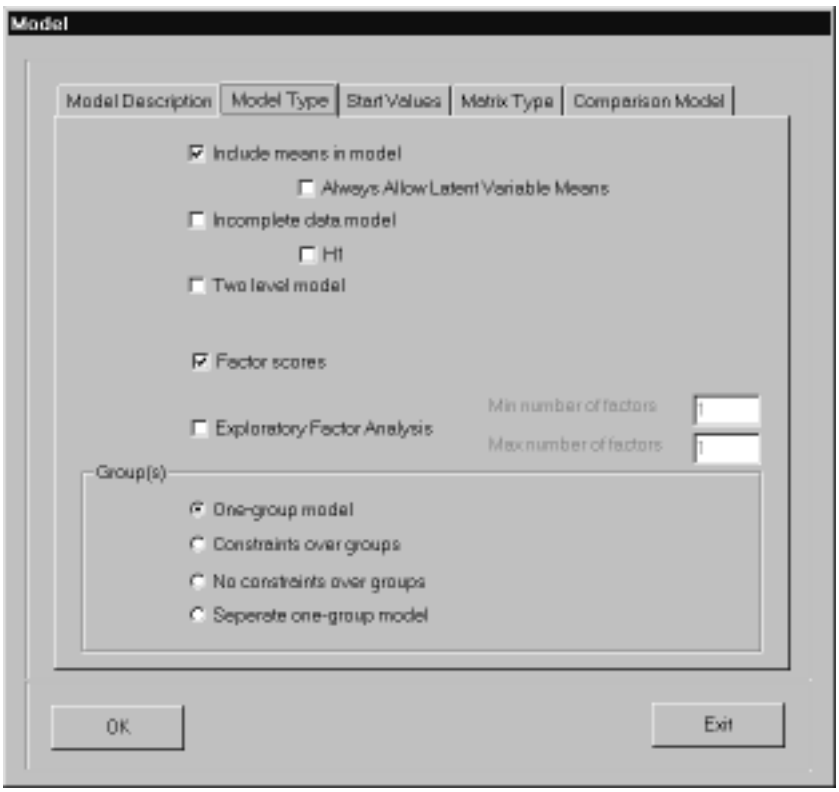
in the HPG project, the following SPSS code is written to the file *vr1.sps*:

SPSS instructions for computation of factor scores constructed by the post-processor

```
COMPUTE Verb = WORD*.27484 +
               READ*.240358 +
                GI*.176937 +
               ERC*.292411 +
                DS*.065153 +
               DTM*.063580 .
COMPUTE Reas = WORD*.04822 +
               READ*.042173 +
                GI*.031045 +
               ERC*.051307 +
                DS*.383831 +
               DTM*.374565 .
EXECUTE.
```

The Mplus Model Type tab

When Mplus is selected as estimation program the **Model Type** tab of the *Model* form offers some program specific options:



One of these is the option **Factor scores,** which causes Mplus to produce a file of individual factor scores, along with the values of the observed variables. The **Factor scores** option requires raw data as input. When this option is used the post-processor also creates a file of SPSS instructions for importing the data into SPSS.

For example, when factor scores are requested for the *vr1.mbi* model the following message is seen in the post-processor listing:

```
Factor scores have been written to
c:\streams\examples\hpg\vr1.fsc,
along with SPSS instructions in
c:\streams\examples\hpg\vr1.sps.
```

The following SPSS instructions are in the file vr1.sps:

```
SET
 UNDEFINED = WARN.
DATA LIST
 FILE='c:\streams\examples\hpg\vr1.fsc' RECORDS = 1 FIXED/
    WORD     1-12 (4)
    READ     13-24 (4)
    GI       25-36 (4)
    ERC      37-48 (4)
    DS       49-60 (4)
    DTM      61-72 (4)
    VERB     73-84 (4)
    REAS     85-96 (4)
 .
EXECUTE.
```

It may be noted that the factor scores computed by Mplus correlate perfectly with those that are computed from the factor score weights produced by LISREL (page 221), even though they are expressed in a different scale. The factor scores that may be computed from the factor score weights produced by Amos (page 217) correlate highly, but not perfectly, with those computed by LISREL and Mplus.

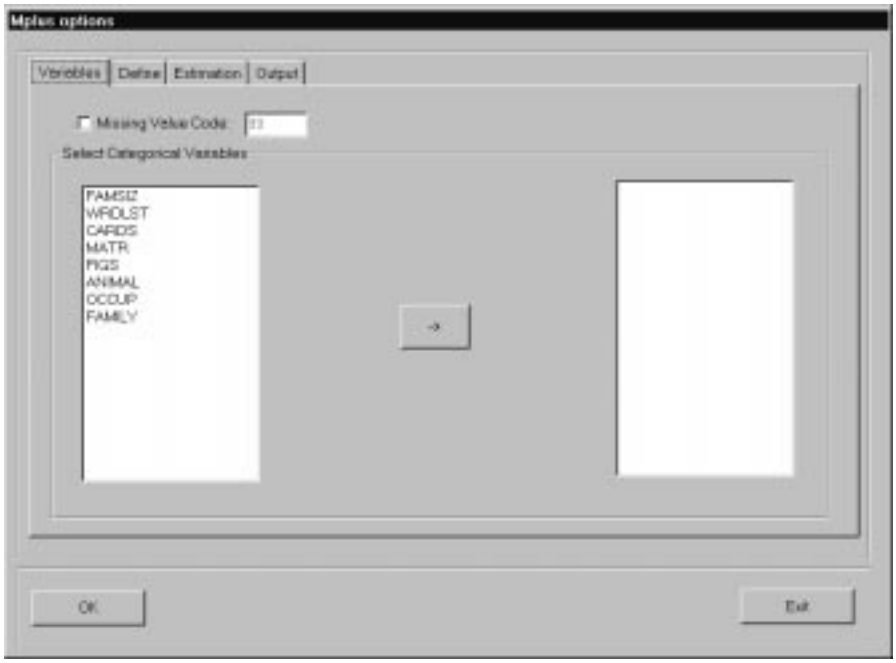
Another option presented on the **Model Type** tab is Exploratory Factor Analysis. When this option is selected Mplus performs exploratory factor analyses, producing solutions with the number of factors ranging between the minimum and maximum given (see Muthén & Muthén, 1998, pp. 133-136).

The Mplus Options form

The *Mplus Options* form offers four tabs with options:

- **Variables**
- **Define**
- **Estimation**
- **Output**

Variables

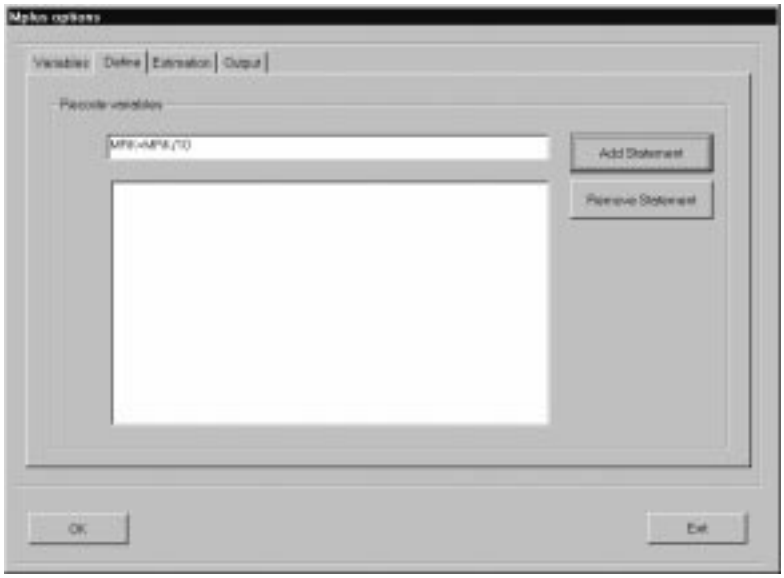


The **Variables** tab allows the user to specify a code value which is regarded as missing data. This option is meaningful only when raw data is input and when the raw maximum likelihood missing data modeling approch described in Chapter 5 is used. A single missing data code may be specified.

On the **Variables** tab those observed variables which are to be treated as categorical variables in the model should also be identified. This is done simply through transfering the categorical variables from the left-hand list-box to the right-hand list-box.

When it comes to categorical variables it should be observed that STREAMS offers full support for analysis of dichotomous variables only with Mplus. For such variables the model specification includes start values, and post-processing of modeling results is performed. When a categorical variable encompasses more than two categories, the pre-processor produces a model specification which does not include start values, and no post-processing of model results is being performed. It should be observed, however, that if start values are copied from a model which treats the variables as continous at least partial start values are obtained.

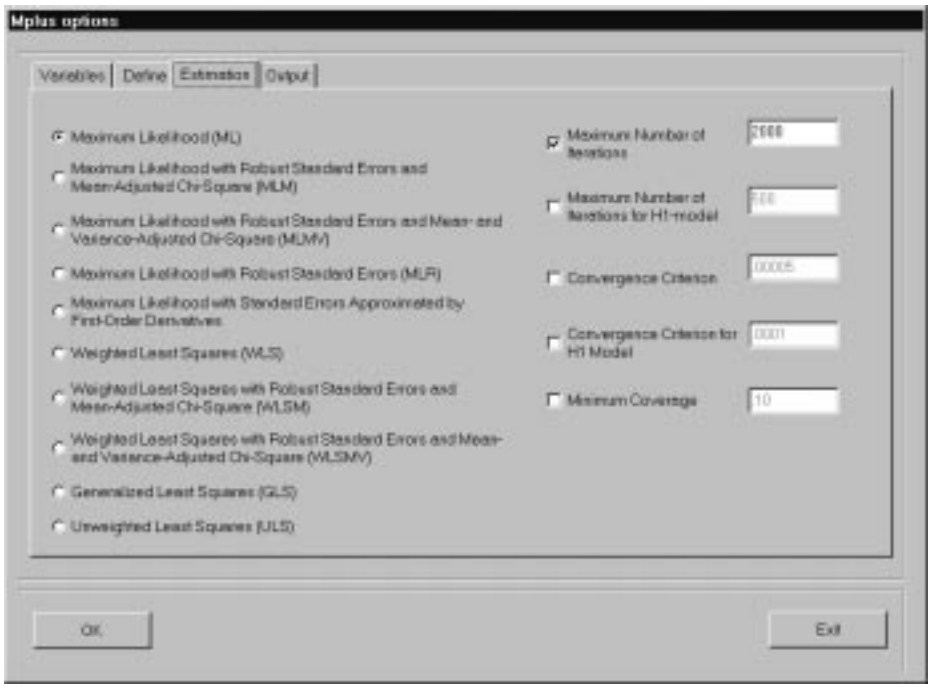
Define



The **Define** tab allows transformations of variables with the functions specified in the Mplus manual (Muthén & Muthén, 1998, pp. 56-58). It should be observed, however, that only existing variables may be transformed, and that no new variables may be specified in this step, which is, of course, because these new variables will not defined in the project dictionary.

Estimation

The **Estimation** tab on the *Mplus Options* form provides a list of choices when it comes

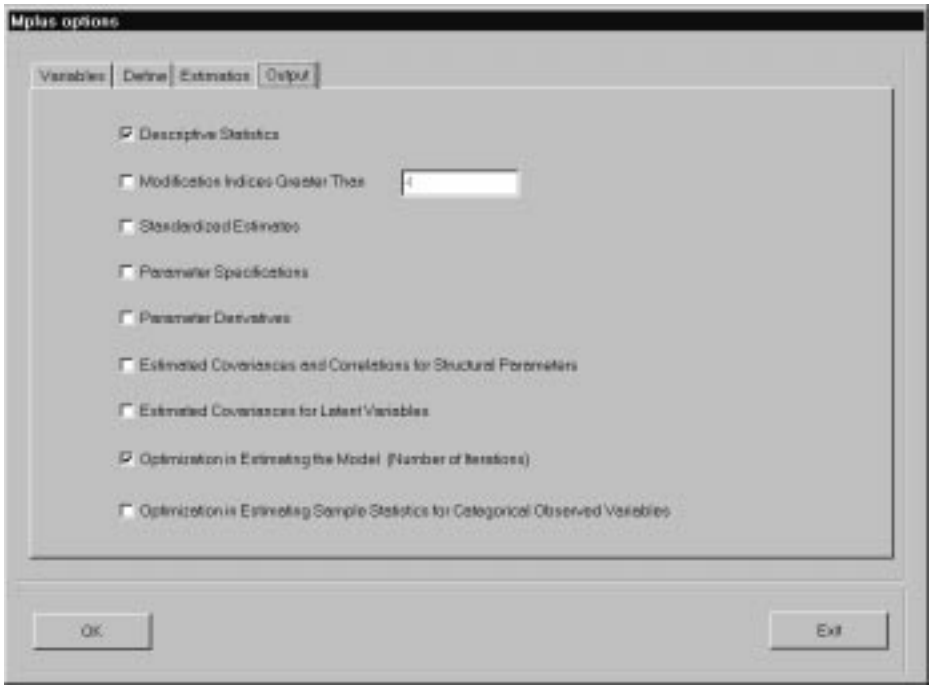


to the process of estimation, in terms of choice of estimators and parameters of estimation.

Mplus offers a rich variety of estimators, and particularly so robust estimators for non-normal data. These estimators produce the same paramterer estimates as does ordinary ML estimation, but the χ^2 test statistic and standard errors are corrected to componsate for the deviation from non-normality, typically resulting in a lower test-statistic and larger standard errors.

Output

The Output tab, finally, offers a set of options concerning what is to be presented in the



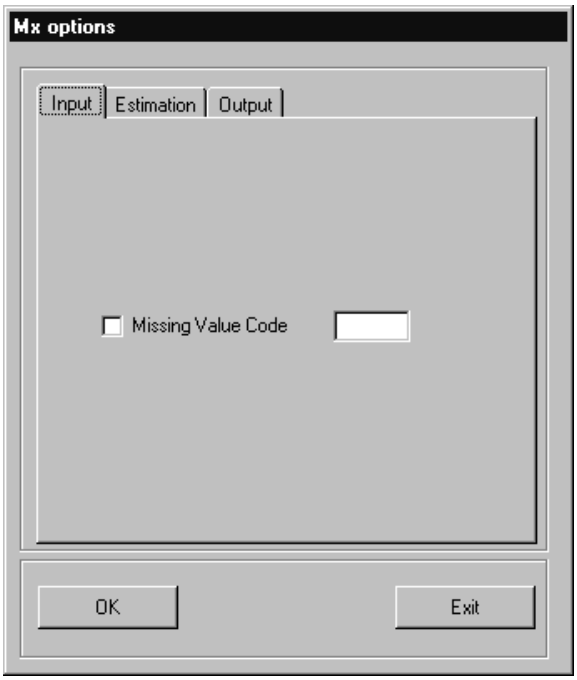
output listing.

The Mx Options form

The *Mx Options* form offers three tabs with options:

- **Input**
- **Estimation**
- **Output**

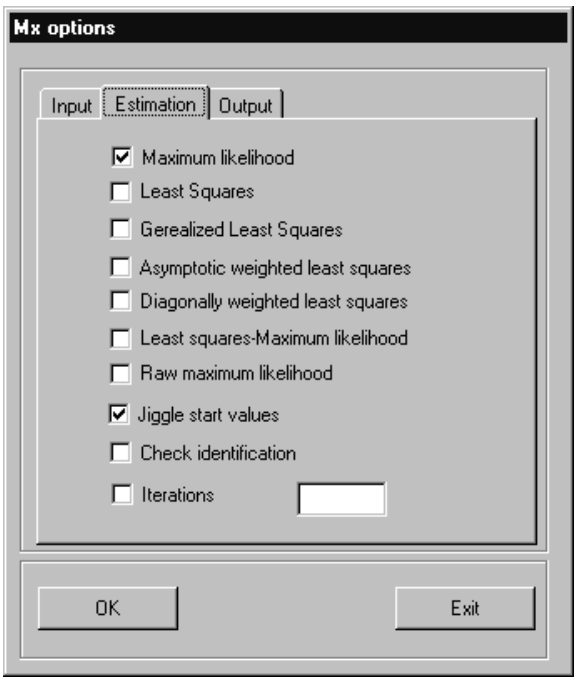
Input



The **Input** tab allows the user to specify a code value which is regarded as missing data:

This option is meaningful only when raw data is input. When a value is supplied here the Mx program uses the raw maximum likelihood missing data modeling approch described in Chapter 5.

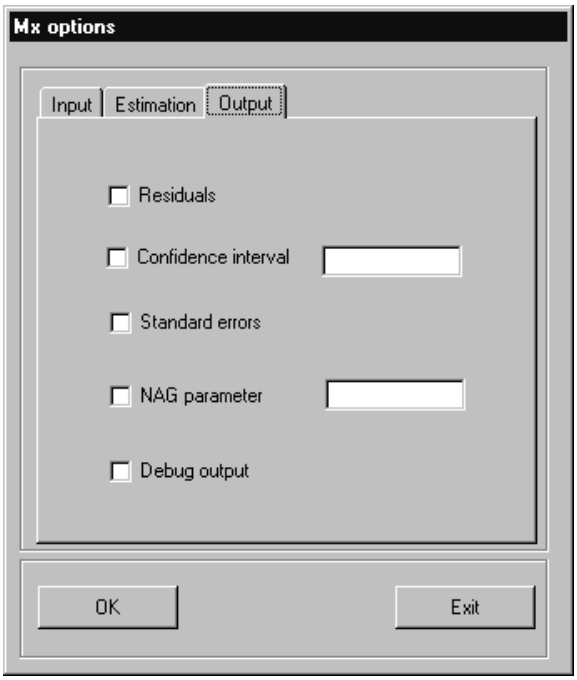
Estimation



The **Estimation** tab on the *Mx Options* form provides a list of choices when it comes to the process of estimation:

Most options on this tab are self-explanatory and more information is given in the Mx guide (Neale, 1997). It should be emphasized that the **Jiggle start values** option, which when checked adds a random component to the supplied start values, may be useful when start values are copied from a previously estimated model. This is because Mx may encounter numerical problems in the minimazation when the start values are close to the final values.

Output



The Output tab, finally, offers a set of options concerning what is to be presented in the output listing.

14 Installing STREAMS

This chapter describes how to install the STREAMS program components, and how to customize the installation to suit the particular configuration of structural equation modeling software available.

Installing STREAMS 2.5

STREAMS 2.5 is typically delivered on CD, or is downloaded from the *www.mwstreams.com* site.

The first installation of STREAMS 2.5 involves the following steps.

1. If an older version of STREAMS is already installed it should be uninstalled, and in particular should the *STREAMS\EXAMPLES* directory be deleted.
2. Insert the CD and start the program *SETUP.EXE*, for example through double clicking the program name, or selecting it and choosing **Run** under the **Start** menu.
3. The program responds by the message Initializing InstallShield Wizard, and then a standard installation procedure starts. The default directory for installing STREAMS is *C:\STREAMS* but this may be changed during the installation. It should be observed, however, that the example projects delivered with the system automatically install into *STREAMS\EXAMPLES*, which makes it convenient to keep the default installation directory.

In addition to the program files and system components, the installation procedure puts a copy of the *STREAMS 2.5 User's Guide* (i. e., the present document) in PDF format in the directory where STREAMS is installed. This document may be read with Adobe Acrobat Reader. This program is included on the installation CD, and may also be downloaded without cost from *http://www.adobe.com.* The installation procedure also puts copies of all the examples projects referred to in the documentation in compressed format in the

STREAMS installation directory. These may be decompressed using the procedure available in STREAMS.

Connecting STREAMS with the Estimation Programs

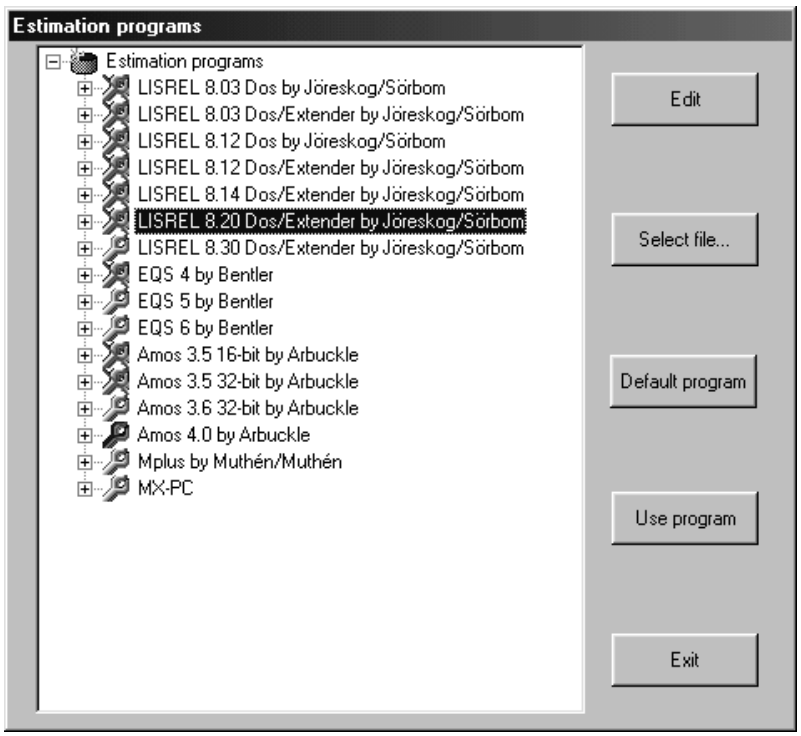
After STREAMS has been installed, connections need to be established between the estimation programs (i. e., Amos, EQS, LISREL and/or Mx), and STREAMS must be informed about which versions of the programs are available.

Amos 4 is automatically localized and connected to by STREAMS whereever it is installed. If the other estimation programs are installed in the default directories assumed by STREAMS they are automatically localized by STREAMS as well. The following default directories are assumed:

Amos 3.5 and 3.6*C:\PROGRAM\AMOS*
EQS 4 and 5:*C:\EQSWIN*
LISREL 8 DOS 8.03:*C:\LIS8*
LISREL 8 DOS 8.12:*C:\LISREL8*
LISREL 8.20:*C:\PROGRAM\INTERACTIVE LISREL*
LISREL 8.30:*C:\LISREL83*
Mplus 1.0:*C:\PROGRAM\Mplus*
Mx 1.44-*C:\STREAMS*

If the estimation programs have been installed in these default directories no further action needs to be taken for STREAMS to be able to use them. However, if the estimation programs have been installed in other directories it is necessary to identify these locations.

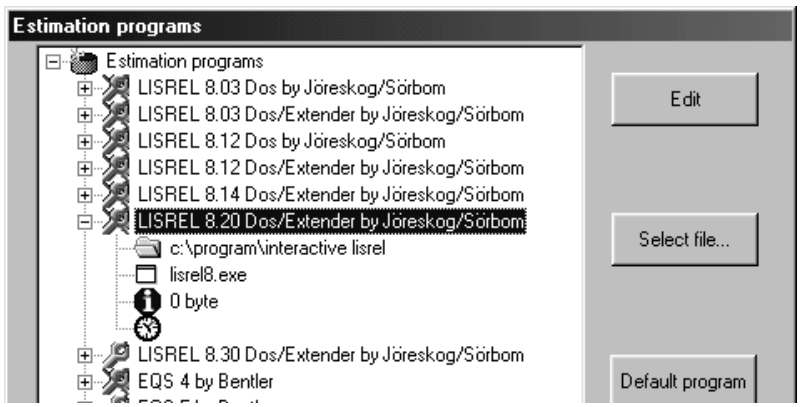
This is done through clicking the **program** button beneath the **Run** button on the *Model Building* form, which produces the *Estimation programs* form:



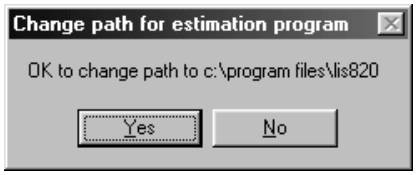
This form displays a series of differently colored wrenches aside the different estimation programs. The colored wrenches have the following meaning:

- Grey wrench: Program is connected to STREAMS, but not selected for use.
- Blue wrench: Program is connected to STREAMS, and selected for use.
- Green wrench: Program is connected to STREAMS, and chosen to be the default program (i. e., this program is used when a new model is created).
- Crossed-over wrench: Program is not installed or not yet connected to STREAMS.

Two different methods may be used to connect a program to STREAMS which has been installed in a another directory than the default directory. One possibility is to enter the pathname of the installation directory. This is done through clicking the + next to the wrench, which presents several items of information, and among them the default installation directory. This is here illustrated for LISREL 8.20:



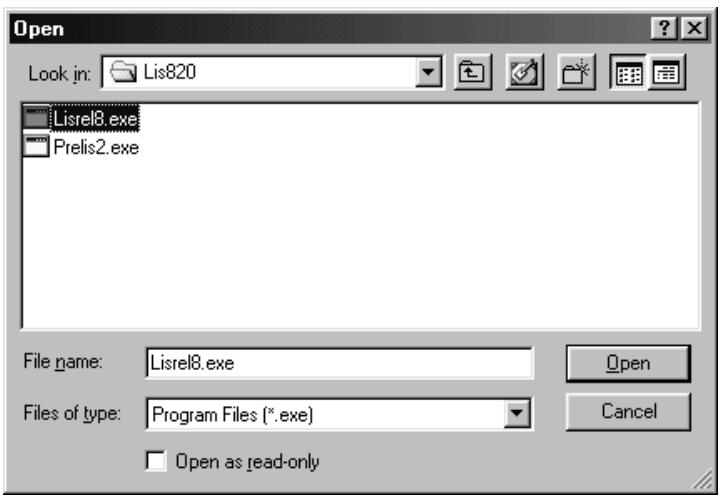
This pathname may be changed through clicking the pathname, and when a box appears around the name the old pathname may be changed. Assuming that LISREL 8.20 has been installed to *C:\PROGRAM FILES\LIS820* this information may be entered. When carriage return is hit the program asks:



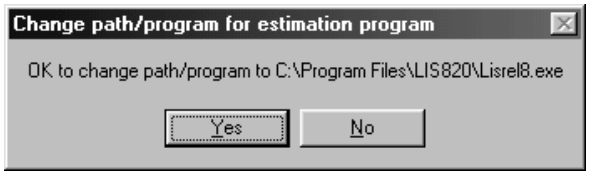
If the **Yes** button is clicked STREAMS connects to the program if it can identify an *.exe* file with the appropriate name in the directory. If STREAMS is successful information about the *.exe* file found in the directory is displayed:

The other method of connecting an estimation program to STREAMS involves identifying a particular *.exe* file, using the standard file open dialogue. This method is useful if the directory contains more than one *.exe* file, or if the *.exe* file has a non-standard name.

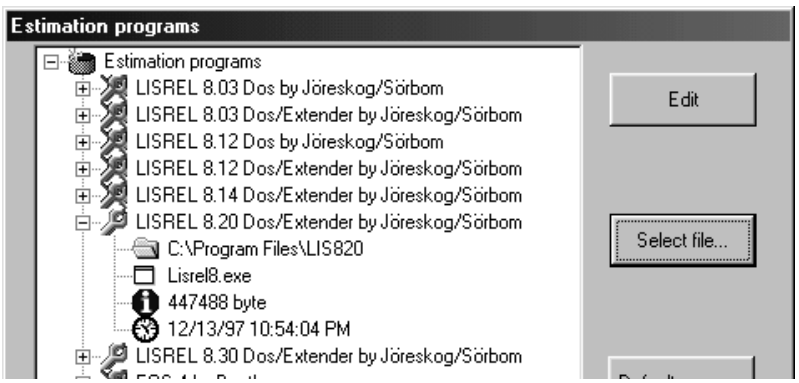
To use this method the **Select file ...** button on the form is clicked, which produces the file open dialogue. This may be used to identify a particular *.exe* file:



When the **Open** button is clicked, STREAMS asks for confirmation:

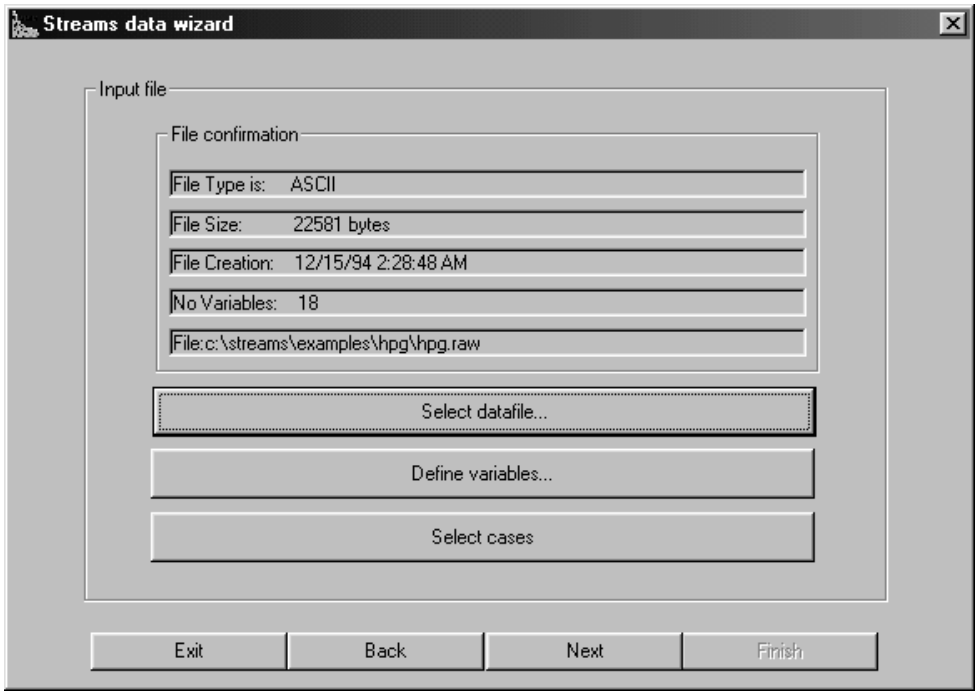


When the **Yes** button is clicked, STREAMS connects to the program, and presents information about the program on the *Estimation programs* form:

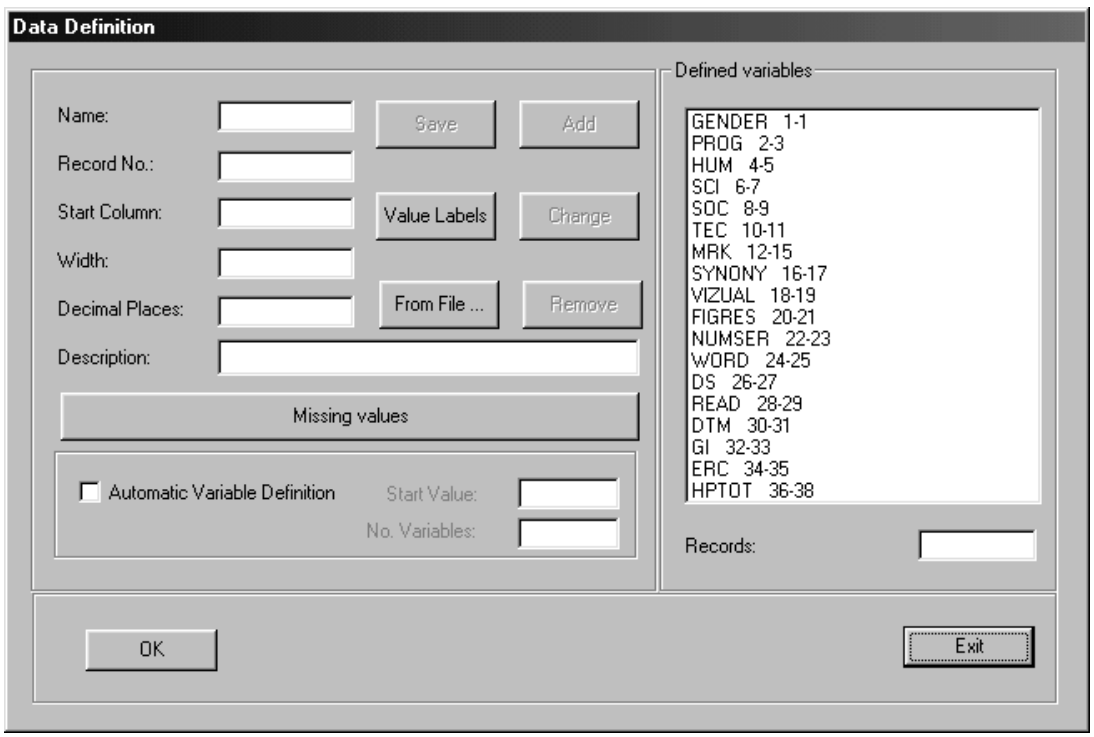


A The STREAMS Rawdata Format

STREAMS offers a simple procedure for describing a rawdata file which is stored in *asci* format (or text format). *The data file must have a fixed format, i. e. for each case the variables must occupy the same positions in the file.* There may be one or more lines of data for each case. With multipe-line input all data records for the first case must appear first, then all data records for the second case (in the same order), and so on. Any record may be at most 4096 characters long. It is somewhat more convenient to have all the information on one record so if there is a choice between organizing the information on one or on several records the former alternative is recommended. It is also recommended that the data file is given the suffix *.raw* (e. g., *hpg.raw*). When such a file is opened when the STREAMS Data Wizard is run the **Input file** form presents information about the file, and the button **Define variables...** is enabled:



If information has previously been entered about the variables and their locations in the data file, this information, which is stored in a file with the file name as prefix and *.sdd* as suffix, is retrieved as well. To add descriptions of variables in a rawdata file, or to update previously given information, the **Define variables...** button should be clicked. This causes the *Data Definition* form to be presented.



This form is used for entering and updating information about the variables in the data file. Previously entered information is shown in the box on the right hand side. The available information may be updated through adding information about further variables, or through changing previously given information. The **Records** field in the lower right-hand corner of the form informs STREAMS about the number of records for each case in the data file. By default this is assumed to be 1 but if the actual number of records is larger the value must be changed.

Entering information about a variable

To enter information about a previously undefined variable the cursor should be moved to the **Name** field, where a variable name is entered. The name should contain no more than 7 alphanumeric characters (i. e., the letters A - Z and the numbers 0 - 9) and it is recommended that upper-case letters are used.

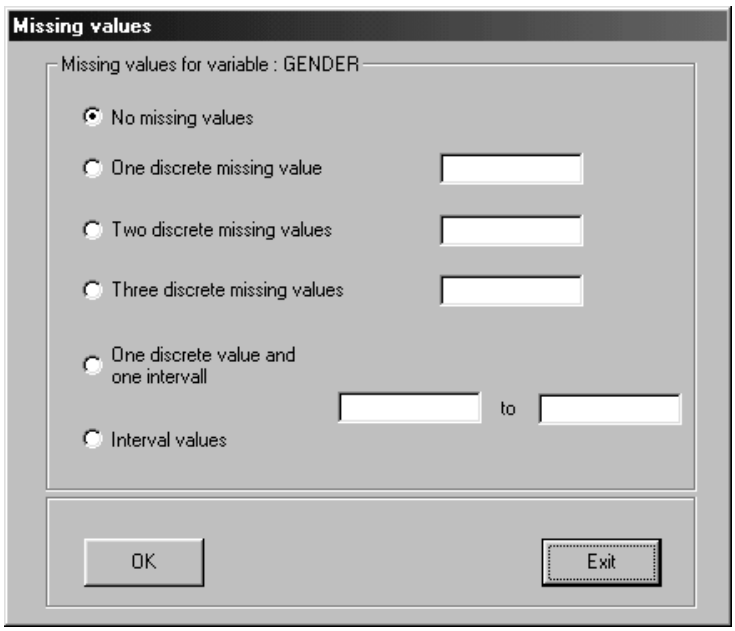
Then use the Tab key (or the Enter key) to move the cursor to the next field, which is labeled **Record No.**. If there is only one record for each case the number 1 is already entered here, but if there are more than one records for each case the appropriate record number should be entered.

The next field is labeled **Start Column**. Here the first position of the variable in the record is given. For example, if the variable occupies positions 10-12 on record 1, the number 10 is entered here. Then use the Tab key to move the cursor to the **Width** field, and enter the number of positions that the variable occupies. For our example the width will be 3. Observe that it is not possible for a variable to span two records.

The **Decimal places** field, which cannot be left blank, is then filled out. It is strongly recommended that decimal points are not actually written to the data file, but that these are implied. For example, if positions 10-12 contain the value 246 for a case this is interpreted as 24.6 if the value 1 is supplied in the decimal places field. If 0 is given the value will be read as 246.0, and if 3 decimal places are specified the value is read as 0. 246. Observe that if decimal points are included in the data values in the file, 0 should be given as the number of decimal points.

Then the **Description** field is filled out, and here descriptive information about the variable is given.

The **Missing Values** button may be clicked if one or more codes to represent missing information is to be entered:

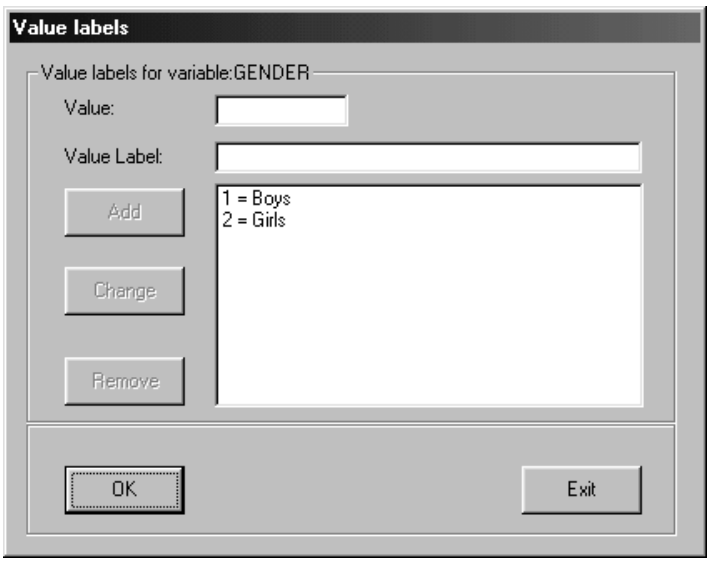


First the number and type of missing values is selected, and then the code used to identify cases who lack a valid value on the variable are entered. It is recommended that extreme values, which may never be a valid value (such as 9 for a dichotomous variable with 0 and 1 as valid codes, or 999 for a variable which may take values between 0 and 144) are used as a missing data codes.

Review the information, and correct any mistakes. To go back to a field use either the Shift-Tab key (i. e., keep the Shift key down as the Tab key is clicked) or move the cursor to the field and click the mouse button. Click the **Add** button when all the information has been enetered.

Adding value labels

For categorical variables the numerical codes signify belongingness to a certain group (e. g., 1 for males, and 2 for females), so for purposes of documentation and ease of interpretation it is often useful also to have the verbal labels for the categories available as well. Such value labels may be defined for a variable if the **Value Labels** button is clicked. When this is done a new form shows:



First enter a code value in the **Value** field (e. g., 1), and then enter the corresponding label in the **Value Label** field (e. g., Boys). When the correct information has been entered click the **Add** button, which moves the information into the white box (i. e., 1 = Boys). Then a new code value and its label may be entered, and so on.

To change previously entered information the line which is to be changed should be clicked, which causes the information to be transferred to the **Value** and **Value Label** fields. Edit the information in the fields, and then click the **Change** button. To remove a value label the line is first marked, and then the **Remove** button is clicked.

Defining many variables with similar characteristics

Sometimes a record contains several consecutively located variables with similar characteristics. An example could be the responses (correct or incorrect) to 30 vocabulary items. The **Automatic Variable Definition** feature may be used to enter information about such variables in an efficient manner. Use the following procedure:

1. Describe the first variable in the sequence and complete each field on the form as if the variable would be entered singly. However, for the **Name** field a prefix only should be entered, and to this prefix STREAMS will then append a sequence number. Thus, if we want the vocabulary items to be labeled VOC1 to VOC30 the prefix VOC should be used.
2. Enter value labels, if any.
3. Click the check box labeled **Automatic Variable Definition** and supply appropriate values in the fields **Number of Variables** and **Start Value** (which in our example would be 30 and 1, respectively).
4. Click the **Add** button, which causes 30 variables to be added to the dictionary.

Observe that this procedure only works when all variables have the same width and are located next to one another in the same record.

Updating information about a variable

In order to change previously entered information the variable is clicked upon in the box on the right hand side. This causes the stored information to be displayed in the fields of the form, and the field or the fields to be updated may now be changed, using the techniques described in the previous section. When the new information has been correctly entered, click the **Change** button.

If a variable is to be removed from the dictionary, click on the variable in the text box so the information about the variable is displayed on the screen, and then click the **Remove** button.

When all information about the variables has been entered and/or updated, click the **OK** button. This causes the dictionary for the rawdata file to be stored in a file, which has the name of the data file, and the suffix *.sdd* (e. g., *hpg.sdd*). If an *.sdd* file exists for a data file it is automatically retrieved into STREAMS next time the data file is opened.

Copying information from an existing data dictionary

There frequently is a need to describe a rawdata file which has the same variables in the same positions as a file for which an *.sdd* file already exists. When this is the case it is not necessary to enter the information again, because it may be copied from the existing dictionary. To do that the **From File...** button on the *Data Definition* form is clicked. This produces a dialogue box, asking the user to identify an *.sdd* file. An appropriate data dictionary in the current directory or another directory should then be identified and opened, which causes the information in this dictionary to be copied to the data dictionary for the new rawdata file.

An Example of a Dictionary for Rawdata

Rawdata may be described using the procedures for describing characteristics of data supplied in STREAMS. An example of such a data dictionary is presented below.

The Swedish Scholastic Aptitude Test Data

The projects included in the *EXAMPLES* subdirectory under the *STREAMS* directory include some examples of data dictionaries. Here a few of these will be briefly commented.

The *hpg.raw* file in the *HPG* subdirectory contains data for the *hpg* project which is used in several examples in this text. The data file comprises 579 subjects who have taken the Swedish Scholastic Aptitude Test (SweSAT), along with some other tests. The data file includes the variable shown in Table 5:

There are no missing data for any subject in this file. The information in Table 5 has been entered into the dictionary *hpg.sdd*, which is retrieved automatically when the *hpg.raw* file is opened.

TABLE 4. **The variables in** *hpg.raw*

Label	Description	Start position	Field Width	Number of Decimals
GENDER	Gender (male=1, female=2)	1	1	0
PROG	Program in Upper Secondary School	2	2	0
HUM	Dummy for the Humanistic program	4	2	0
SCI	Dummy for the Science program	6	2	0
SOC	Dummy for the Social program	8	2	0
TEC	Dummy for the Technical program	10	2	0
MRK	Mean grades from Upper Sec School	12	4	2
SYNONY	Synonyms test	16	2	0
VIZUAL	Spatial test	18	2	0
FIGRES	Figural Reasoning test	20	2	0
NUMSER	Number Series test	22	2	0
WORD	The vocabulary subtest of the SweSAT	24	2	0
DS	The Data Sufficiency subtest of the SweSAT	26	2	0
READ	The reading subtest of the SweSAT	28	2	0
DTM	The Diagrams, Tables and Maps subtest of the SweSAT	30	2	0
GI	The General Information subtest of the SweSAT	32	2	0
ERC	The English Reading Comprehension subtest of the SweSAT	34	2	0
HPTOT	The total raw score on the SweSAT	36	3	0

References

Aish, A. M., & Jöreskog, K. G. (1990). A panel model for political efficacy and responsiveness. An application of LISREL 7 with weighted least squares. *Quality and Quantity, 24*, 405-426.

Allison, P. D. (1977). Estimation of linear models with incomplete data. In C. Clogg (Ed.) *Sociological Methodology 1987*, pp. 71-103. San Francisco: Jossey Bass.

Arbuckle, J. L. (1996). Full information estimation in the presence of incomplete data. In G. A. Marcoulides & R. E. Schumacker (Eds.) *Advanced structural equation modeling. Issues and techniques*. Mahwah, New Jersey: Erlbaum, pp. 243-277.

Arbuckle, J. L. (1997). *Amos Users' Guide. Version 3.6*. Chicago, IL: SmallWaters Corporation.

Bentler, P. M. (1988). Causal modeling via structural equation systems. In J. R. Nesselroade & R. B. Cattell (Eds.) *Handbook of multivariate experimental psychology* (2nd ed., pp. 317-335).

Bentler, P. M. (1989). *EQS: Structural equations program manual*. Los Angeles: BMDP Statistical Software.

Bentler, P. M. (1990). Comparative fit indexes in structural models. *Psychological Bulletin, 107*, 238-246.

Bentler, P. M. (1992). *EQS: Structural equations program manual*. Los Angeles: BMDP Statistical Software.

Bentler, P. M. (1995). *EQS: Structural equations program manual*. Encino, CA: Multivariate Software.

Bentler, P. M., & Dudgeon, P. (1996). Covariance structure analysis: Statistical practice, theory and directions. *Annual Review of Psychology, 47*, 563-592.

Bentler, P. M., & Woodward, J. A. (1978). A Head Start reevaluation: Positive efects are not yet demonstrable. *Evaluation Quarterly, 2*, 493-510.

Bielby, W. T., Hauser, R. M., & Featherman, D. L. (1977). Response errors of black and nonblack males in models of the stratification process. *American Journal of Sociology, 82*, 1242-1288.

Bock, R. D. (1989) (Ed.). *Multilevel analysis of educational data*. San Diego: Academic Press.

Bollen, K. (1989). *Structural equations with latent variables*. New York: Wiley.

Bollen, K., & Long, J. S. (Eds.) (1993). *Testing structural equation models*. Newbury Park, CA: Sage.

Bollen, K., & Stine, R. A. (1992). Bootstrapping goodness-of-fit measures in structural

equation models. *Sociological Methods and Research, 21,* 205-229.

Boomsma, A. (1985). Nonconvergence, improper solutions, and starting values in LISREL maximum likelihood estimation. *Psychometrika, 50*(2), 229-242.

Browne, M. W. (1982). Covariance structures. In Hawkins, D. M. (Ed.) *Topics in applied multivariate analysis*. Cambridge: Cambridge University Press, 72-141.

Browne, M. W. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology, 37,* 62-83.

Browne, M. W., & Cudeck, R. (1993). Alternative ways of assessing model fit. In K. A. Bollen & J. S. Long (Eds.) *Testing structural equation models,* pp. 136-162. Thousand Oakes, CA: Sage.

Byrne, B. M. (1994). *Structural equation modeling with EQS and EQS/Windows. Basic concepts, applications, and programming.* Thousand Oakes, CA: Sage.

Byrne, B. M. (1995). One application of structural equation modeling from two perspectives: Exploring the EQS and LISREL strategies. In R. H. Hoyle (Ed.). *Structural equation modeling. Concepts, issues, and applications.* Thousand Oaks, CA: Sage., (pp. 138-157).

Carroll, J. B. (1982). The measurement of intelligence. In R. J. Sternberg (Ed.), *Handbook of human intelligence,* (pp. 29-120). New York: Cambridge University Press.

Carroll, J. B. (1993). *Human cognitive abilities. A factor-analytic approach*. Cambridge: Cambridge University Press.

Chou, C.-P., & Bentler, P. M. (1995). Estimates and tests in structural equation modeling. In R. H. Hoyle (Ed.). *Structural equation modeling. Concepts, issues, and applications*. Thousand Oaks, CA: Sage., (pp. 37-55).

Cronbach, L. J. (1976). *Research on classrooms and schools: Formulations of questions, design and analysis.* (Stanford, California: Occasional papers of the Stanford Evaluation Consortium).

Cronbach, L. J., & Snow, R. E. (1977). *Aptitudes and instructional methods*. New York: Irvington.

Dempster, A. P., & Rubin, D. B. (1983). Overview. In *Incomplete data in sample surveys. Vol II: Theory and annotated bibliography* (W. G. Madow, I. Olkin, & D. B. Rubin, Eds.). New York: Academic Press.

Draper, N. S., & Smith, H. (1967). *Applied regression analysis*. New York: Wiley.

Efron, B., & Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals and other measures of statistical accuracy. *Statistical Science, 1,* 54-74.

Ekstrom, R. B., French, J. W., & Harman, H. H. (1976). *Manual for kit of factor-referenced cognitive tests, 1976.* Priceton, NJ: Educational Testing Service.

Elley, W. B. (1994) (Ed.). *The IEA study of reading literacy: Achievement and instruction in thirty-two school systems.* Oxford: Pergamon.

Gerbing, D. W., & Anderson, J. C. (1993). Monte Carlo evaluations of goodness-of-fit

indices for structural equation models. In K. A. Bollen & J. S. Long (Eds.) *Testing structural equation models,* pp. 136-162. Thousand Oakes, CA: Sage.

Goldberger, A. S. (1964). *Econometric theory*. New York: Wiley.

Goldstein, H., & McDonald, R. (1988). A general model for the analysis of multilevel data. *Psychometrika, 53,* 455-467.

Gustafsson, J. E. (1984). A unifying model for the structure of intellectual abilities. *Intelligence, 8,* 179-203.

Gustafsson, J. E. (1988). Hierarchical models of individual differences in cognitive abilities. In R. J. Sternberg, *Advances in the psychology of human intelligence* Vol. 4, (pp. 35-71). Hillsdale, NJ: Erlbaum.

Gustafsson, J. E. (1989). Broad and narrow abilities in research on learning and instruction. In R. Kanfer, P. L. Ackerman, & R. Cudeck (Eds.), *Abilities, motivation, and methodology. The Minnesota symposium on learning and individual differences,* (pp. 203-237). Hillsdale, NJ: Erlbaum.

Gustafsson, J. E. (1994). Hierarchical models of intelligence and educational achievement. In A. Demetriou, & A. Efklides (Eds.), *Intelligence, mind, and reasoning. Structure and development,* (pp. 45-73). Amsterdam: North-Holland.

Gustafsson, J.-E. (1997). Measurement characteristics of the IEA Reading Literacy scales for 9-10 year-olds at country and individual levels. *Journal of Educational Measurement, 34,* 233-251.

Gustafsson, J.-E. (in press). Social background and teaching factors as determinants of reading achievement at classroom and individual levels. *Journal of Nordic Educational Research.*

Gustafsson, J.-E., & Balke, G. (1993). General and specific abilities as predictors of school achievement. *Multivariate Behavioral Research, 28*(4), 407-434.

Gustafsson, J.-E., & Undheim, J. O. (1996). Individual differences in cognitive functions. In D. Berliner & R. Calfee (Eds.). *Handbook of Educational Psychology.* New York: Macmillan.

Gustafsson, J.-E., Wedman, I., & Westerlund, A. (1992). The dimensionality of the Swedish Scholastic Aptitude Test. *Scandinavian Journal of Educational Research, 36,* 21-39.

Guttman, L. (1954). A new approach to factor analysis: The radex. In P. F. Lazarsfeld (Ed.), *Mathematical thinking in the social sciences,* (pp. 216-257). Glencoe, Illinois: The Free Press.

Hayduk, L. A. (1987). *Structural equation modeling with LISREL: Essentials and advances.* Baltimore: Johns Hopkins.

Härnqvist, K. (1978). Primary mental abilities at collective and individual levels. *Journal of Educational Psychology, 70,* 574-584.

Härnqvist, K., Gustafsson, J. E., Muthén, B. O., & Nelson, G. (1994). Hierarchical models of ability at individual and class level. *Intelligence, 18,* 165-187.

Hershberger, S. L. (1994). The specification of equivalent models before the collection of data. In A. von Eye & C. C. Clogg (Eds.). *Latent variables analysis.*

Applicatiions for developmental research. Thousand Oaks, CA: Sage.

Hoyle, R. H. (1995) (Ed.). *Structural equation modeling. Concepts, issues, and applications*. Thousand Oaks, CA: Sage.

Holzinger, K. J., & Swineford, F. (1939). *A study in factor analysis: The stability of a bi-factor solution.* Supplementary Educational Monographs, no. 48. Chicago, IL: The University of Chicago.

Hox, J. J. (1994). *Applied multilevel analysis*. Amsterdam: TT-Publikaties.

Humphreys, L. G. (1968). The fleeting nature of college academic success. *Journal of Educational Psychology, 59*, 375-380.

Jöreskog, K. G. (1969). A general approach to confirmatory factor analysis. *Psychometrika, 34*, 183-202.

Jöreskog, K. G. (1970). Estimation and testing of simplex models. *British Journal of Mathematical and Statistical Psychology, 23*, 121-145.

Jöreskog, K. G. (1971). Statistical analysis of sets of congeneric tests. *Psychometrika, 36*, 109-133.

Jöreskog, K. G., & Sörbom, D. (1988). *PRELIS - A program for multivariate data screening and data summarization. A preprocessor for LISREL.* Chicago: Scientific Software.

Jöreskog, K. G., & Sörbom, D. (1989a). *LISREL 7. A Guide to the Program and Applications. (2nd Edition)*. Chicago: SPSS Inc.

Jöreskog, K. G., & Sörbom, D. (1989b). *LISREL 7 User's Reference Guide*. Chicago: Scientific Software.

Jöreskog, K. G., & Sörbom, D. (1993a). *PRELIS 2 User's Reference Guide*. Chicago: Scientific Software.

Jöreskog, K. G., & Sörbom, D. (1993b). *LISREL 8 User's Reference Guide*. Chicago: Scientific Software.

Jöreskog, K. G., & Sörbom, D. (1993c). *LISREL 8: Structural Equation Modeling with the SIMPLIS Command Language*. Hillsdale, NJ: Erlbaum.

Jöreskog, K. G., & Sörbom, D. (1996a). *PRELIS 2 User's Reference Guide (2nd ed.)*. Chicago: Scientific Software.

Jöreskog, K. G., & Sörbom, D. (1996b). *LISREL 8 User's Reference Guide (2nd ed.)*. Chicago: Scientific Software.

Kenny, D. A. (1979). *Correlation and causality*. New York: Wiley.

Kerchoff, A. C. (1974). *Ambition and attainment*. Rose Monograph Series.

Little, R. J. A., & Rubin, D. B. (1987). *Statistical analysis with missing data.* New York: Wiley.

Loehlin, J. C. (1992). *Latent variable models. An introduction to factor, path, and structural analysis (2nd ed.)*. Hillsdale, NJ: Erlbaum.

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental test scores.* Reading: Addison-Wesley.

Magidson, J. (1977). Toward a causal model approach for adjusting for pre-existig differences in the non-equivalent cotrol group situation. *Evaluation Quarterly*, 1, 399-420.

McArdle, J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action model for moment structures. *British Journal of Mathematical and Statistical Psychology, 37,* 234-254.

McDonald, R. P. (1993). A general model for two-level data with responses missing at random. *Psychometrika, 58,* 575-585.

McDonald, R. P. (1994). The bilevel reticular action model for path analysis with latent variables. *Sociological Methods & Research, 22,* 399-413.

McDonald, R. P., & Goldstein, H. (1989). Balanced versus unbalanced designs for linear structural relations in two-level data. *British Journal of Mathematical and Statistical Psychology, 42,* 215-232.

McDonald, R. P., & Marsh, H. W. (1990). Choosing a multivariate model: Noncentrality and goodness-of-fit. *Psychological Bulletin, 107,* 247-255.

Mulaik, S. A., James, L. R., Van Alstine, J., Bennett, N., Lind, S., & Stilwell, C. D. (1989). Evaluation of goodness-of-fit indices for structural equation models. *Psychological Bulletin, 105,* 430-445.

Muthén, B. (1988). *LISCOMP User's Manual*. Chicago: Scientific Software.

Muthén, B. O. (1989). Latent variable modeling in heterogenous populations. *Psychometrika, 54,* 557-585.

Muthén, B. O. (1990). *Mean and covariance structure analysis of hierarchical data*. (UCLA Statistics Series No. 62). Los Angeles: University of California.

Muthén, B. O. (1991). Multilevel factor analysis of class and student achievement components. *Journal of Educational Measurement, 28,* 338-354.

Muthén, B. O. (1994). Multilevel covariance structure analysis. *Sociological Methods & Research, 22(3),* 376-398.

Muthen, B., & Kaplan, D.(1985). A comparison of some methodologies for the factor analysis of non-normal Likert variables. *British Journal of Mathematical and Statistical Psychology, 38,* 171-189.

Muthen, B., & Kaplan, D.(1992). A comparison of some methodologies for the factor analysis of non-normal Likert variables: A note on the size of the model. *British Journal of Mathematical and Statistical Psychology, 45,* 19-30.

Muthén, B. O., Kaplan, D., & Hollis, M. (1987). On structural equation modeling with data that are not missing completely at random. *Psychometrika, 52,* 431-462.

Muthén, B. O., Khoo, S.-T., & Nelson Goff, G. (1994) Multidimensional description of subgroup differences in mathematics achievement data from the 1992 National Assessment of Educational Progress. Paper presented at the Annual Meeting of the American Educational Reseach Association.

Muthén, B. O., & Satorra, A. (1989). Multilevel aspects of varying parameters in structural models. In R. D. Bock (Ed.). *Multilevel analysis of educational data,* pp. 87-99. San Diego: Academic Press.

Muthén, L. K., & Muthén, B. O. (1998). *Mplus User's Guide*. Los Angeles, CA: Muthén & Muthén.

Neale, M. C. (1995). *Mx: Statistical Modeling*. Box 710 MCV, Richmond, VA 23298: Department of Psychiatry. 3rd edition.

Nelson, G., & Muthén, B. O. (1991). *Analysis preparation steps for multilevel analysis using SOURCE.BW and LISCOMP*. Unpublished manuscript, University of California, Los Angeles, Graduate School of Education.

Pedhazur, E. J., & Pedhazur Schmelkin, L. (1991). *Measurement, design, and analysis. An integrated approach.* Hillsdale, NJ: Erlbaum.

Reuterberg, S.-E., & Gustafsson, J.-E. (1992). Confirmatory factor analysis and reliability: Testing measurement model assumptions. *Educational and Psychological Measurement, 52*, 795-811.

Rosén, M. (1995). Gender differences in means, variances and structure of hierarchically structured abilities. *Learning and Instruction, 5*(1), 37-62.

Rosén, M. (1997). Country Differences in Reading Performance: A Reanalysis of the IEA Reading Literacy Study. Paper presented at the Annual Meeting of the American Educational Research Association, Chicago.

Rosén, M. (1998a). Gender differences in hierarchically ordered ability dimensions: The impact of missing data. *Structural Equation Modeling, 5*(1), 37-62.

Rosén, M. (1998b). *Gender differences in patterns of knowledge*. Göteborg: Acta Universitatis Gothoburgensis.

Rosén, M. (in press). Gender differences in reading performance on documents across countries. *Reading & Writing.*

Roth, P. L. (1994). Missing data: A conceptual review for applied psychologists. *Personnel Psychology, 47*, 537-560.

Satorra, A., & Bentler, P. M. (1994). Corrections to test statistics and standard errors in covariance structure analysis. In A. von Eye & C. C. Clogg (Eds.) *Latent variable analysis: Applications for developmental research* (pp. 399-419). Thousand Oaks, CA: Sage.

Schumacker, R. E., & Lomax, R. G. (1996). *A beginner's guide to structural equation modeling*. Mahwah: New Jersey, Lawrence Erlbaum.

Sireci, Thissen, & Wainer

Snow, R. E. (1996). Individual differences in affective and conative functions. In D. Berliner & R. Calfee (Eds.). *Handbook of Educational Psychology*. New York: Macmillan.

Sobel, M. E. (1994). Causal inference in latent variable models. In A. von Eye & C. C. Clogg (Eds.) *Latent variables analysis. Applications for developmental research*. Thousand Oakes, CA: Sage, (pp. 3-35).

Spearman, C. (1904a). "General intelligence," objectively determined and measured. *American Journal of Psychology, 15*, 201-293.

Spearman, C. (1904b). The proof and measurement of association between two things.

American Journal of Psychology, 15, 72-101.

Steiger, J. H. Shapiro, A. & Browne, M. W. (1985). On the multivariate asymptotic distribution of sequential chi-square statistics. *Psychometrika, 50*, 253-263.

Steyer, R., & Schmitt, T. (1994). The theory of confounding and its application in causal modeling with latent variables. In A. von Eye & C. C. Clogg (Eds.) *Latent variables analysis. Applications for developmental research*. Thousand Oakes, CA: Sage, (pp. 36-67).

Sörbom, D. (1974). A general method for studying differences in factor means and factor structures between groups. *British Journal of Mathematical and Statistical Psychology, 27*, 229-239.

Sörbom, D. (1981). Structural equation models with structured means. In K. G. Jöreskog & H. Wold (Eds.) *Systems under indirect observation: Causality, structure and prediction, Vol 1*. Amsterdam: North-Holland Publishing Co.

Thurstone, L. L. (1931). Multiple factor analysis. *Psychological Review, 38*, 406-427.

Thurstone, L. L. (1938). Primary mental abilities. *Psychometric Monographs*, No. 1.

Thurstone, L. L. (1947). *Multiple factor analysis.* Chicago: University of Chicago Press.

Undheim, J. O. (1981). On intelligence IV: Toward a restoration of general intelligence. *Scandinavian Journal of Psychology, 22*, 251-265.

Waller, N. G. (1993). Software review. Seven confirmatory factor analysis programs: EQS, EzPATH, LINCS, LISCOMP, LISREL 7, SIMPLIS, and CALIS. *Applied Psychological Measurement, 17*(1), 73-100.

Werts, C. E., Rock, D. R., Linn, R. L., & Jöreskog, K. G. (1978). A general method for estimating the reliability of a composite. *Educational and Psychological Measurement, 38*, 933-938.

Wheaton, B., Muthén, B., Alwin, D., & Summers, G. (1977). Assessing reliability and stability in panel models. In D. R. Heise (Ed.) *Sociological Methodology 1977*. San Francisco: Jossey Bass.

Whotke, W., Bock, R. D., Curran, L. T., Fairbank, B. A., Augustin, J. W., Gillet, A. H., & Guerrero, Jr. C. (1991). Factor analytic examiation of the Armed Services Vocational Aptitude Battery (ASVAB) and the Kit of Factor-Referenced Tests. Brooks Air Force Base, TX: Air Force Human Resources Laboratory.

Willett, J. B., & Sayer, A. G. (1994). Using covariance structure analysis to detect correlates and predictors of change. *Psychological Bulletin, 116*, 363-381.

Wright, S. (1934). The method of path coefficients. *Annals of Mathematical Statistics, 5*, 161-215.