

7/12/2006

Chapter 6

Building A Model

In the simplest, most straightforward case, building a multi-equation econometric model with MODLER basically involves three steps.

The first is to define a Model File to hold the equations, which also involves a description of the model and the specification of its observation frequency. From the Central Control Screen, there are two ways to proceed. One is to click **Model | Build or Edit | Create New Model**. Alternatively, you can type the command:

DEFINE MODEL NEWMODL

In either case, MODLER responds by displaying the form shown in Figure 6-1, which invites you to provide a name, a description, and an observation frequency for the model. All these characteristics can be changed later, except that changing the name is the most involved: the one you supply is applied both to the model as a set of equations and to the file that holds the model. In this example, the file will be named NEWMODL.MOD. At this stage, the model is simply an “empty box;” we have simply created a container for it

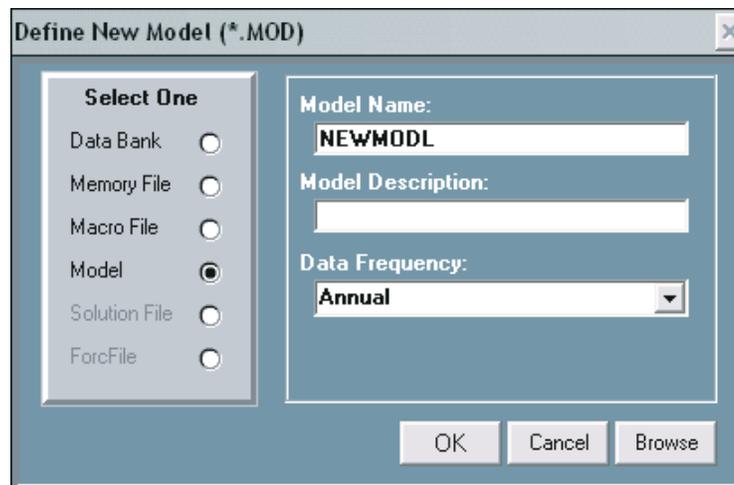


Figure 6-1. Create New Model

As a second step, we then need to provide the accounting and any technical identities. These define the framework or superstructure of the model.

The third and final step is to estimate the parameters of the behavioral relationships, and then include the estimated equations in the model. This activity ordinarily involves performing a series of regressions.

However, with the exception of the first of these operations, there is no necessary order in which things must be done and there are alternative ways to do them: the model's equations can be entered from the keyboard or, they can be loaded from a file, or—at least in the case of behavioral relationships—they can be created as a direct output of the parameter estimation process. It is possible therefore to start the discussion with any one of several topics, but since the previous chapter dealt with regression, it might be useful, as a topical bridge, to begin by considering how to include estimated equations and work from there.

Putting Estimated Equations Into A Model File

To fix ideas, consider, for instance, the regression command:

$$\text{LN(EMPL)=F(LN(GNP),LN(EMPL(-1)))}$$

which specifies a log equation for employment as a function of GNP and lagged employment and is also a regression command. Since the model has been given a name and the MOD file created, once this regression command has been successfully executed, the next command could take the simple form:

PUTEQ

This command, as described in Chapter 5, has the effect of loading the estimated equation into the model file, assuming that the regression has been successfully performed.

In order to view the model at this stage, from the Central Control Screen, click on **Model | Model Builder and Equation Editor | View**. In response, you will be presented with the choice of viewing the complete model or “selected equations.” But as these are the same thing in the case of a model with only one equation, make the first choice and you will next see a display similar to that shown in Figure 6-2 on the next page.

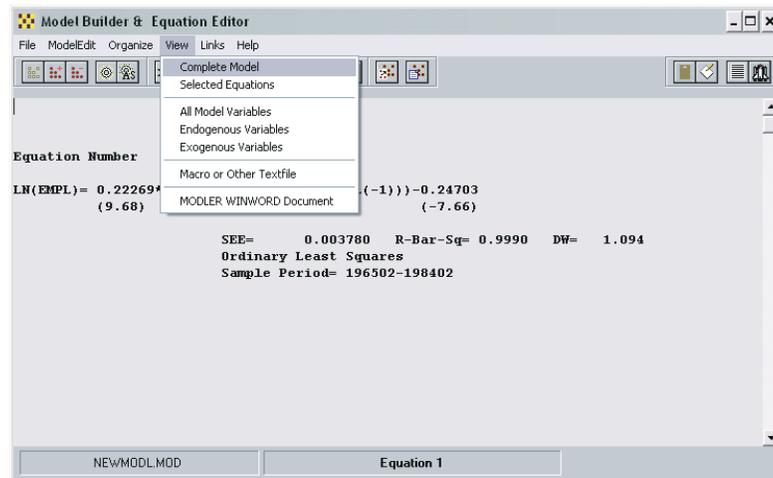


Figure 6-2. Model Equation Display

Thus once you have estimated an equation, to include it in the model requires only telling MODLER to perform this task. The program takes care of all the details. Furthermore, you are not necessarily limited to one choice. If you are not sure whether the above equation is suitable, you can estimate several alternatives and, for the time being, put all these into the model file; for example, if you were to do this using a macro, the relevant commands might be:

```

EMPL=F(GNP,EMPL(-1),TREND)
PUTEQ
LN(EMPL)=F(LN(GNP),LN(EMPL(-1)),TREND)
PUTEQ
LN(EMPL/EMPL(-4))=F(LN(GNP/GNP(-4)),LN(EMPL(-4)/
GNP(-4)))
PUTEQ

```

Notice that if these commands are executed, the effect is that the model will now have four equations with EMPL on the left-hand side. To see them all, in the context of the model, once again choose **Model | Model Builder and Equation Editor | View | Complete Model**. At this stage, if you would like to have hard copy, simply click on **File | Print Display Window**. You will then be given the choice whether to send the output to your printer or word processor.

In order to establish the concepts, the foregoing description has focused on the process as command based. However, once the model has been defined you might prefer to work using the menu system. In this case, instead of proceeding directly to the Parameter Estimation screen, begin from the Central Control screen by clicking on **Model | Model Builder and Equation Editor | ModelEdit**. The effect will be to invoke the Model Builder & Equation Editor screen as your general model

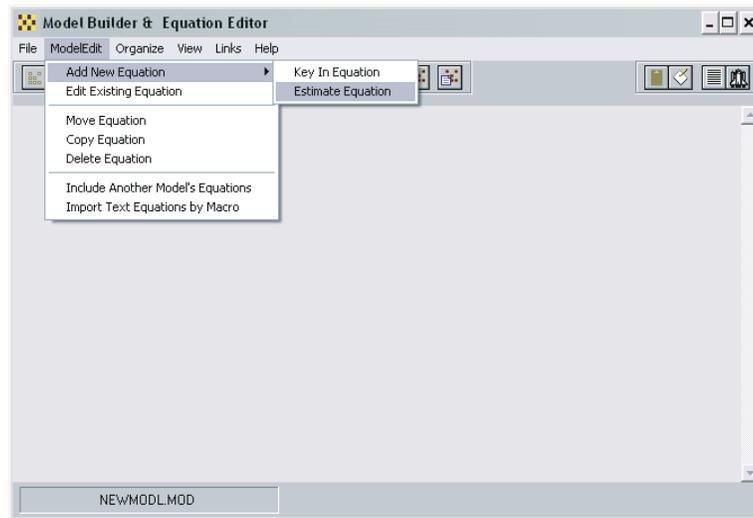


Figure 6-3. Estimating the First Equation

building environment, and you will see a screen display similar to that shown in Figure 6-3.

Looking at Figure 6-3, it is evident that equations can be “added” in two ways, either by clicking on **Key In Equation** or on **Estimate Equation**. If you click on the latter, you will next see the Parameter Estimation screen familiar from Chapter 5, so that the only issue is the approach. The estimation itself still proceeds as before: estimate the equation, then put it into the model (into the NEWMODL.MOD file) as described in Chapter 5.

However, before considering the alternative, **Key In Equation**, look once again at Figure 6-3. Note that there are actually two further options as well: **Include Another Model’s Equations** and/or **Import Text Equations by Macro**. Models can be constructed equation by equation, or by using as building blocks sets of equations from previously developed models. If you have a model created in some other, non-MODLER environment, such as AREMOS or TROLL or EPS, use the option **Import Text Equations by Macro**. You will probably first need to edit the textfile containing these equations, because of syntax differences in the way equations are represented in those environments, but otherwise transferring such models into MODLER is quite straightforward. The process is described in further detail starting on page 152.

Deleting Equations

Now suppose that you want to review the equations of your NEWMODL model onscreen and delete some of them. Once equations are included in a model, one of

NEWMODL.MOD

Specify Model Equation to DELETE

Equation # 1 UnNormalized

$$\text{LN(EMPL)} = 0.22269 * \text{LN(GNP)} + 0.70029 * (\text{LN(EMPL(-1))}) - 0.24703$$

Selection Criterion:

Variable

Equation No

Equation No: 1

Previous

Next

OK

Cancel

Figure 6-4. Equation Deletion Form

the choices under **ModelEdit** is **Delete Equation**. If you click on this choice, you should next see displayed the form shown in Figure 6-4. There are two ways to find an equation in order to delete it. If you know the equation number, simply click the radio button labelled “Equation No” found under the heading “Selection Criterion” at the bottom left of the form. Then at the right enter the number and press the <Enter> key. Press OK and you will be asked if you really mean to delete the equation. If you then click “Yes” the deed will be done. Alternatively, if you click the radio button labelled “Variable” you will be given the opportunity to specify the name of some variable that presumably appears in the equation you wish to delete. In this case, using the Previous and Next buttons, you can scroll through all the equations in the model that contain that variable. When you find the equation to delete, press OK.

Inserting Equations From The Keyboard

But generally, at this stage, the question is how to add equations. One way of including identities and technical relations is to type them in interactively. Choosing **ModelEdit | Add New Equation | Key In Equation** causes the form shown in Figure 6-5 to be displayed. This form presumes that the equation to be added will be the last in the model (thus it shows an equation number equal to the number of equations currently in the model plus 1), but this is just the default. Another choice can be made instead.

Note the **Index**, **Find**, and **Macro** buttons on the form, which provide you with access to names of variables, based upon the names of series in any open data banks or Memory File, plus the ability to capture equations or equation fragments from existing macro files. Or you can of course simply begin typing in the equation.

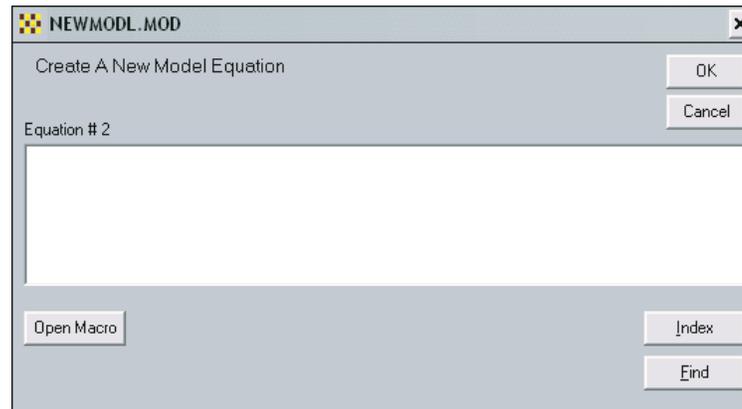


Figure 6-5. Key In Equation

Generally, model equations should obey all MODLER syntactic conventions and be in normalized form. However, there is some latitude: you can type in unnormalized equations, although you will need to normalize them subsequently—using the `NORMALIZE` command—before attempting to compile or solve the model.

MODLER's mathematical, logical, and relational operators can be used. Most functions can also be used, but there are certain ones that cannot: `TIME`, `MEAN`, `SUM`, `SDEV`, `DUM`, and others that are fundamentally defined on a fixed historical time interval—note that the values of these all depend upon the particular observation date range in force at the time they are executed. But their evaluated values *can* be entered as constants.

Including Equations From A Macro File

If you already have many equations in machine-readable form—which could be the case if you are importing a model that has been developed in another system—it is often a good idea to put them into a macro file, which you then can edit with a plain text editor, one that does not include control characters in the text. This macro can then be loaded using the commands:

```
EDIT modelname
INCLUDE MACRO macroname
SAVE
```

located within another, driver macro file; remember that one macro file can execute another. Alternatively, if the MOD file has never been created, it is possible to add the command:

```
DEFINE MODEL modelname
```

to the beginning of the Macro file and:

DESC: model description
FREQ=frequency
SAVE

to the end, thus both defining the model and specifying its equations. The model can then be created from inside MODLER at the Central Control screen simply by typing RUN, followed by the macro name.

There are several possible variations on this theme. For example, the entire set of commands to create a model—containing a mixture of both regression commands and pre-defined equations—may be formed into a macro file. See the chapter appendix for such an example; this file is also provided as a demonstration macro, called DEMMAKE.MAC.

However, there are certain limitations to this approach, particularly in the case of large, complicated models. The command **INCLUDE MACRO** appends macro file equations to the *end* of a model, thus it cannot be used to selectively copy individual equations, or groups of equations, from that file to the model. In contrast, the command **INCLUDE MODEL** can be used this way, once the model exists. It is therefore possible, although slightly tedious, to break a macro file into several smaller files, each containing a group of equations, to create several MODLER models, which then can be selectively copied using **INCLUDE MODEL** to form the final model.

Defining New Variables For Use In The Model

It is sometimes useful to create new variables for use in a model, or to display in model report tables. For example, you could use an explicit variable GNPGR in order to impose assumptions about quarterly GNP growth. You only need to provide an equation that shows how the new variable relates to some existing series, such as:

$$\text{GNP}=\text{GNP}(-1)*(1+0.01*\text{GNPGR})^{**0.25}$$

This equation assumes that GNP growth will be measured by an annualized percentage rate—as is otherwise calculated by MODLER's APCT function.

Given the simplicity of the present model, it may not be obvious why one might wish to create a relationship such as this. However, in the context of a larger model, it is easy to imagine technical relations that exhibit this type of structure, allowing you to define policy handles.

Note also that you can use logical constructs (which if true evaluate as 1 and if false as 0); for example, consider:

$$X = (X1.GT.0)*X1 + (X1.LE.0)*X2$$

Where X1, X2, and X are model variable names. Since the logical expressions are mutually exclusive, X will take the value X1 or X2, depending upon the value of X1. You can mix mathematical, logical, and relational operators in model equations to form quite complex objective functions, if you wish. Of course, when you create such constructs you must test them carefully to make such that they behave as you expect during the solution of the model.

Editing and Normalizing Equations

In the display shown in Figure 6-2 above, the equation for EMPL appears with the heading: **UNNORMALIZED**. This term indicates that the left-hand side of the equation contains a function or algebraic expression (in this case a logarithm)—and that the equation must be defined as solvable (normalized) for a specific variable before the model as a whole can be compiled and solved.

In order to change an existing equation, first click on the menu item **ModelEdit** of the Model Builder & Equation Editor screen, then click **Edit Existing Equation**. You should next see displayed the form shown in Figure 6-6:

Using this form, there are two ways to select the particular equation you wish to edit. First, you can click on the radio button labelled “Equation No” and then insert the location number of the equation; see Figure 6-4 for an example that illustrates selection by equation number. Alternatively, as illustrated in Figure 6-6, if you choose Variable as the selection criterion, and if you enter the name of a model variable in the textbox labelled “Variable,” using the **Previous** and **Next** buttons you can then scroll through all the equations in which it appears. Or, in fact as a third option, the **Variable List** button, if pressed, will display the complete list of variables for the currently attached model. If you double click on one of these, the first model equation containing that variable will be selected immediately and will appear in the center panel of the form displayed in Figure 6-6. The **Previous** and **Next** buttons can then be used to discover other equations containing it.

When you have chosen the equation to edit, you can edit it in place. Pointing with your mouse at the textbox containing the equation, if you click the right mouse button, a floating edit menu will appear. In addition, all the standard editing keys

Figure 6-6. Equation Editor Form

operate in the usual way. When you finish, press OK. MODLER will respond by displaying the form shown in Figure 6-7:

You can use the edited equation to replace the original equation, or put it into some other location in the model, thus leaving the original equation unaffected. Note also that you can normalize the equation on any contemporaneous variable it contains by stating that name in the textbox on the last line of this form; in principle, there are rare cases that this normalization will not work. When you have made your choice(s), press OK; until you press the OK button, the model is not affected by any changes you may have made.

An Alternative Normalization Procedure

Alternatively, it may be more convenient to request normalization at the time each estimated equation is put into the model with the command PUTEQ. For example, assume that hourly wages, WAGEPH, are to be brought into the model as a lagged function of prior movements of wages and GNP over the preceding 8-12 quarters. To estimate the function, type:

```
LN(WAGEPH)-LN(WAGEPH(-4))=F(LN(WAGEPH(-4)),
LN(WAGEPH(-8)),LN(WAGEPH(-12)),
LN(GNP),LN(GNP(-4)),LN(GNP(-8)),
LN(GNP(-12)))
```

Then put the result into the model by typing:

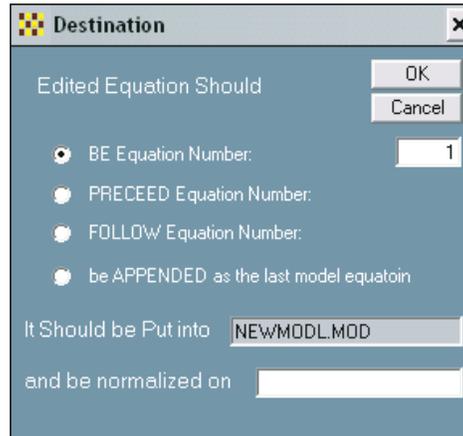


Figure 6-7. Disposition Form

PUTEQ, NORMALIZE WAGEPH

Using commands or the menus and dialogue boxes both will get the job done. It is your choice which to use, and there are occasions that one or the other of these methods will be the most convenient. The aim is to provide you the choice.

Compiling The Model

In the context of MODLER a model must first be compiled before it can be solved or used in any meaningful way. That is, it must be put in a form that the computer understands. This *compile* step is invoked either from a menu selection such as **Organize | Compile Model | Execute** of the Model Builder & Equation Editor screen or from the Central Control Screen, using the command:

COMPILE

with or without associated, optional modifiers that affect the specific way in which it is compiled.

Basically, what occurs in response to such a command is that MODLER first sorts the model into its most recursive ordering and then translates it into machine-readable code, writing this code to a file and giving the file the same name as the model,

but with the extent CMF (Compiled Model File). You will not be able to view or otherwise interpret the CMF file, but if necessary you can access another file that is simultaneously created—called ORDMOD.TMP—that is the same as the original model file except that its equations have been *re-sorted* in the model solution order dictated by the Compile process. Normally, only for the most complex applications will it be necessary for you to be more than vaguely aware of these files, but if you are interested they are described in some detail in the section of MODLER’s online help that focuses on readying a model for solution.

Editing Equations and Correcting Mistakes

A MODLER feature that you should be particularly conscious of is that a CMF file is associated with a particular “state” of the model’s MOD file. If, once you have compiled a model, you then change the MOD file in some way, for example by adding or editing an equation, MODLER will automatically delete the existing CMF file *if it exists within the same directory as the MOD file*. The reason for this action is to keep you from working unintentionally with an older generation of a model than the one the current MOD file portrays. Therefore, if for some reason you wish to prevent MODLER from deleting your only copy of a CMF file, you must move or copy it to some other directory prior to making any subsequent change in the MOD file. However, if you do this it is then up to you to make sure that you know the characteristics of any solution you make with that model. You also need to be aware that a copy of the current CMF file for a model must be in the same directory as the MOD file in order to be able to solve and otherwise use the model.

Of course, before attempting to compile and solve a model, it is always wise to make a full listing of the model and then check the equations carefully. To make any necessary changes in model equations after they have been entered, including correcting mistakes, you must once again edit the model file, as described above. Alternatively, you can also list the equations in your model from the Model Builder & Equation Editor screen by clicking on **View**, as illustrated by Figure 6-2 above.

View can also provide you a *tentative* pre-compile classification of the endogenous and exogenous variables. But bear in mind that here these terms should not be interpreted in the statistical sense, but instead simply as referring to the dependent and independent variables of a set of equations. In this specific context, before a model has been compiled, MODLER interprets its endogenous variables to be those that can be identified as explained by a particular equation, and the exogenous as those that cannot, hence the tentativeness of this classification. However, such a listing, when different than you expected, can also alert you to the fact that you either may have misnormalized an equation, failed to normalize an equation, or possibly even left out one or more equations that you intended to include.

Furthermore, do not ignore that one of the fundamental analytical strengths of MODLER is that it permits you to document your model at each stage of the model's development. You can preserve machine-readable or hard copy regression output. You can do the same for the model as a whole, by listing the set of equations before and after any significant changes. MODLER allows you to generate a complete audit trail of the entire model building process, if you wish, and once you have organized to do this, it can be done with a minimum of effort on your part. MODLER's documentation features are an integral part of the software's design.

Changing The Model's Structure

Finally, you should note that normalization can have an important effect on the use of a model. A model can be restructured by renormalizing one or more equations so as to put different series on the left-hand side. This may change the list of series that are treated as exogenous for solution purposes. For example, the model in this chapter can be normalized to answer questions about the pattern of GNP growth needed to make employment follow any given target path:

```
EDIT NEWMODL  
>*FIND #1 (original equation for EMPL)  
>*NORMALIZE GNP  
>*FIND #4 (original equation for GNP)  
>*NORMALIZE GNPGR  
>*END
```

The equation for employment becomes an equation for GNP, and the former GNP equation may now be inverted to compute implied quarter-to-quarter GNP growth. Solving the model in this form will compute GNP growth consistent with given target paths for employment.

Extensively Documenting a Model

As you have been using MODLER you may have noticed in passing that the first item in the **Model** dropdown menu list on the Central Control screen is **Select Model**. If you should click on this item, you will then see a form similar to that shown in Figure 6-8, although it may not show the model as *Not Yet Catalogued*, particularly if you have already compiled it and loaded a solution file at the point you display this form.

Figure 6-8 Model Selection Form

The purpose of this particular form is to permit you to select and manage effectively the models that you use, by the process of automatically associating with each model a variety of files related to it, some of which we have yet to consider. Notice that the textboxes that are labeled *Model Choices*, *Solution Files*, and so on, down the left-hand side of the form, are all dropdown text boxes and are therefore capable of containing multiple lines of information for each of these labeled categories. Furthermore, what is listed on this form is in fact closely keyed to a third type of model-related file that has not so far been mentioned but which MODLER automatically creates, in addition to the MOD and CMF files that have been described earlier.

This third file is called a DEF file and it is created when you first define the model, in fact at exactly the same time that the MOD file is created. From that point on, as you go about your model building tasks, MODLER loads this file with relevant information about the model. It is also located in the same directory as the MOD (and CMF) files for that model and, in addition, the filename is the same as for these other two files, so that a particular model's DEF file is easily identified. This DEF file, when fully populated, contains a sectional listing of all the files associated with the individual model, each documentation section corresponding to one of the textboxes on the Model Selection screen. Should you be interested in the technical details, this file is formatted in the same way as any Windows INI file.

When, as a result of your clicking on **Select Model**, this form initially appears the specific configuration and what is displayed will then depend upon current circumstances. For instance, if you have not previously attached a model, using any of the model access commands, including *Use*, *Define Model*, *Edit* and so on, the model choice that is first displayed in the topmost text box will be the first model alphabetically by name. Of course, it is possible that, at that moment, you will have only a single model, but if you have more you can select a particular model by

choosing from among those in the *Model Choices* drop down text box. Furthermore, whenever a particular model is selected, the entries in the other list boxes on the Model Selection Screen will change conformably so as to identify the individual solution files, banks, macro files, document files, etc. that are associated with that model. However, whenever this occurs in the context of the *Model Selection* screen, this listing operation is simply a “read only” process; nothing happens to the model at that point. In particular, *unless it was attached previously*, the model identified in the Model Choices text box is *not* attached to MODLER: at that point, its name and the files related to it are simply being listed for your information. You must in this case actively select the model before it will become attached.

Incidentally, this description is actually provisional—because of the fact that not *all* files related to a given model will necessarily be listed. Some of the sectional listings are created automatically as you develop each model; however, other listings *must* be created and maintained by you, inasmuch as you will be the final authority on what set of files actually comprise *all* those associated with each of your models. MODLER has only a limited ability to identify these: obviously, the MOD and the CMF files share a common name, and if you have created solution files using the same computer, MODLER will also be able to capture information about these, as well as the names of any data banks used in their creation. However, other files, such as model-related macro files, are not immediately linked to a model. Furthermore, if you simply copy model-related files to the hard disk of a specific machine from some other machine, the program has no way of knowing that these are related to a specific model. Consequently, although MODLER attempts to make the requisite linkages to the best of its ability, at the end of the day you are responsible for identifying which files are associated with a given model, using facilities that will be described both below and in subsequent chapters of this user guide.

There are therefore certain caveats involved, but to give you an idea of how model selection works let us assume, for the moment, that each model is fully documented. Then, in order to attach a particular model, once it has been selected and its related files are displayed, you must subsequently press the **Attach Model** button. In response, two things will happen: first, the model will be attached to MODLER. Second, the remaining buttons on the *Model Selection Screen* will become enabled.

However, before you press the **Attach Model** button, look first at the bottom left-hand-corner of the *Model Selection Screen* and observe the check boxes there. Notice that these imply that you are able not only to attach the model, but simultaneously, at your option, also whichever solution file is currently shown in the text box labeled *Solution Files*, plus all the data banks that are listed in the drop down text box entitled *Data Banks*. As we will discover in the next chapter, only one solution file can be attached at a time, but up to 15 banks can be simultaneously open for access. Be careful here: if you open banks at this point, it will usually be best to have previously closed all banks prior to displaying the *Model Selection Screen*: if

other, unrelated banks are already open and contain data series with the same name as any used in the model, there is a risk that these other data series could be retrieved, instead of those in the model-related data banks.

Notice also the check box that, by default, is not checked; it is labelled “Automatically Cull Unavailable Files.” If this box is checked, whenever a model is attached using the *Model Selection Screen*, and by pressing the **Attach Model** button, MODLER will quickly review all the model-related files that are listed in the pertinent DEF file and will delete from this file any file names for which an associated file cannot be found in whichever directory/folder it ostensibly should appear. If a filename appears on its own, without a path specification, MODLER will infer its directory location on the basis of the default directory settings. If a filename appears complete with path, then the location of that file will be understood to be that explicitly specified by this path.

As a beneficial characteristic, what is implied by this culling convention is that if you delete an originally relevant file from one of the MODLER directories, it will automatically be culled from the model’s DEF file the next time you use the *Model Selection Screen* to attach that model; for example, you might independently consolidate several transformation macro files into a single file, deleting the extra files from the directory. These surplus files will then automatically be culled from the model’s DEF file. Of course, on the negative side, if you have moved a file to some other directory and MODLER cannot find it when attaching a related model, the program will cull its name from the DEF file for that model if this check box is checked. This culling will also occur if the directory cannot be found, as may happen if the drive originally used was removable and has since been removed. However, in this case, as a safety measure, the culling will take place in two steps: initially, an error message will appear, indicating that the directory is unknown, at which point the directory portion of the path will be deleted, leaving only the filename. The next time the model is attached, this file name will then be culled if you have not in the meantime put a file of that name in the relevant default directory for that file type.

An obvious question at this point is, how do you add related model files, in order to create and maintain the other listings? It was mentioned earlier that whenever a model is attached, using the **Attach Model** button, all the greyed-out buttons on the *Model Selection Screen* become enabled. Once these are enabled, you can use them to add files to the model DEF file for a particular (attached) model. With the exception of the **Help**, **Close Form**, **Attach Model**, and **Model Details** buttons, pressing the buttons individually will in each case cause an appropriate file selector form to appear. This form will be keyed to the file type and will display the relevant files located in the relevant directory, thus allowing you to select a pertinent file. Alternatively, if you press the **Browse** button on that form, you will be able to search for additional files of that type in any directory and on any accessible drive. Files

that are found in the default directory for files of that type will be displayed without any drive or directory information. However, if you browse for the filename, then the complete path plus filename will be shown, thus indicating that the file is not located in a default directory.

Should you later move the file to another directory, you may need to add it again to the DEF file, for if you actually move it, it cannot then subsequently be found in its original directory. In this case, the fully pathed file name may be culled, inasmuch as (with the exception of the MOD, DEF, and CMF files) there is no automatic general mechanism to associate a file name with a model. The exception to this rule, as mentioned earlier, is if you move the file to the specified directory for that filetype, in which case, once its original path details are removed, it will be found in the directory to which you have moved it whenever the model is next attached.

An additional, related facility is the capability to archive or export a model, options that can each be selected by first clicking the **Model** menu item of the *Central Control Screen*. Then choose either **Archive Model** or **Export Model** from the listed elements. Actually, the only significant difference between these alternatives is conceptual, for “export” in this context refers to copying model and related files for use at another MODLER installation, whereas “archiving” refers to backing up your files locally. Obviously, the difference fundamentally consists of the possible later use of these files in either the same location or at another location, although in practice different storage locations and even different types of storage media may be involved in the archiving versus export process. In contrast, exporting a model and related data for use in a foreign, non-MODLER software context will generally involve instead, or as well, the creation of a text representation of the model and various types of data exchange files, including perhaps ASCII or spreadsheet files.

Generally, when archiving or exporting a model in a MODLER-only context, there are two important rules that must be obeyed. First, a model cannot be archived or exported to a destination directory that already contains a model: you must first delete all model-related files from this directory and its subdirectories. In any case, attempting to copy a model overtop another model (or even another copy of itself) is not a good idea inasmuch as file version conflicts and other such problems can easily occur. Second, you should make sure that any files you are attempting to archive or export are not at that time being used by either MODLER or another program. In particular, you should close all banks and detach all model files before you attempt to archive or export a model.

Notice that MODLER simply copies a model and its files, as determined by its DEF file, and does not attempt to “zip” or otherwise compress or contain these files in any type of library file. However, once the model and its related files are all copied to a particular directory, then it is possible to use WinZip, Pkzip or other file compression or “library” software to create some type of single file containing all

these files, possibly in a compressed format. Obviously, before you take the step of compressing these files, or otherwise containing them, you will be able to inspect the copied files to make sure that all the relevant files have been copied.

Other Options

The specific purpose of this chapter is to describe a number of the MODLER model building facilities. For this reason, very little attention has been paid in this presentation to the model as a representation of an economy or economic process. In contrast, you might now find it interesting to consider the construction of an actual model and some of the associated issues. If so, look on the www.modler.com website for the documented description of Klein Model I. The construction of this model is considered in the document found there entitled *Building and Using a Small Macroeconometric Model: Klein Model I as an Example*. Klein Model I is of course a very simple, now somewhat dated model. However, the model has been widely used as a teaching tool. Both its familiarity and its characteristics make it an interesting model to examine, as a follow on to the present user guide.

Appendix

Macro File To Build The Demonstration Model

```
define model DEMODL
desc: How GNP growth affects employment and hourly wages
frequency=QUARTERLY
```

```
“ Identities and imposed equations
```

```
UNRATE=100*(1-EMPL/LABF)
LABF=0.4205*EMPL+0.6198*POPWA-23.765
GNP=GNP(-1)*(1+0.01*GRGNP)**0.25
end
```

```
“ Regressions for estimated equations
```

```
ln(EMPL)=f(ln(GNP),ln(EMPL(-1)))
puteq,norm EMPL
LN(WAGEPH)-LN(WAGEPH(-4))=F(LN(WAGEPH(-4)),
LN(WAGEPH(- 8)),LN(WAGEPH(-12)),
LN(GNP),LN(GNP(-4)),LN(GNP(-8)),LN(GNP(-12)))
puteq,norm WAGEPH
```

```
“ Compile and build solution file
```

```
COMPILE
MAKE SOLVFILE DEMODL 197501-198402
```