A User's Manual for the Mongolia CGE Model

David Roland-Holst, Enkhbayar Shagdar,
and Dominique van der Mensbrugghe

FAO

June 25, 2013

Questions and inquiries should be addressed to:

Professor David Roland-Holst
Departments of Economics and
Agricultural and Resource Economics
338 Giannini Hall
University of California
Berkeley, CA 94720
Tel: 1-510-643-6362
Fax: 1-510-524-4591
dwrh@are.berkeley.edu
http://are.berkeley.edu/~dwrh

Enkhbayar Shagdar (Ph.D.)
Associate Senior Researcher,
Research Division and External Relations Division
Economic Research Institute for Northeast Asia (ERINA)
13th Floor, Bandaijima Building, Bandaijima 5-1, Chuo-ku,
Niigata-city, 950-0078, Japan
Tel: (+81)-25-290-5545
Fax: (+81)-25-249-7550
E-mail: enkhee@erina.or.jp
Url: http://www.erina.or.jp

Acknowledgements

The authors would like to acknowledge the support and help from the Asian Development Bank, in particular Teruhisa Oi. Strong cooperation was provided by the Mongolian counterparts, and we are very grateful to the Government of Mongolia for their interest in this capacity building exercise.

1

Introduction

The following document describes an economic policy decision tool that has been developed to support more forward looking and inclusive development policy by the government of Mongolia. The primary objectives of this approach are to promote evidence based policy and improve information available to public and private stakeholders. The tool takes the form of a flexible prototype computable general equilibrium (CGE) single-country model that has been applied to a number of countries. The model is designed to be used in either comparative static or dynamic mode. It has also been designed to capture the structural features of most economies—though it has been adapted in each case for the specificity of the target country. It is based on a relatively standard CGE backbone that includes the following features:[1]

- Nested CES production structure
- Multi-output production
- Multiple factor types (labor, capital, land and sector specific resources)
- Detailed income distribution
- LES consumer demand system
- Nested trade system with multiple trading partners on both the import and export side
- Segmented trading regimes, with tariffs regime-specific (for example for duty-drawbacks)
- Various possibilities regarding factor mobility
- Domestic trade and transport margins
- Exhaustive fiscal instruments
- Flexible closure rules

Other features of the model include the potential for segmented labor markets, non-market clearing wage formation and implementation of recursive dynamics—with factor accumulation and productivity.[2]

[1] This documentation draws directly on van der Mensbrugghe (2005).

[2] Country-specific features have included scale economies and oligopolistic pricing, efficiency wages and models of wage bargaining.

The model is implemented in GAMS.[3] As coded, the model is essentially dimensionless and the input data files determine the actual size of the model. Users are urged to develop the broadest possible set of accounts for the model since the model comes with a front-end aggregation facility than can reduce the size of the model to focus on sectors of particular concern.

Unless the model specification or model output is to be modified the typical user will only have to work with two GAMS files—the input data file containing the dimensions of the model and the key model parameters and data and the simulation file that determines the nature of the simulation(s). The output of the simulation is a data text file—commonly known as a CSV (with data values separated by commas) file. It is formatted in a database style format that renders it easy to process—for example using Excel's pivot table feature.

The rest of the document is organized as follows. The second section describes the data files. There is a base data file that contains the base SAM. Preparing this is typically the most difficult part of any exercise, but fortunately, is only done at the beginning or when an update is needed. The other main data file is the aggregation and parameter file. The third section describes the key model files. These model files are rarely modified by users except occasionally the output routines when new model results are needed. The fourth section describes the simulation files. After the parameter file, these are the most important since this is where users design the particular simulations they wish to run. Output and report processing are detailed in section 5. Annex A provides a generic description of the use of pivot tables.

[3] See Brooke et al. (1998) or visit the GAMS web site at www.gams.com for further information regarding GAMS.

2

Data files

Each country model requires at least two data files. The first is the base data file that contains the Social Accounting Matrix (SAM) and auxiliary data for a given base year. This document assumes that the base data file exists and only describes its content. Putting together a SAM is an arduous task that fortunately is done only on a periodic basis.[4] The second data file is a user-prepared data file. It has two purposes. First, it allows the user to aggregate the original data to a desired level of aggregation (including using the fully disaggregated data) and second, it contains values for the model's key elasticities and other required data.

The SAM file

The SAM file is a GAMS-conformable file that contains the initial dimensions of the data, the SAM and auxiliary data files. The SAM is not fully disaggregated. For example, if there are multiple trading partners the flows with these trading partners will be represented in the respective rows and columns. However, there is only one account for tariffs even if these are distinguished by trading partner. Similarly, the commodity accounts of the SAM are based on the Armington consumption of goods. The disaggregation into domestic and import components is provided in an auxiliary table.

SAM accounts

Table 1 provides a description of the functional SAM accounts. The SAM has three types of accounts—'user-determined', 'constant' and 'constant, aggregate'. The user-determined accounts typically, though not necessarily are multi-dimensioned accounts. Accounts that are constant have hard-coded labels in the model code and cannot be changed (without making modifications to the model code). Accounts that are constant and aggregate have hard-coded labels, but the information in the SAM is typically ignored because auxiliary accounts are needed to initialize model variables and parameters. For example, the "imptx" row has total import tariff revenues, but not their allocation across trading partners. Unless the tariff rate is uniform across trading partners, additional information is required to initialize bilateral import tariff rates.

Activities: Activities represent the production accounts. The intersection of the commodities' rows with the activities' columns represents the input-output matrix. In the SAM these are in user prices and at the Armington level of aggregation. The intersection of the activities' columns with

[4] There are a number of resources available on SAM construction. See for example Pyatt and Round (1985), Reinert and Roland-Holst (1997) and Robinson et al. (1998).

the commodities' rows represents aggregate commodity supply for both domestic absorption and exports.

Table 1: SAM accounts

	Description	Label	Type
	Activities	i	User-determined aggregation
	Commodities	k	User-determined aggregation
Produc-tion factors	Labor categories	l	User-determined aggregation
	Capital types	kt	User-determined aggregation
	Land types	lt	User-determined aggregation
	Sector-specific resource	nr	User-determined aggregation
	Enterprises	e	User-determined aggregation
Final demand accounts	Households	h	User-determined aggregation
	Government	"Govnt"	Constant
	Investment	"Invst"	Constant
	Change in stocks	"DelSt"	Constant
	Domestic margins	"Margn"	Constant
	Regions	r	User-determined aggregation
Fiscal accounts	Production taxes	"prdtx"	Constant
	Value added taxes	"vattx"	Constant, aggregate
	Export taxes	"exptx	Constant, aggregate
	Taxes on factors of production	"fcttx"	Constant, aggregate
	Subsidies on factors of production	"fctts"	Constant, aggregate
	Other indirect taxes	"indtx"	Constant, aggregate
	Import tariffs	"imptx"	Constant, aggregate
	Import subsidies	"impts"	Constant, aggregate
	Household subsidies	"hldts"	Constant, aggregate
	Direct taxes	"dirtx"	Constant
	Revenues from non-tariff barriers	"ntbY"	Constant, aggregate
	Balance of payments account	"BOP"	Constant
	Accounting total	"Total"	Constant
	Accounting residual	"Resid"	Constant

Commodities: Commodities are distinguished from activities, though in many SAMs there will be a one-to-one mapping from activities to commodities. Commodities are consumed by domestic agents and are a combination of domestic and imported goods. The diagonal portion of the commodities section represents the consumption of goods and services needed for the domestic trade and transport sector. This is the aggregated trade and transport sector with all transportation nodes combined—from farm/factory gate to domestic markets, from farm/factory gate to domestic ports/frontiers for export, and from ports/frontiers to domestic markets for imports. Domestic supply is augmented by imports that will appear in the regional rows of the commodity columns. Many of the domestic distortions will also appear in the commodity columns in the relevant tax rows. Commodities are aggregate accounts since they represent Armington demand. Additional information is needed to separate Armington demand into its respective domestic and import components.

There is one special SAM account, labeled "Margn" that has no data in the SAM but must be included in the SAM labels. It is used by the model as an aggregate Armington agent for the separate cost functions for the domestic margin accounts. There is a separate margin account in the SAM for each commodity. For the purposes of determining the domestic/import split of the demand for trade and transport services, an aggregate Armington agent is created and is labeled "Margn". Thus the label must be included with the SAM definition, since the set of Armington agents is derived from the SAM, but the account will never appear explicitly in the SAM.

The balance of payments account, 'BoP', provides a snapshot of the annual flows of both current and capital accounts. The trading partner section of the 'BoP' column represents the aggregate gross export flows at world prices to the various trading partners. The trading partner section of the 'BoP' row represents the aggregate gross import flows at world price from the various trading partners. In other words, the net trade balance by trading partner is the difference between the 'BoP' column and the relevant 'BoP' row.

The first part of the data file contains the various dimensions of the SAM—translated into GAMS, this contains the set and subset definitions. Almost all of the sets are a subset of the master set, i.e. the full SAM accounts. Table 2 provides a complete list of all the sets and subsets. The meaning of most should be relatively self-evident. The following provides a few additional comments.

Armington agents. The Armington agents represent a subset of SAM accounts. Any domestic agent consuming commodities is an Armington agent—thus they include all activities and all domestic final demand accounts (households, government, investment and stock changes). There is an additional Armington agent that is a SAM account and that is the 'constant' Armington agent designated by 'Margin', see above.

The regional dimension of the Mongolian SAM corresponds to three regions, China, Russia, and Rest of the World.

There are three auxiliary dimensions that are used by the auxiliary data accounts and the model. The first is labeled 'tr0'. This set contains the number of individual trading regimes. If all agents face the same tariff regime, then only a single trading regime is needed. In other cases, it is useful to distinguish across Armington agents and segment them into separate regimes. For example, many countries have duty-drawbacks for certain industries that then effectively face zero-tariffs. The set 'tr0' is user-determined and can be aggregated.

In the case of Mongolia, three regimes have been identified—labeled 'INT', 'CON' and 'INV'. The first represents import demand by activities, including the domestic margin sector and change in stocks. The second represents current final demand expenditures and includes import consumption by households and the government. The third, 'INV', represents consumption by the investment account. The mapping from Armington agents to trade regime is done with the set variable 'mapa', see below.

The second auxiliary dimension, labeled 'pl' represents the different price levels at which commodities can be evaluated. The price levels themselves are context specific (activities, imports, exports, etc.). The four dimensions are:

Base	The basic price of a good. For domestically produced goods—either sold domestically or abroad—this is the producer price. For imports, it is the world price.
Tax	The base price of a good with some tax. This is detailed further below.
Marg	The price of the good including the trade and transport margins.
User	The user-price of the good including final taxes.

The third auxiliary dimension refers to the different import distortions (at the border). They are:

TAR	The standard *ad valorem* tariff.
SUB	Import subsidies
NTB	The *ad valorem* equivalent of non-tariff barriers.

The SAM data

The basic SAM is provided in a single GAMS table. This is followed by all of the auxiliary data needed to fully initialize and calibrate the model. Table 3 provides the list of all the data tables in the SAM file.

Table 3: SAM data variables in input SAM file

Name	Description	Unit of account
sam00(s0,ss0)	Base year SAM	LCU
xddY00(pl,k0,a0)	Domestic demand for domestic goods	LCU
xmtY00(pl,k0,a0)	Domestic demand for imported goods	LCU
XMY00(pl,tr0,r0,k0)	Imports by trading regime and by trading partner	LCU
IMPY00(tr0,r0,k0,md)	Revenues generated by import distortions	LCU
XEY00(pl,k0,r0)	Exports by region of destination	LCU
hldtsY00(k0,h0)	Household subsidies in value	LCU
fcttx00(v0,j0)	Taxes on factors of production	LCU
fctts00(v0,j0)	Subsidies on factors of production	LCU
tk00	Aggregate capital stock	LCU
tpop00	Total population	Natural units
popsh00(h0)	Population shares by household	Decimal shares
erd0	Dollar exchange rate	LCU/USD
ppp0	PPP exchange rate	USD to PPP conversion
eta00(k0,h0)	Base year income elasticities	Natural units
labvol00(i0,l0)	Labor volumes	Natural units

All of the SAM-related data will be in local currency (LCU). The unit of account does not have to be the base currency itself, but some multiple of the base currency, for example millions or billions. The user has the opportunity to scale the model data when creating the aggregate file for input into the model. Both the total population and labor volumes are in natural units. They can also be scaled into thousands, millions or some other scale. Note that if labor volumes are not available, the 'labvol00' table would typically contain the labor remuneration table from the

SAM. By definition, this will imply that all wages are initialized to 1. The exchange rates, not currently used, would only be used upon output.

The base year SAM as noted above does not contain all of the structural detail to calibrate and initialize the model. It is nonetheless consistent accounting-wise, and will be output using the same dimensions at the end of each simulation.

XDDY. The first auxiliary matrix contains the demand for domestically produced goods by the domestic Armington agents. It has three dimensions—price level, commodity and agent. The data only covers three price levels, corresponding to two price wedges. The 'base' level price is the producer price. Relative to the model, this corresponds to the price 'PD'. The second price level, 'Marg', corresponds to the margin-augmented price of domestic goods.[5] In the context of the model, this corresponds to the following formula:

$$(PD_k + PTMG_k \tau_{k,a}^{mg,D}) XD_{k,a}^d$$

The 'base' price is uniform across all Armington agents, but the margin-augmented price is allowed to be agent-specific. In other words, it allows for the possibility that oil delivered to large industrial users has a lower margin than oil delivered for household consumption.[6] The difference between the 'Marg' value and the 'Base' value is the value of the trade and transport services used to deliver the commodity. The final price level is the 'User' price. This corresponds to the final end-user price inclusive of the sales tax.[7] The model formula is given by:

$$(1 + \tau_{k,a}^{cd}) (PD_k + PTMG_k \tau_{k,a}^{mg,D}) XD_{k,a}^d$$

The difference in the values between 'User' and 'Marg' is the tax revenue generated by the sales tax. There are consistency requirements between the values in XDD and the SAM. The 'User' price of domestic demand plus the 'User' price of imports will have to add up to the overall Armington demand as provided in the SAM. The margin revenues (including those from the other transportation nodes) will have to add up to the margin information contained in the SAM. And the tax revenues, added to the sales tax on imports, will have to add up to the 'indtx' row of the commodities columns.

XMTY. The second auxiliary matrix, XMTY00 contains the demand for the aggregate import composite, i.e. aggregated over all regions of origin. Armington agents are then aggregated into one of the possible trading regimes and aggregate import demand is allocated across trading partners by trading regime. It is possible to have as many trading regimes as Armington agents, though the data requirements are daunting.

---

5 The data at the different price levels must be present even if the underlying price wedge is 0. An alternative would be to store the price wedge revenue itself and only the base values. While this would potentially save on storage (and makes eyeballing the wedges somewhat easier), it has the possible disadvantage of losing data precision if the wedges are sufficiently small.

6 While this possibility exists in the model, in actual practice most national data would not distinguish margins by end-user.

7 Note that the end-user price for households is not the Armington price, but the Armington price adjusted for a household- and commodity-specific tax/subsidy, see below.

The XMTY table has only two price levels. The 'Base' price level corresponds to the aggregate value of imports (i.e. across all trading partners), inclusive of all import distortions, margins, and value added taxes. The 'User' level corresponds to aggregate imports including the domestic sales tax on imports. The latter is differentiated from the sales tax on domestic goods. The 'Base' price level corresponds to:

$$PMT_{k,a} XMT_{k,a}$$

The 'User' price level corresponds to:

$$(1+\tau^{cm}_{k,a}) PMT_{k,a} XMT_{k,a}$$

The revenues from the sales tax on domestic and import goods must add to the total in the 'indtx' row in the commodities columns.

XMY. The XMY00 matrix contains the detailed import transactions (though not all!). It is represented by four indices—the price level, the trading regime, the commodity and the region of origin. For each combination of trading regime, commodity and origin country, there are four price levels corresponding to three wedges. The first level is the 'Base' level corresponding to the relevant import evaluated at the world price level. The world price is uniform irrespective of the trading regime, i.e. Daewoo's purchased by households or by firms have the same world price. The relevant formula for the 'Base' price is:

$$ER.WPM_{k,r} XM_{tr,k,r}$$

The 'Tax' price level corresponds to the landed (CIF) price plus the border distortions—normally tariffs. In the XMY00 matrix, the border distortions are an aggregation of three possible distortions—tariffs, subsidies and non-tariff barriers. The split of this aggregate revenue is done in a subsequent matrix. The 'Tax' price level therefore corresponds to the following formula:

$$ER.WPM_{k,r} (1+\sum_{md} \tau^{m}_{tr,k,r,md}) XM_{tr,k,r}$$

Thus the difference between the 'Tax' price and the 'Base' price is the revenue generated by the three border distortions. The 'Marg' level corresponds to the 'Tax' level adjusted by the value added tax on imports.[8] This corresponds to:

$$ER.WPM_{k,r} (1+\sum_{md} \tau^{m}_{tr,k,r,md}) (1+\tau^{v,M}_{tr,k,r}) XM_{tr,k,r}$$

Note that the value added tax on imports is assumed to be specific to the trading regime and the country of origin (as are the other import price distortions). The final level, 'User', corresponds to the price of imports inclusive of the domestic trade margins:

[8] This is somewhat inconsistent with the way margins are interpreted in the case of domestic goods. The reason is that it is assumed that the value added tax on imports is imposed at the point of origin of the imports, for example the ports, and before the goods are transported to the wholesale or retail markets. Thus the margins will be added to the value of the goods including the value added tax.

$$\left[ER.WPM_{k,r}(1+\sum_{md}\tau^{m}_{tr,k,r,md})(1+\tau^{v,M}_{tr,k,r})+\tau^{mg,M}_{tr,k,t}PTMG_{k} \right] XM_{tr,k,r}$$

Consistency requires that the revenues generated by the import distortions line-up with the information in the 'imptx', 'impts' and 'ntbY' rows of the commodities columns in the SAM and that the value added tax revenues from imports add up to the corresponding values in the 'vattx' row of the commodities column. The margins information will have to be consistent with the overall margin values contained in the SAM.

IMPY. The IMPY00 matrix provides the finer detail on the revenues generated by the import distortions. The key dimension is the 'md' dimension that has three components—'TAR', 'SUB' and 'NTB' corresponding to tariffs, subsidies and non-tariff barriers. All three distortions are additive and reflect different components of an overall *ad valorem* price wedge. The respective aggregations across distortions have to line up with the 'imptx', 'impts' and 'ntbY' rows in the commodities columns of the SAM. They also need to be consistent with the wedge information contained in the XMY matrix.

XEY. The XEY00 matrix corresponds to export flows evaluated at three different price levels for each commodity and trading partner. The 'Base' level reflects the producer price:

$$PE_{k,r}XE_{k,r}$$

The 'Marg' level reflects the pre-FOB price, i.e. the producer price plus the cost of shipping the commodity to the border/port:

$$(PE_{k,r}+\tau^{mg,E}_{k,r}PTMG_{k})XE_{k,r}$$

The 'User' price corresponds to the FOB price, inclusive of any export/tax or subsidy. Thus the 'User' price of exports is the world price:

$$(PE_{k,r}+\tau^{mg,E}_{k,r}PTMG_{k})(1+\tau^{e}_{k,r})XE_{k,r}=ER.WPE_{k,r}XE_{k,r}$$

The export tax revenues contained in this data matrix have to line up with the 'exptx' row of the commodities columns of the SAM.

HLDTSY. The next auxiliary data matrix, hldtsY00, contains detailed information on household consumption subsidies.[9] The information is specified by commodity and household. It corresponds to the following expression:[10]

$$\tau^{c}_{k,h}PA_{k,h}XA_{k,h}$$

The aggregation of the subsidies, across commodities, have to line up with the 'hldts' row in the household columns of the SAM.

[9] Note that from the model's perspective this data is treated as tax revenues. Thus subsidies would be entered as negative revenues.

[10] Unlike most of the other distortions, the data in this matrix corresponds to the distortion revenue, not the value of consumption at the subsidized price.

FCTTX. The matrix FCTTX00 contains the revenues generated by factor taxes. It covers all factors over all activities. Like the household consumption subsidies, it corresponds to the tax revenues, not the value of factors net of tax. In the absence of factor taxes, the data matrix will be empty. The values in the matrix correspond to the following formula:

$$\tau_{i,v}^{xfv} NFR_{i,v} FD_{i,v}$$

The variable *NFR* corresponds to the net of tax factor return (for example wages or land rental), *FD* is the relevant factor demand and τ^x is the factor tax rate. The aggregate factor taxes (by activity) need to line-up with the 'fcttx' row in the Activities column.

FCTTS. The final SAM-related auxiliary data matrix is the matrix of factor subsidies, fctts00. It covers all factors over all activities. Like the household consumption subsidies, it corresponds to the subsidy costs (and will typically be negative). In the absence of factor subsidies, the data matrix will be empty. The values in the matrix correspond to the following formula:

$$\tau_{i,v}^{sfv} NFR_{i,v} FD_{i,v}$$

The variable *NFR* corresponds to the net of tax factor return (for example wages or land rental), *FD* is the relevant factor demand and τ^s is the factor tax rate. The aggregate factor subsidies (by activity) need to line-up with the 'fctts' row in the Activities column.

TK. TK00 is a scalar that represents the volume of the national capital stock. It needs to be in units consistent with the level of investment and with the operating surplus in the SAM. For example, assume the operating surplus is 400, GDP is 1000 and investment is 300. If the capital stock is 4000, this implies the average rate of return on capital is 10 percent (400/4000), and the growth in the capital stock is 2.5 percent, assuming a rate of depreciation of 5 percent ([0.95*4000+300]/4000-1).

TPOP. TPOP00 is a scalar with the national population level. It can be in any convenient units. The population and shares (see below) are not essential for the model but are used on output.

POPSH. POPSH00 is a vector with the population shares by household. The values are decimal shares (for example 0.2 instead of 20).

ERD. ERD0 is the base year nominal market exchange rate in units of local currency per U.S. dollar. It is not used by the model but can be used for output. To calculate per capita income in dollar terms, take the value of GDP from the model, divide this by the exchange rate to convert GDP into dollars, and then divide by the population.

PPP. PPP0 is the PPP exchange rate. It converts values from dollars into purchasing power parity dollars. Say the model has per capita income at 2000 LCU. If the dollar exchange rate is 2, then the per capita income in dollars is 1000. If the PPP exchange rate is 3, per capita income is 3000 in purchasing power parity terms.

ETA. ETA00 is a matrix of income elasticities. It has the same dimension as the original SAM and is commodity and household specific. The aggregation facility will take care of aggregating the income elasticities using consumption shares.

LABVOL. Labvol00 contains the matrix of labor demand in units of workers. If information is lacking, the matrix would normally contain the labor remuneration matrix from the SAM itself. In this case, all wages will be initialized at 1.

The aggregation and parameter file

Each application of a country-specific model is associated with a specific configuration of the base SAM, the key model elasticities and model closure. Simulations therefore involve preparing the aggregation and parameter file and then a file describing the nature of the experiment.

The aggregation file is essentially separated into three parts. The first part defines the dimensions of the resulting model. It is virtually identical with the initial part of the SAM file itself, except that the dimensions can be different. The user can aggregate the original SAM in several dimensions—Armington accounts (including activities and households), commodities, factors of production, enterprises and trading partners. The most aggregate model can have a single activity, one household, an account each for labor and capital and a single trading partner. The maximum of course is simply a replication of the original SAM dimensions.

The second part of the file contains the so-called aggregation mapping, i.e. the correspondence between the aggregated accounts and the original SAM dimensions. The third part of the file contains the key model parameters. Users are free to add additional information to this file that may be useful for a specific application—for example the definitions of subsets.

SAM accounts

Table 4 provides a description of the functional SAM accounts. It is almost identical to the SAM accounts of the base SAM, so only significant differences will be highlighted below.

Some of the accounts in the SAM are constant—for example 'Margn', the fiscal accounts and the 'BoP' account. All other dimensions are under user control though there are some restrictions due to the definition of subsets.

The value added accounts have more subsets than the original SAM. First, the labor subset is split into two subsets—'ul' and 'sl'. The first subset refers to labor that is substitutable with capital, typically, but not limited to, unskilled labor. The second, its complement, represent labor that is a complement with capital, oft-times skilled labor. Refer to the model documentation for further information. The aggregation file also contains subsets for the capital account(s), the land account(s) and the sector-specific resource. Even if the national data has a single capital or land account, it is possible to break these out to reproduce market segmentation for resources.

Table 4: SAM set and subset definitions

SAM dimensions	
s	Full list of SAM accounts—consistent with Table 1 above
ss	An alias for s
a	The subset of Armington agents, a subset of s
i	The subset of activities, a subset of a
j	An alias for i
k	Commodities, a subset of s
kk	An alias for k
v	Value added accounts, a subset of s
l	Labor accounts, a subset of v
ul	'Unskilled' workers, a subset of l
sl	'Skilled' workers, a subset of l and a complement of ul
lt	Land types, a subset of v
kt	Capital types, a subset of v
nr	Sector-specific factors (e.g. natural resources), a subset of v
e	Enterprise accounts, a subset of s
fd	Domestic final demand accounts, a subset of a
h	Household accounts, a subset of fd
f	Other domestic final demand accounts, a subset of fd
r	Trading partners, a subset of s
Auxiliary dimensions	
tr	The set of trading regimes—user determined

Aggregation mappings

Each aggregation dimension is associated with a set mapping. A set mapping maps each aggregate account to one or more of the original accounts. Table 5 provides a list of the standard aggregation mappings for the model. There is one special mapping, described below, that maps Armington agents to the individual trade regimes.

In most typical CGE applications, there is a single trading regime and all agents face the same tariff structure. The model allows for special trading regimes. Each Armington agent is assigned to a specific trading regime. Import demand across agents in a specific regime is aggregated. Aggregate import demand is then allocated across trading regimes using the tariffs specific to that trading regime. The 'mapa' aggregation provides the mapping from Armington agents (according to the dimensions of the aggregate SAM) to the specific trading regimes. For the classical model, the dimension of the set 'tr' is simply 1, i.e. all agents face the same tariff rates.

It is possible to eliminate accounts via aggregation. For example, if the initial SAM includes land and/or natural resource accounts, they can be aggregated away by mapping these accounts to the capital account.

Table 5: Aggregation mappings

Name	Description
mapa(a,tr)	Mapping of the Armington agents to the trade regimes
mapArm(a0,a)	Aggregation mapping of Armington agents
mapk(k0,k)	Aggregation mapping of commodities
mapv(v0,v)	Aggregation mapping of factors
mape(e0,e)	Aggregation mapping of enterprises
mapr(r0,r)	Aggregation mapping of trading partners
maptr(tr0,tr)	Aggregation mapping of trade regimes

Model parameters

The section of the model parameters starts with two parameters that are not really part of the model, but are used to scale the model numerically. The GAMS solvers are sensitive to the overall scale of the variables of the model. A rule of thumb is to have the scale of GDP between 10 and 100. If the initial SAM implies a GDP of say one million, using a scale factor of a thousand (10^3) or million (10^6) would be appropriate. All output will be re-scaled to match the scale of the input SAM. The scale factor has the name 'scale' and should be entered as a multiplicative factor. Hence, to divide the initial SAM by 1000, use the scale factor 0.001 (or 1e-3). Similarly, population and labor volumes should be scaled to some value between 1 and 100 in aggregate. The population scale factor is named 'pscale'.

The key model parameters are divided into two tables—a table with the 'activity' based parameters, and a table with the 'commodity' based parameters.

The activity-based key parameters are contained in the table 'KeySect'. Table 6 describes the list of key activity-based parameters. CET transformation elasticities are allowed to take the value of 'inf', in which case the law of one price holds.

Table 6: Key activity-based parameters

Name	Description	Variable
omegas	CET elasticity in multi-production output function	ω^s
sigmap	CES elasticity between aggregate intermediate demand and aggregate value added	σ^p
sigman	CES elasticity across intermediate demand	σ^n
sigmav	CES elasticity between land and the other factors of production.	σ^v
sigmakl	CES elasticity between 'unskilled' labor and other factors of production	σ^{kl}
sigmaks	CES elasticity between 'skilled' labor and capital.	σ^{ks}
sigmau	CES elasticity across 'unskilled' labor	σ^u
sigmas	CES elasticity across 'skilled' labor	σ^s
sigmak	CES elasticity across different types of capital	σ^k
sigmat	CES elasticity across different types of land	σ^t

The commodity-based key parameters are contained in the table 'KeyComm'. Table 7 describes the list of commodity-based parameters. The export demand elasticity ('etae') is allowed to take the value of 'inf', in which case the world price is fixed (and the demand curve is horizontal). (N.B. The input of the trade elasticities needs to be reformulated since some of them have multiple dimensions, and not simply the commodity dimension. The top-level Armington elasticity is specific to Armington agents, for example, households' preferences for cars is allowed to differ from industries' preferences for cars. The second-level Armington elasticity is specific to each trading regime. And the export demand elasticity is specific to each region of destination.)

Table 7: Key commodity-based parameters

Name	Description	Variable
sigmac	CES elasticity for aggregating produced commodities	σ^c
sigmam	Armington CES elasticity between domestic goods and aggregate imports	σ^m
sigmaw	Armington CES elasticity across imports by trading partner	σ^w
sigmax	CET elasticity between domestic supply and aggregate export supply	σ^x
sigmaz	CET elasticity across foreign export markets	σ^z
etae	Export demand elasticity	η^e

The remaining key parameters and data are specific to particular blocks of the model.

Frisch. The Frisch parameter is used to calibrate the LES consumer demand system combined with the income elasticities (that are included with the base SAM data file).

Other final demand expenditure elasticity. The other final demand functions (i.e. aside from households) use a generic CES expenditure function. The expenditure elasticity is given by sigmaf and is specific to each other final demand account.

Capital market. Capital supply is governed by a nested CET structure. At the top level, aggregate capital is allocated to different types of capital (for example rural and urban). At the second level, capital is allocated across sectors according to their relevant rate of return. The top level CET elasticity is given by 'omegakt'. A value of 0 implies that the capital stock, by type of capital, is fixed. An elasticity of infinity ('inf') assumes perfect capital mobility across types. (N.B. If the elasticity is negative, the model will assume perfect segmentation of the capital stock by type of capital. The aggregate supply of capital, by type, will be governed by a constant elasticity supply function. The supply elasticity is given by 'epskt'. The latter is ignored if the top-level CET elasticity is zero or greater.) The allocation of capital across sectors, by type of capital, will be governed by the CET elasticity omegak. The two extreme values are zero, corresponding to sector-specific capital, and infinity, corresponding to perfect capital mobility.

Land market. The land market, like the capital market, is governed by a nested CET structure. The top-level CET, governed by the CET elasticity 'omegatl', allocates aggregate across types. The second level, governed by the CET elasticity 'omegat'. Similar to capital, the extreme values are given by 0 and 'inf'.

Sector-specific resource. Supply of the sector-specific resource is governed by a constant elasticity supply function. The supply elasticity is given by 'omegar'. A value of 0 implies fixed supply. An elasticity of infinity implies a horizontal supply curve and the real price of the sector-specific resource is fixed in this case.

Labor market. The labor market is driven by two facets—first, the possibility of a segmented labor market (rural versus urban), and within each segment the potential for minimum wages and unemployment.

Labor market segmentation requires at least two inputs from the user. First, the user must specify the segmentation mapping, i.e. the division of activities into the two segments. There is a set, gz, that defines three zones—rural, urban and total. (The user cannot change the set definition, only the mapping, since the set elements are hard coded in the model.) The user then provides the mapping from activities to the rural and urban zone using the mapping set named 'mapg'. The 'total' segment should contain all activities. The other required input is the rural to urban migration elasticity, 'omegam'. If the migration elasticity is set to infinity, the model will automatically assume that there is a single integrated labor market. (N.B. The labor parameters are all specific to each type of labor. Hence it is possible to have segmented labor markets for one type of labor, for example 'unskilled' and an integrated labor market for other types of labor, for example 'skilled' labor). If 'omegam' is finite, labor markets will be segmented between rural and urban. The segmentation could be perfect if the initial level of migration is 0. The initial level of migration is given by 'migr00'. It should have the same units as labor volume. If the migration elasticity is 0, then migration will be fixed in levels. Labor force growth is given by the parameter 'glab0'.

The remaining labor market parameters govern the functioning of labor markets on the individual labor market segments. There are two possibilities.

Initial full employment. Set the level of initial unemployment 'ue0' to the minimum level of unemployment 'uemin0'. The minimum wage will be set to some level of the actual equilibrium wage. If the parameter 'wmin0' is equal to 1, then the labor market is in equilibrium exactly at the

minimum wage. If 'wmin0' is equal to 0.95, then the equilibrium wage could decline by 5 percent before hitting the minimum wage and creating unemployment. If a labor segment has no effective minimum wage, it is best to set 'wmin0' to a very low number, such as 0.001. Then the labor market will clear under virtually any circumstance.

Initial unemployment. The user needs to provide an initial level of unemployment (in decimal points) above the minimum level of unemployment 'uemin0'. The minimum wage in the parameter file will be ignored. It will be initialized to the actual wage being paid.

Minimum wage function. The minimum wage function is driven by two variables—the aggregate price level and the level of unemployment. The elasticity 'omegap' defines the elasticity of the minimum wage with respect to the price level. A value of 1 results in full indexation. A value of 0 means that the minimum wage is fixed in nominal terms. The elasticity 'omegaue' governs the relation between the minimum wage and 1 minus the unemployment rate. In other words, it should be positive since the minimum wage would be expected to decline as unemployment rises.

The final section of the aggregation and parameter file includes some additional sets useful in the output routines of the model, see table 8. The sets provide additional aggregations for activities, commodities, labor, capital and trading partners. In particular, the user will most likely desire an aggregation for the totals, but may also desire functional aggregations (for example agriculture, textile and clothing, energy, services, etc.). For each aggregation, the user provides two sets. The first defines the dimension of the aggregation. The second defines the mapping from the model dimensions to the output dimensions. At a minimum, there would normally be a one-to-one mapping between the model dimensions and the output dimensions. The additional dimensions are up to the user.

Table 8: Aggregation mappings for output routines

Name	Description
Set ia	Dimensions of output activities
mapia(ia,i)	Mapping of model activities to output activities
Set ka	Dimensions of output commodities
mapka(ka,k)	Mapping of model commodities to output commodities
Set ra	Dimensions of output regions
mapra(ra,r)	Aggregation mapping of model regions to output regions
Set la	Dimensions of output labor accounts
mapla(la,l)	Aggregation mapping of model labor accounts to output labor accounts
Set kta	Dimensions of output capital accounts
mapkta(kta,kt)	Aggregation mapping of model capital accounts to output capital accounts

This ends the description of the data files for the model. In summary, the base SAM data file is normally created only on a very periodic basis—say every two to three years depending on the cycle of updating of the detailed national accounts. There may also be modifications made to augment the initial data—for example for additional household or trade detail. The main data file the user prepares on a more regular basis is the aggregation and parameter file. It is normally

tailored for each new project—with potentially a different level of aggregation and/or changes to the main parameters.

3

Model files

There are only three model files:

Model.gms	Contains base model code—including declarations, initializations, calibration and model equations
Postsim.gms	Contains code to save model results to an external file—typically used for post-processing in Excel.
Sam.gms	Contains code to save simulation SAMs to an external file—intended for use in Excel.

The most important one, Model.gms, contains all of the model code. It is divided into three sections. The first part reads in the input data, declares the initial model variables, initializes the initial model variables, and calibrates the model's parameters. All model variables have an associated initial variable. For example, the model variable `xp`, sectoral output, has a corresponding initial variable labeled `xp0`.[11] The initial variables are useful when calculating changes in model variables relative to the initial baseline. The second part of the 'Model' file declares the model variables and initializes them to their initial value as they are generated by the initialization and calibration part of the module. For example, sectoral output will be initialized by the following statement:

```
xp.l(i) = xp0(i)
```

The initialization will also set all exogenous variables subject to the default closure rules. These are listed below:

```
* Default fiscal closure
Rsg.fx = rsg0 ;
dirtxadj.fx     = dirtxadj0 ;
dirtxhadj.l     = dirtxhadj0 ;
vattxadj.fx     = vattxadj0 ;
vatdtxadj.fx    = vatdtxadj0 ;
vatmtxadj.fx    = vatmtxadj0 ;

* Default savings/investment closure
xf.l(f)         = xf0(f) ;
savadj.fx       = savadj0 ;
savhadj.fx      = savhadj0 ;
```

[11] In terms of GAMS, the initial variables, i.e. those with a '0' suffix are declared as parameters since they do enter the model specification itself.

```
* Default balance of payments closure and choice of numéraire
Savf.fx = savf0 ;
er.fx   = er0 ;
cpit.l  = cpit0 ;
```

The third part of the model file declares and defines the model equations. The final statement defines the model itself, i.e. the list of equations to be included in the model.

The other two files are used only for reporting output. Most of the output will be in the form of a text-based (ASCII) file using the common CSV (comma separated values) format. These files are most commonly loaded into Excel for post-processing analysis. The file 'postsim.gms' reports most of the model's variable values. Each result will appear on an individual line with additional information. The additional information at a minimum includes the code of the scenario[12], the 'year' of the simulation[13] and the variable name. Multi-dimensional data will also include the additional dimension(s), for example sector, skill level, trading partner, etc.[14] Section 5 will describe how to use these files for post-processing.

The final model file, 'sam.gms' outputs the SAM for each model run and year, also as a CSV file.

[12] A brief mnemonic identifying individual simulations.

[13] In comparative static simulations, the 'year' is most often a mnemonic identifying the individual simulation shock. In dynamic simulations, it would represent actual calendar years.

[14] Most of the output is also segmented by type of output, for example macro variables have the type 'M1' or 'M2', and sectoral data 'S1' and 'S2'. In practice, this segmentation has not been widely used and may be dropped in subsequent versions.

4

Simulation files

The core of the work for most model users will be preparing the aggregation file (see section 3) and the simulation files. Simulations can be run in either comparative static mode or in dynamic mode. In either case, the simulation files are virtually identical, though there are some differences. The simulation files have a similar structure—the time dimension is defined, input and output filenames are declared, simulation shocks are defined and the model is solved iterating over the defined time horizon with output saved after each model solution.

Comparative static simulation files

Figure 1, is annotated GAMS code with the minimum requirements for a comparative static simulation. The key part is defining the 'time' horizon of the model. For comparative static experiments, the 'time' horizon means the number of different simulations (or shocks) that will be conducted. The other initialization information determines the file names of the input and output files. The simulation file also needs to pull in the model code—this is done through GAMS' '$include' statement. Finally, the program will loop over all 'time' periods and optionally solve the model. Typically, the first 'year' will not be solved for since it represents the initialized database.

Diagnostic testing

It is critical in CGE modeling to undertake detailed diagnostic testing when a new model is implemented. A new model can mean a new initial dataset, modifications or additions to model specification, modifications to parameters, and/or changes to model closure. At a minimum, diagnostic checking requires three phases and two simulations. The first simulation is the so-called residual check. The main purpose of the residual check is to see if the model can re-produce the base data in the absence of any shock. The secondary purpose is to see if the model initialization and calibration is consistent. The second simulation involves testing the price homogeneity of the model. This is done by changing the value of the model numéraire. Say it is raised by 50 percent. In this case, all nominal prices should increase by the exact same amount, i.e. 50 percent, all value variables should likewise increase by 50 percent, for example tax revenues, and all volume variables should be identical to their base year levels. Figure 2 provides a sample simulation file for undertaking the simple diagnostic exercise.

Figure 1: A minimal comparative static file

```
* 1. Set the simulation code
Set simcode / test / ;

* 2. Define the 'time' dimension for the simulation.
*    The 'Base' year is not solved.
*    The 'Check' year will be used to see if the model
*    is able to reproduce the base.
Set t time / Base, Check / ;

* 3. For comparative static simulations, CalFlag should
*    always be set to 0.
Scalar Calflag / 0 / ;

* 4. The header parameter determines whether the first
*    line of output should include a header line
*    describing the output fields. Set to 1 to output the
*    header line, else set to 0.
Scalar Header / 1 / ;

* 5. Model results will be saved to a CSV file defined
*    by the 'report' filename.
File report / comp.csv / ;

* 6. Simulation SAMs will be saved to a CSV file defined
*    by the 'samf' filename.
File samf / 'compsam.csv' / ;

* 7. The user must prepare the main input file, i.e. the
*    file containing the aggregation codes and key model
*    parameters. It will automatically pull in the base
*    SAM parameter file.
$include 'agg.dat'

* 8. The next include statement pulls in the model code.
$include 'model.gms'

* 9. The final part loops over all time periods. Within
*    the loop, there would normally be at least one
*    'solve' statement, though this is not strictly
*    necessary. Oft-times, the first 'year' will not
*    be solved for since this represents the base year
*    data, initialization and calibration. Normally, after
*    each solve statement will be the inclusion of the
*    'postsim' file, which saves the model results to
*    external files.

Loop(t,
   If (ord(t) gt 1, solve cge using mcp ; ) ;
$include 'postsim.gms'
) ;
```

The ability to reproduce the base data set is not sufficient to see if the model has been appropriately calibrated and initialized. The latter check requires that each model equation is satisfied exactly (or at least within the degree of the base data's accuracy). GAMS provides a

useful diagnostic tool for this. One of its options is to print out in the list file for each equation (or a subset of each block of equations) the difference between the left-hand and right-hand side of an equation. If the equation is satisfied using the initial volumes and calibrated parameters then the residual will be zero (or a very small number).

Figure 2: Model diagnostics

```
Set simcode / test / ;

Set t time / Base    Base year solution
             Check   Residual check and model verification
             Homog   Homogeneity test
/ ;

Scalar Calflag / 0 / ;

Scalar Header / 1 / ;

File report / comp.csv / ;

File samf / 'compsam.csv' / ;

$include 'agg.dat'

Options limrow=3, limcol=3 ;

Loop(t,

*  Test for model homogeneity. Assumes the exchange rate
*  is the numéraire.

   If (ord(t) eq 3, er.fx = 1.5*er0 ; ) ;
   If (ord(t) gt 1, solve cge using mcp ; ) ;

$include 'postsim.gms'

) ;
```

To turn on this option, the GAMS 'options' statement must include the 'limrow' parameter, see Figure 2. If 'limrow' is set to 3 it will print at most three equations for each block of equations. When the diagnostic simulation is over, open the list file and search for the characters 'LHS'. There are three possible situations. If the model is well initialized and calibrated, the value of LHS will be 0 or a small number. If the equation has an isolated exogenous variable, such as the import price equation, the value of 'LHS' will be equal to the exogenous variable (if the equation is satisfied). The third situation is when the equation is not satisfied in which case LHS will differ from zero (or differ from the exogenous right hand side variable). GAMS will identify these equations as infeasible. A quick way to search for infeasible equations is to look for '***', though this will also pick up equations where the infeasibilities are small numbers. (N.B. If this option is on for the homogeneity test, only the equations where the numéraire appears directly will be infeasible.)

When running the price homogeneity test, it is important to simulate the model by either removing potential non-homogeneities in the model, or by dealing with them exogenously. For example, the minimum wage equation has the potential for introducing non-homogeneities if the elasticity of the minimum wage with respect to prices is not equal to 1. To test the homogeneity, one could temporarily set this elasticity to unity.

Policy simulations

There is almost an infinite variety of potential policy simulations. The best way to simulate new policy shocks is to start with pre-existing policy simulation files of which several are shipped with the prototype model. Almost all policy simulations involve changing one or more policy instruments and perhaps changing the default closure rules. Figure 3 shows a typical simulation demonstrating how to possibly model a free trade arrangement with a set of trading partners for a subset of commodities.

The shock will be run three times—i.e. the output will contain four sets of results, the base plus the results from the three policy simulations. The shock is identical in all three policy simulations and the purpose is simply to assess the impacts of different closure rules. The policy shock is the elimination of tariffs between the home country and a subset of trading partners. This subset is labeled 'fta' and is pre-defined, for example in the parameters file. The reform only covers non-agricultural goods and this latter subset is also pre-defined and labeled 'nag'. All agents benefit from the FTA, and thus the shock covers all possible trading regimes, *tr*. Finally, only tariffs are removed. If there are non-tariff barriers, 'NTB', or import subsidies, 'SUB', these are maintained at their base year values.

The closure rule is governed by a vector of flags, FiscClos, that can take one of three values[15]– with 0 corresponding to the default, 1 replacing the direct tax instrument with the value added tax, and 2 corresponding to holding all tax instruments constant and allowing the fiscal deficit (or surplus) to adjust. This vector is initialized for each of the four 'time' periods.

When running multiple policy simulations within one simulation file, it is good practice to reset the model defaults at the beginning of each solution iteration. It is also good practice to reset all of the policy instruments to their initial values, though in the case of this particular simulation, this is not done since a unique policy shock is being undertaken in all three policy simulations. The fiscal closure will be set according to the value of the flag vector FiscClos for each of the iterations. The code does not test for the value '0', since this is the default closure, which is reset automatically.

[15] There is a wider choice of fiscal closure rules so this vector of flags could easily take on more values.

Figure 3: Example of simulating a free trade area

```
Set simcode / fta / ;

Set t time / Base    Base year solution
             FTA_def FTA with the default closure
             FTA_vat FTA using the VAT instrument
             FTA_rsg FTA running a deficit

/ ;

Scalar Calflag / 0 / ;

Scalar Header / 1 / ;

File report / fta.csv / ;

File samf / 'ftasam.csv' / ;

$include 'agg.dat'

Options limrow=0, limcol=0 ;

* Setup the shock

* Define the closure rule options
*
*   0 - Default closure (household direct taxes)
*   1 - Adjust the VAT tax rate (both domestic and import)
*   2 - Allow the government to increase the deficit

Parameter FiscClos(t) /
   Base    0
   FTA_Def 0
   FTA_VAT 1
   FTA_Rsg 2
/ ;

Loop(t,

*  Reset the default closure

   rsg.fx = rsg0 ;
   dirtxhadj.lo = -inf ; dirtxhadj.up = +inf ;
   vattxadj.fx  = vattxadj0 ;

*  Define the policy shock

   if (ord(t) eq 2,
      tm.fx(tr, nag, fta, "TAR") = 0 ;
   ) ;

*  Set the fiscal closure

   If (FiscClos(t) eq 1,
      dirtxhadj.fx = dirtxhadj0 ;
      vattxadj.lo = -inf ; vattxadj.up = +inf ;
   elseif (FiscClos(t) eq 2),
      dirtxhadj.fx = dirtxhadj0 ;
      rsg.lo = -inf ; rsg.up = +inf ;
```

```
   ) ;
   If (ord(t) gt 1, solve cge using mcp ; ) ;

$include 'postsim.gms'

) ;
```

5

Running simulations and processing results

This section explains how to run the simulations using GAMS and to post-process the results in Excel.

Running simulations

The description herein explains how to simulate the CGE model in a DOS-like environment. The main alternative is to use the GAMS IDE that provides a more user-friendly interface but in the end has the same functionality as the DOS-like environment.[16] All the examples described in this section assume that the model files are contained in the directory 'v:\CGEProto\Mongolia'.

The first step is to open a DOS-box in windows. This could be done by double-clicking on the 'cmd' icon on the desktop if it exists, or by issuing the command 'cmd' from the 'Run…' task in the 'Start' task bar. The DOS-box will open to a default directory. Issue the following commands to get to the desired directory:[17]

```
v:
cd v:\CGEProto\Mongolia
```

To run a simulation file, for example ieq15ct.gms , simply issue the following command:[18]

```
gams ieq15ct -pw=115 -charset=1
```

There are two parameters on this command line, one is optional, the other may be required. The optional parameter is 'pw' that sets the page width of the listing file, in the case above 115 characters per line. The value for this option will depend on screen resolution (and the editor being used to view the list files). Some screens may allow viewing longer lines in the list file in which case a higher value for the page width can be used. The second option, charset, allows for the use of the international character set—including accented vowels, etc. If the input files contain any international characters, this option is required.

(Hint. This hint section describes alternative ways to organize the files and directory structure. One alternative is to keep the three model files in a separate directory. This allows to keep a

[16] There are alternative IDE's to GAMS, for example OxEdit. One reason to continue to use the DOS environment despite some of the amenities available in the IDE's is the wide availability of powerful editors. The GAMS editor has some nice features, but is not very powerful in other respects.

[17] In the GAMS IDE, use the 'Open' task to open a simulation file in the v:\CGEProto\Mongolia directory.

[18] In the GAMS IDE, simply issue the run command if the simulation file is open.

single copy of the core model, but to have multiple directories holding only the simulation files. This avoids the problems of synchronizing different versions of the core model.[19] Since the core files would be in a different directory it is necessary to let GAMS know where the core models can be found. One alternative is to hard-code the full path name of the core files in the simulation files (e.g. $include 'v:\CGEProto\Generic\model.gms'). This is not ideal if sharing files across machines with a different folder structure or if for some applications the core files are in one location and in other in a different location. An alternative solution is to provide the directory link on the command line, for example:

```
gams ieq15ct -pw=115 -charset=1 -idir=v:\CGEProto\Generic
```

The 'idir' command line option tells GAMS to search the optional directory for include files if it can locate them in the existing default directory.

The proliferation of directories and cross-machine work has sometimes made working with standard structures tedious. One useful option is to create a virtual machine that always maps to a fixed directory structure. This can be done in DOS (and Windows) using the 'subst' command. For example, the CGEProto directory may be located in different directories on different machines (for example if it is being shared in a work environment). One way to make sure that the files appear to be in the same structure regardless of the environment is to use the 'subst' command, for example:

```
if not exist v:\ subst v: "C:\Documents and Settings\wb1234\My Documents"
```

This will create a virtual drive labeled 'v:' that is mapped to the 'My Documents' directory on a specific machine. The 'subst' command can be customized for each machine and/or user. After which, all users on all machines can use the exact same directory structure. This feature is particularly helpful for the pivot tables used in the Excel files described below. This is because the Excel files contain the full path name of the underlying external data source and this can create problems if the data file isn't where Excel expects it to be.)

On most machines, the comparative static simulations should execute fairly rapidly and most of the screen output will scroll by rapidly. When the simulation is done, it is good practice to open the listing file to quickly assess whether the simulations solved successfully. If doing diagnostic checks, the listing file should also be analyzed for the residual check.

Apart from the output of diagnostic results, including Walras' Law, the listing file rarely includes any significant model output. Most relevant output will be saved in the two CSV files—one containing the core model results and the other containing the simulations SAMs. Both of these files are structured to be read into Excel (and any program that can read database style CSV files). The next section describes post-processing in Excel in more detail.

Processing results

Each simulation file will result in two CSV output files—the core model results and the simulation SAMs. It is expected that these files will be loaded into Excel using Excel's pivot

[19] It can raise problems when re-running old simulations since the simulation and data files may be inconsistent with changes to the core model.

table feature (see Annex A). Once the data is loaded into an Excel pivot table, there are two main options for creating standardized reports.

The first option is to create customized pivot tables for each of the basic report tables. This can be done in separate worksheets. Each customized pivot table can be built from the original source pivot table (and assuming minimal, if any, filtering of the input data). If the various pivot tables are linked to a single original pivot table, all pivot tables can automatically be updated by using the refresh function from within any of the linked pivot tables. Simply place the cursor in any pivot table. Under the Data menu, click on the Refresh Data menu item. Alternatively, right-click from anywhere in a pivot table and choose the Refresh Data menu item. Thus one worksheet could contain basic macro data, another sectoral output results, and another sectoral demand data, etc.

The second option is to create a master pivot table in a data worksheet that has all of the data the user wants to create the various tables for reports. Customized tables can then be designed using Excel's GetPivotData function. The function is a bit cumbersome, but it has a great advantage in that it doesn't rely on any particular structure of the data file, i.e. it is independent of where the actual data is stored (for example in a specific cell and row). The pivot table can be modified multiple times, but the GetPivotData function will always locate the right data (assuming it exists). The function has only two arguments. The first indicates from which pivot table the value should be extracted. The second provides the relevant field names and values of the fields to be extracted. For example, the following formula will extract the value of 'GDP' for the 'Base' year from the pivot table stored in worksheet Sheet1:

```
Getpivotdata(Sheet1$A$6,"Variable[GDP] Year[Base]")
```

The function is typically more complex than above, because most pivot tables have more than two fields. Creating standardized tables would be tedious if the fields have to be identified individually in each GetPivotData function call. The field descriptors can be constructed using the Concatenate function. For example, if one wants to construct a table of sectoral output for a number of different simulations, the table could be constructed by putting the sectoral labels down a column (for example starting in A4), and the different 'years', i.e. simulations across a top row, say starting in C1. The table can be started by typing the following formula in cell C4:

```
Getpivotdata(Sheet1$A$6,Concatenate("Variable[XP] Sector[",$A4,"Year[",C$1,"]"))
```

This formula can be replicated to fill the entire data matrix by copying the formula and pasting it into all of the rows covered by the sectors and all of the columns covered by the 'years'.

Since the output from GAMS typically uses cryptic mnemonics for sectoral and structural fields, tables need to be augmented with descriptive labels. A good practice is to have a master list of labels—both the GAMS mnemonics and the descriptions. These should be contained in a two- or three column matrix somewhere in the workbook. It is also convenient to give this matrix of labels a defined range name by using the Insert/Name/Define feature of Excel. Say the range name for the labels is called 'Labels'. The vlookup function can then be used to insert descriptive names for the GAMS labels. Using the same example as above, the following formula can be entered in cell B4:

```
Vlookup($A4,Labels,2,false)
```

This instructs Excel to lookup the value contained in cell A4 in the list of labels called ‘Labels’ and to retrieve the second column of that list. The value ‘false’ indicates that the list is not sorted in which case it will read the entire list until it gets a match.

References

Brooke, Anthony, David Kendrick, Alexander Meeraus and Ramesh Raman (1998), *GAMS: A User's Guide,* GAMS Development Corporation, Washington, DC. http://www.gams.com/docs/gams/GAMSUsersGuide.pdf.

Pyatt, Graham and Jeffrey Round, editors (1985), *Social Accounting Matrices: A Basis for Planning,* The World Bank, Washington, DC.

Reinert, Kenneth A. and David W. Roland-Holst (1997), "Social Accounting Matrices," in Joseph F. Francois and Kenneth A. Reinert, editors, *Applied Methods for Trade Policy Analysis: A Handbook,* Cambridge University Press: Cambridge, UK.

Robinson, Sherman, Andrea Cattaneo and Moataz El-Said (1998), "Estimating a Social Accounting Matrix Using Cross Entropy Methods," *TMD Discussion Paper*, No. 33, Trade and Macroeconomics Division, International Food Policy Research Institute (IFPRI), October, Washington, DC. http://www.ifpri.org/divs/tmd/dp/papers/tmdp33.pdf.

Van der Mensbrugghe (2005), "Prototype Model for a Single Country Computable General Equilibrium Model", *mimeo*, February, The World Bank.

Annex A: Pivot tables in Excel®

Saving GAMS Results in Database Format

The GAMS "put" facility enables virtually unlimited possibilities for saving GAMS results to text (i.e. ASCII) files. Later versions of Excel (and perhaps other spreadsheet programs) include a very powerful feature for structuring database type data in tabular format—these are known as pivot tables. Pivot tables are particularly useful for analyzing data of greater than two dimensions. Examples of these abound in GAMS programs. For example, macro data might have three dimensions if it is defined by variable name and has an index for time and scenario. Sectoral data may have three dimensions or more. Sectoral data will be defined by variable name, and indexed by sector, time, and scenario. Multi-regional models add an extra dimension. The best way to transfer this from GAMS is in database format. A database in text format has one record (i.e. line) per data item. The first line in a data base file contains the name of the fields of the record. For example, to save the macro results from a dynamic scenario would require GAMS code such as:

```
* Define the output file
file report / 'BaU.csv' / ;
put report ;

* Define the reporting years
set tr(t) reporting years ;
tr(t) = yes ;

*----- Write the header
put 'Scenario,year,variable,sector,qualifier,type,value' / ;
report.pc = 5 ;

*----- Loop over the reporting years
loop(tr,
*----- Output the macro data
   put simcode.tl, tr.tl, 'gdpmp',    '', '', 'M1', (rscale*gdpmp.l) / ;
   put simcode.tl, tr.tl, 'rgdpmp',   '', '', 'M1', (rscale*rgdpmp.l) / ;
   put simcode.tl, tr.tl, 'pgdpmp',   '', '', 'M2', (pgdpmp.l) / ;
 ) ;
```

After opening the file, the first put statement writes the data base fields. In the example above, there are seven fields: the simulation title, the year, the name of the output variable, and the corresponding sector, qualifier, type, and value. The output routine loops over the number of reporting years. In this case, it is all years of the simulation, but the subset *tr* can be defined to be less than the full number of years. (N.B. This structure can also be used for a sequence of comparative static experiments where the index time is replaced by simple ordinal indices rather than by defining calendar years.) Macro variables have no sector definitions, so in the output of each record, the sector definition is blank, and the consecutive commas indicate this. The reason why the sector field may be included is that often the output will mix both sectoral data and macro data. These could be separated into different files, but it is also possible to filter the data when being read into Excel, which makes this unnecessary (see below). The qualifier field could

be needed for data that have additional dimensions—for example labor by skill and sector, or imports by sector and region of origin. The 'type' field has been used to segment output into different blocks. (N.B. This feature has been little used and may be dropped in subsequent versions.) The value part of the record may include further processing. For example, most of the value and volume output data will be scaled back to the original scale of the input data. GAMS requires that formulas in a 'put' statement be surrounded by parentheses. Sectoral data can be saved into the same file by GAMS code such as:

```
*   Output the sectoral data

loop(k, put simcode.tl, t.tl, 'xet', k.tl, '', 'S1', (rscale*xet.l(k)) / ; ) ;
loop(k, put simcode.tl, t.tl, 'p',   k.tl, '', 'S2', (p.l(k))   / ; ) ;
loop(k, put simcode.tl, t.tl, 'pd',  k.tl, '', 'S2', (pd.l(k))  / ; ) ;
loop(k, put simcode.tl, t.tl, 'pet', k.tl, '', 'S2', (pet.l(k)) / ; ) ;
loop(i, put simcode.tl, t.tl, 'tp',  i.tl, '', 'S2', (tp.l(i))  / ; ) ;
```

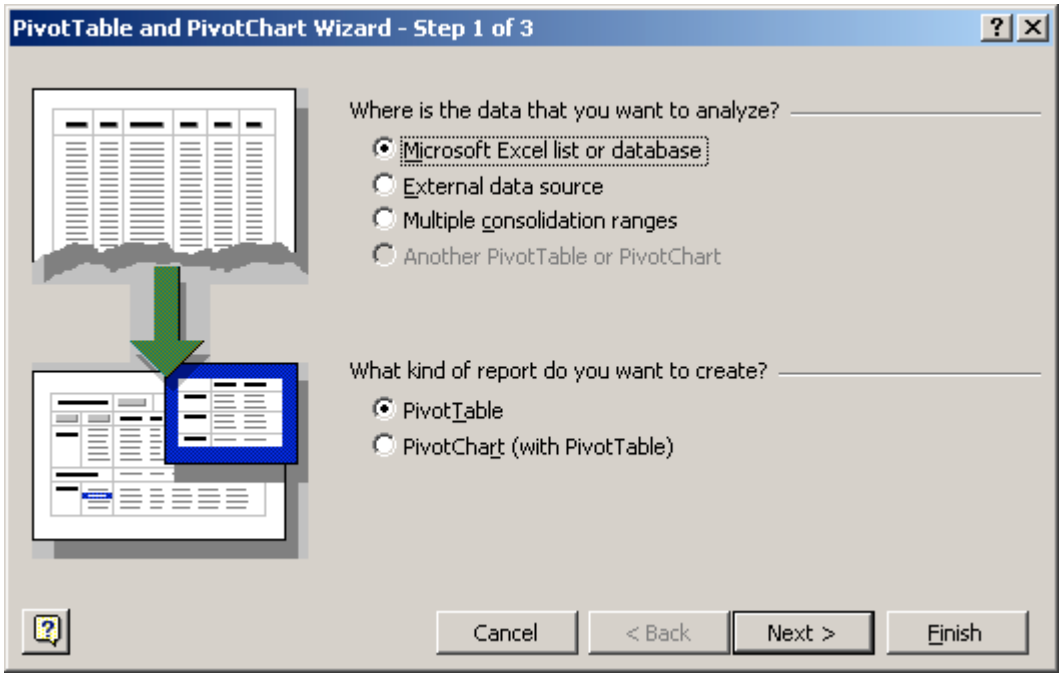
The inner loops ranging over the sectoral indices *i* and *k*, are contained in the outer loop over the reporting years *tr*.

In many cases it is desirable to compare the results of different simulations. In this case, all one has to do is to concatenate the different output files into a single file. Only the first file should contain the field descriptors. All subsequent files should only contain data records. There are a variety of ways to concatenate files. The DOS copy command works well, as does cutting and pasting from any editor or word processor (remembering to save the file as text only if that is necessary). (N.B. Some of the result files may have spurious blank lines that will show up as blank records in the pivot table. These can either be deleted in an editor, or else the blank records can be hidden from within the pivot table.) GAMS will also allow for concatenation of files. Use the '.ap' option on a file to append data to an existing file (for example report.ap=1).

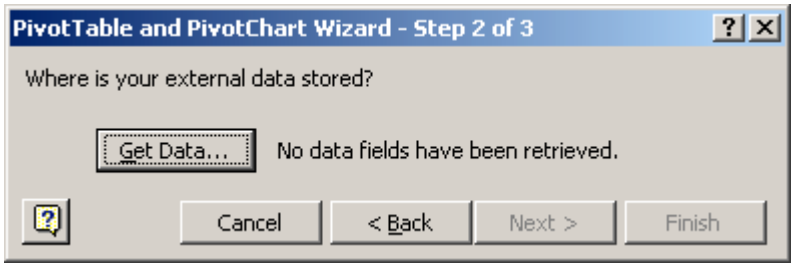
Excel Pivot Tables

To create a pivot table from a text (or CSV) file, start Excel.[20] Typically one initiates the process from a blank work sheet, but this is not strictly necessary. Under the Data Menu, choose the item PivotTable and PivotChart Report..., which starts the PivotTable Wizard and brings up the following screen:

[20] This assumes that the pivot table function, with all of its options have been fully installed during the installation of Microsoft Office. In particular, reading pivot tables from external databases requires the installation of Microsoft Query, plus the pre-defined database types available with Microsoft Query. For our purposes, these pre-defined databases include text and csv files. While the windows and examples in this document have been prepared with Office 2000 and Windows 2000, the functions should be similar using other versions of Excel (starting with Version 5 and other operating systems).

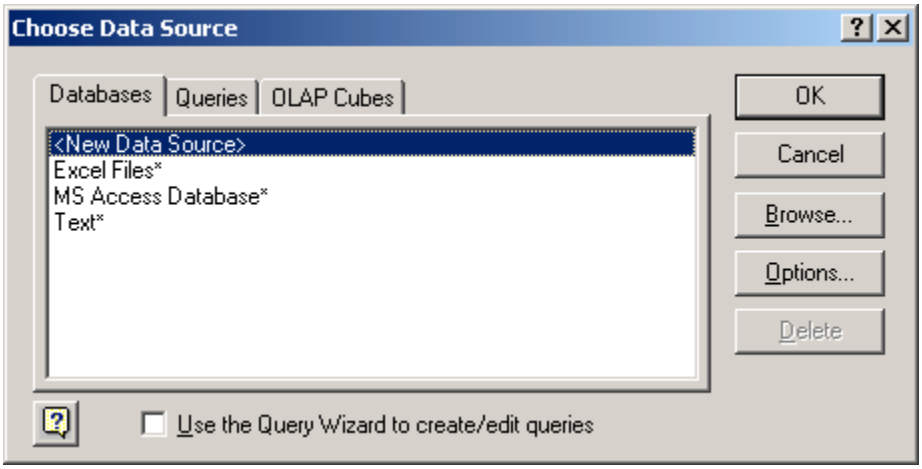


Since the data is located in an external text file click on the button indicated by External data source and then click on the Next > button. This will bring up Step 2 of the PivotTable Wizard, which is indicated by the following screen:

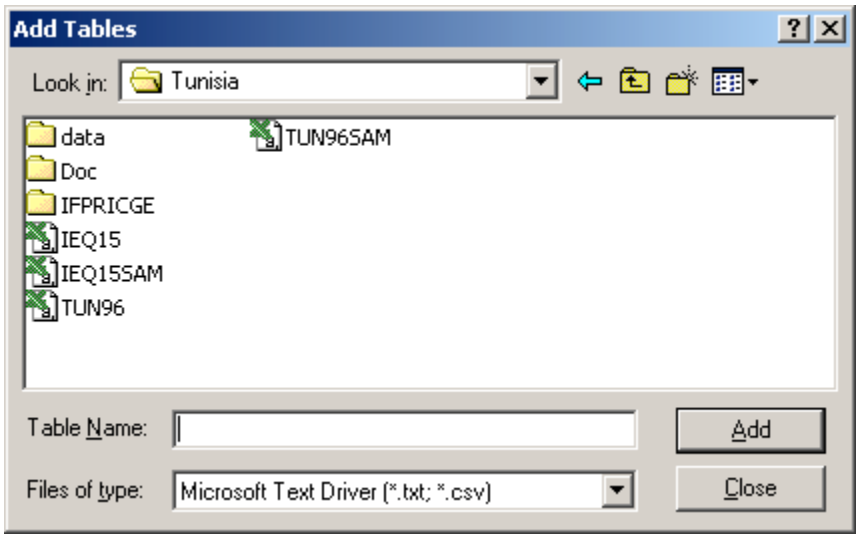


This step proceeds to retrieve the data from the external data source, it actually builds a link between the Excel pivot table and the data in the text file. To retrieve the data, click on the button Get Data... This action will start a new procedure known as Microsoft Query, which is built into MS Office, and starts with the following screen:[21]

[21] It might be necessary to add the correct driver for reading text files. See the end of this appendix for how to add drivers for the MS Query package.

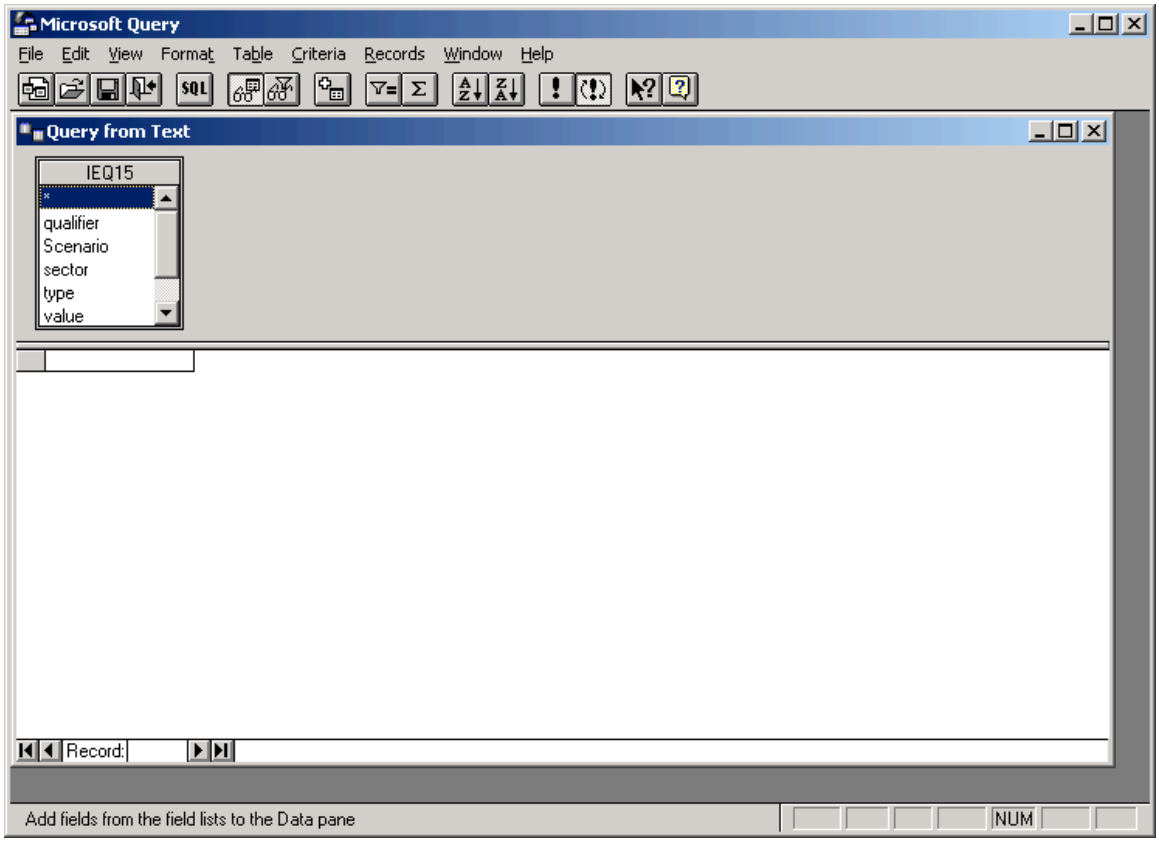


Click once on Text and then on the OK button, or double click on the text choice. (**Important**: Uncheck the Use the Query Wizard to create/edit queries at the bottom of the dialogue box if it is checked.) Microsoft Query will then bring up the standard open file dialogue box, for example:

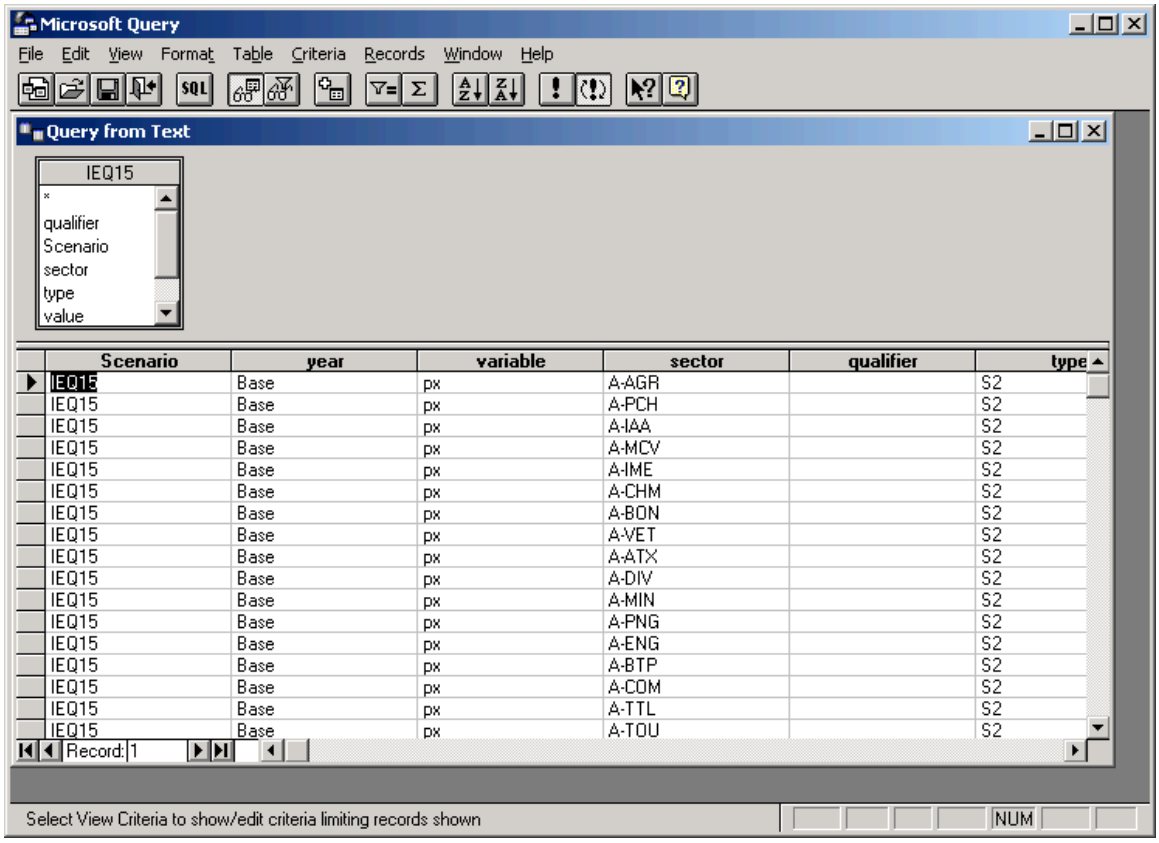


It may be necessary to navigate to the correct sub-directory. By default, only files ending in '.csv' and '.txt' will appear in the open file dialogue box. Once the correct directory has been located, choose the name of the file containing the GAMS results in database format. The file extension should be either **txt** or **csv**. Click (once) on the appropriate file name, and then click once on the Add, followed by clicking once on the Close button.

Assuming a link has been made with the external file (called IEQ15 in this case), Microsoft Query will parse the first line of the database into the different fields. The next window will look something like the following:

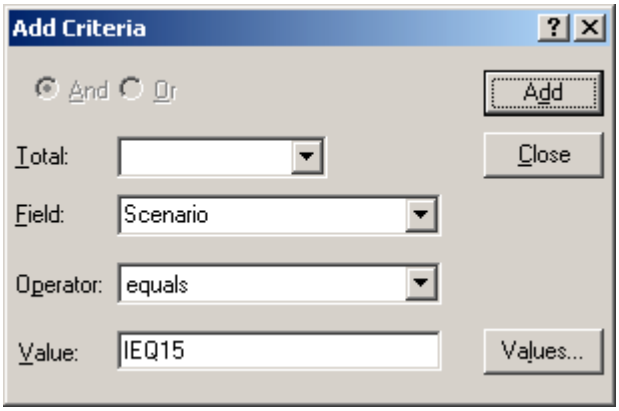


Drag the highlighted rectangle containing the asterisk into the blank rectangle in the bottom half of the screen. This indicates to Microsoft Query that all fields will be passed forward to the pivot table. Upon successful completion of this step, the main Microsoft Query window will have an appearance similar to the following:

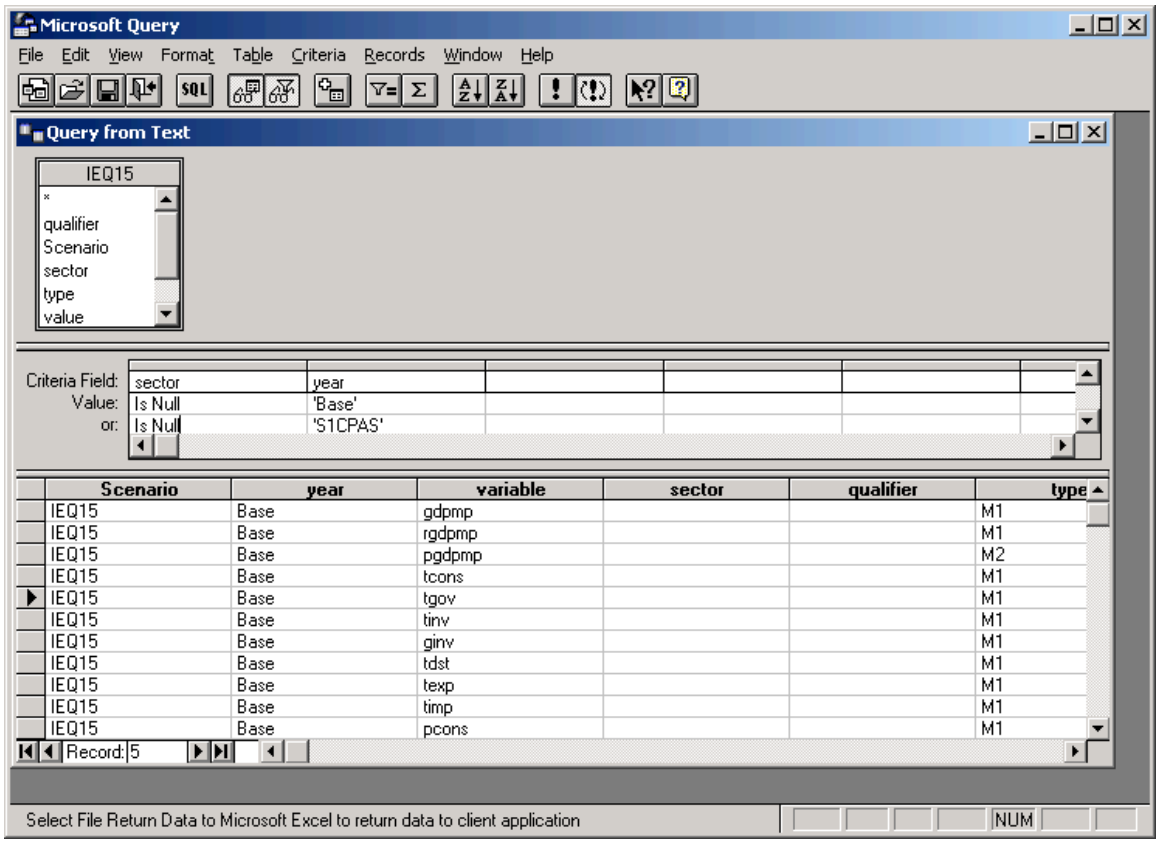


where each field is already laid out in tabular format, with the field headers at the top of the data matrix, and each data record on a corresponding line of the data matrix. One of the more powerful features of Microsoft Query is the ability to filter the data before transferring it to Excel. Filtering is done by using the Add Criteria... item under the Criteria Menu. This document is only meant to be an introduction to the use of pivot tables, so it will not go into all the many different uses of the add criteria function. Some examples may help in providing some initial insights into its usefulness.

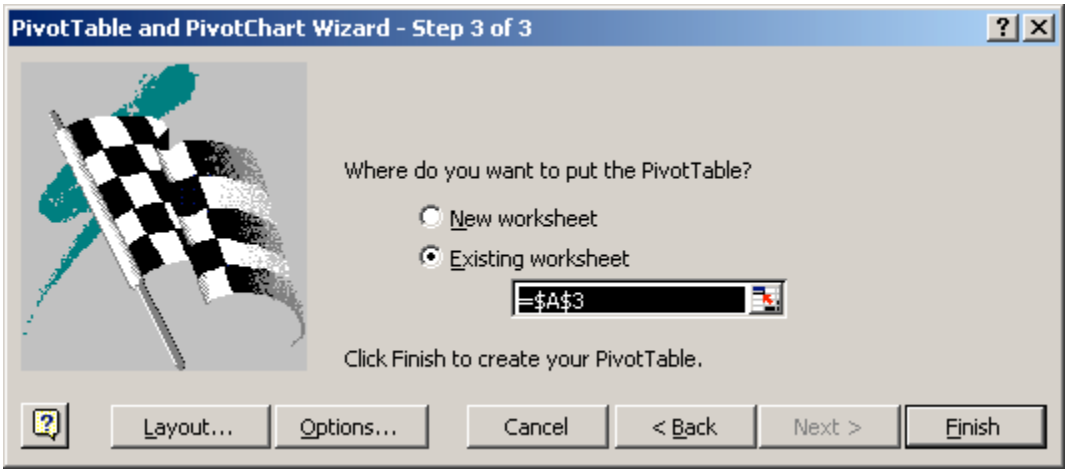
The Add Criteria... menu item brings up the following window:



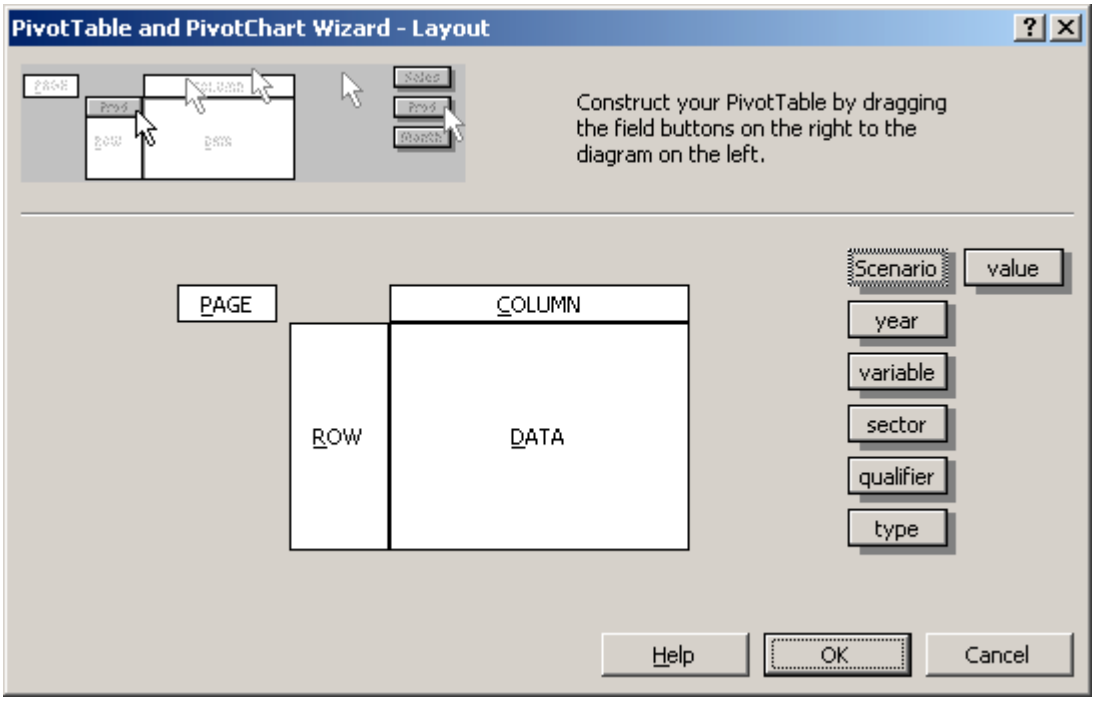
The criteria can be defined over any field and can be filtered in many ways. For example, to select only macro data, i.e. records with no data in the Sector field simply choose the field Sector in the text box designated by Field and choose the filter "is Null" in the text box designated by Operator. When finished with the first criteria, click on the Add button. You can add additional qualifiers (using both the Add and Or operators). In the example shown below, all macro variables for select years will be returned to Excel. When finished, click on the close button. This will return the user back to the main Microsoft Query window:



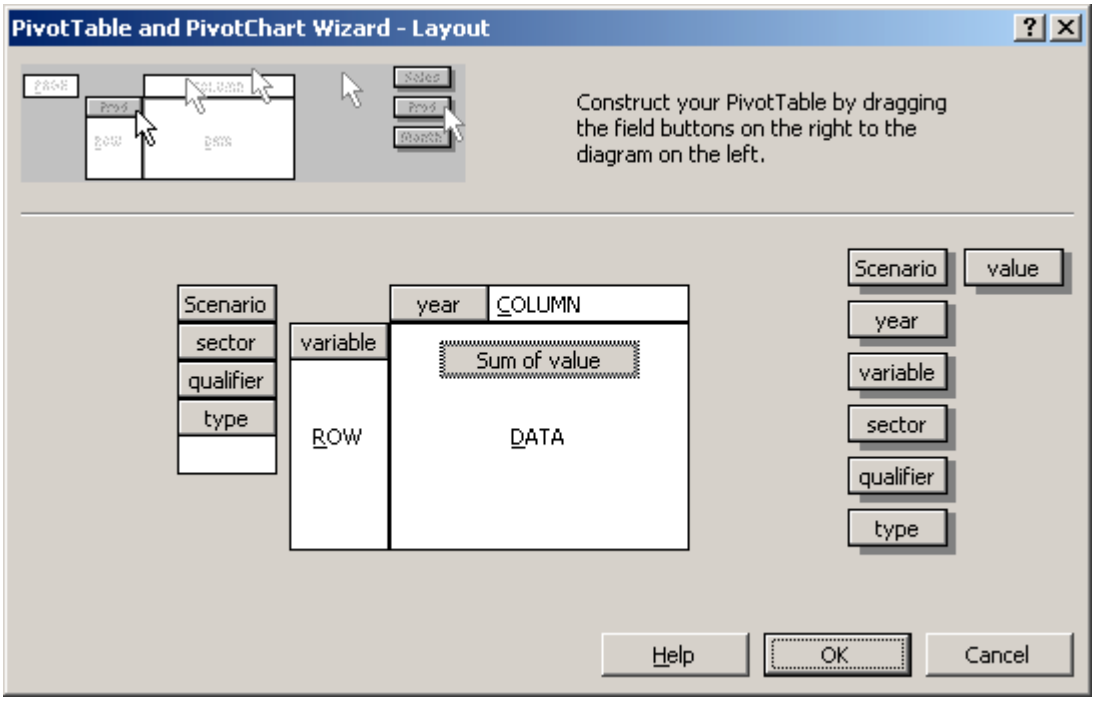
After selecting the data and optionally adding any filtering criteria, the next step is to send the data back to the Excel PivotTable Wizard. From the Microsoft Query File Menu, click on the item Return Data to Microsoft Excel. This sends the user back to step 2 of the Pivot Table Wizard. Click on the Next > button to start step 3 of the PivotTable Wizard, which presents the following screen:



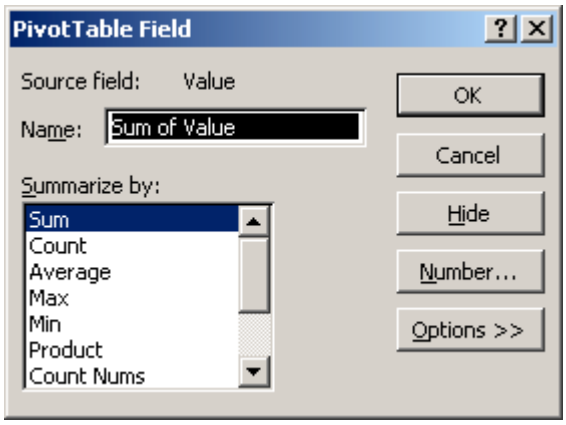
This step allows the user to designate the structure of the pivot table and options. Note that the structure is very easy to modify later once the initial pivot table is constructed. Click first on the Layout… button. The following screen appears:



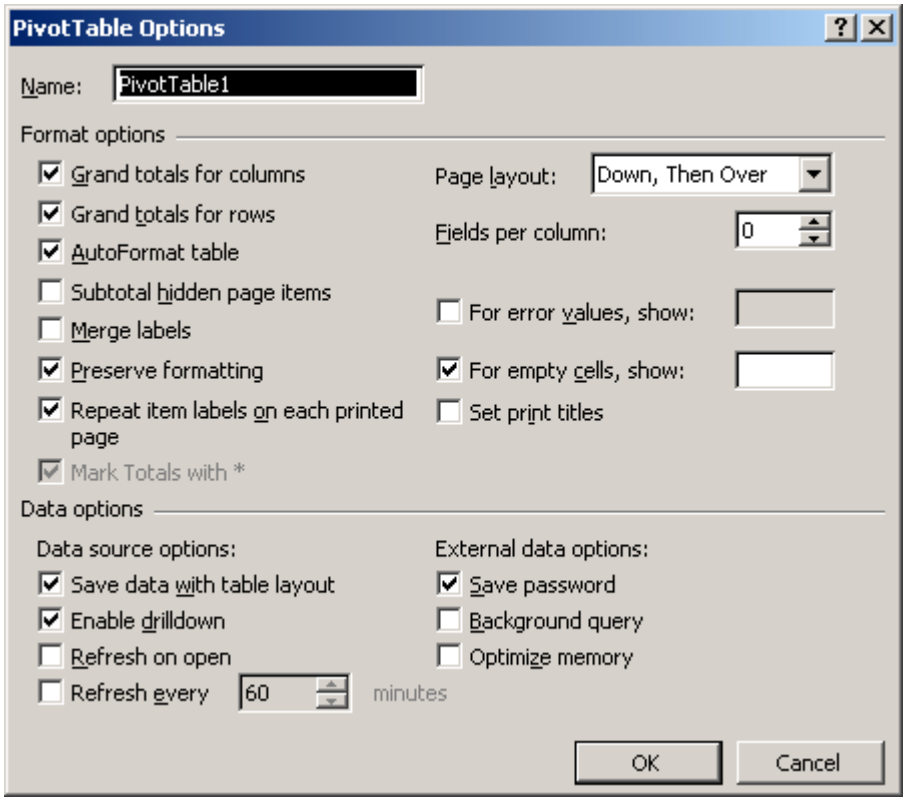
Pivot tables have four dimensions, though in this example, there are seven fields. Therefore, one or more dimensions of the pivot table will have multiple fields. The 'Value' field virtually always goes in the Data section of the pivot table. The other six fields can be arranged in a variety of ways. If the file contains data from only one scenario (i.e. the 'Scenario' field is uniform), and the data only contains macro results, than it is traditional to put the 'Scenario', 'Sector', 'Qualifier' and 'Type' fields in the Page dimension, let 'Year' represent the Columns, and put the 'Variable' in the Rows. For results from more than one scenario, it might be practical to have the 'Scenario' and 'Year' side by side in the Columns. The former structure is depicted by the next window:



There are still a variety of options which can be set from the window above (i.e. in Step 3), though some can also be done at anytime once the pivot table is finished. One option is to define the format for the data. Doubling clicking on the Sum of Value button brings up the following window:



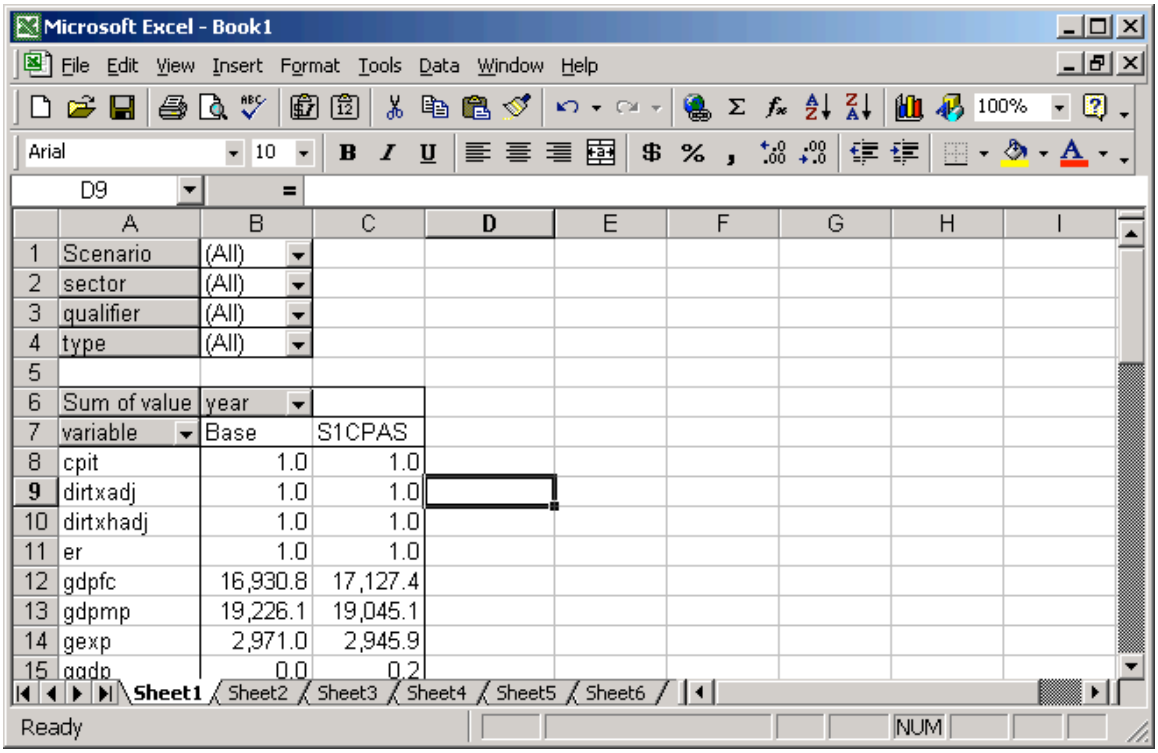
Double clicking on the Number... button brings up the usual Format Cells Excel dialogue. Choose any appropriate format and then click on the OK button twice to return to the PivotTable Wizard Step 3 window. After having finished the layout under step 3 , click on the Options… button. The following window appears:



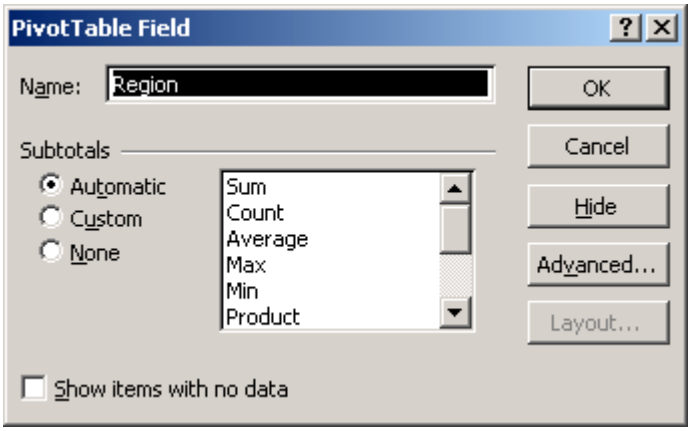
Typically, given the nature of the data in the pivot table, it does not make sense to add the data in the columns or in the rows. The fields Grand totals for columns and Grand totals for rows should be unchecked in this case. There are clearly times when the totals make sense and can be useful. For example, the row and column sums are useful in the SAM pivot tables. It is normal usage to uncheck the AutoFormat table option. It is possible to have Pivot tables calculate percent differences automatically (see the Options >> button from the previous dialogue box). This can be useful in providing tables of growth rates or percent difference across simulations. If there are any divisions by zero, these can be excluded from the pivot table by checking the For error values, show: option above and leaving the text box blank. Similarly, blank cells can be filled with alternative text, such as ".." for example. After clicking on the OK button, the final step is click on the Finish button from the Step 3 window of the PivotWizard.

The appearance of the final pivot table will have a form similar to the following window:

The table is not meaningful because all macro variables are aggregated together due to the value of '(All)' in the variable list. To simply see GDP for example, choose the variable 'rgdpmp' from the drop down list to see the following:



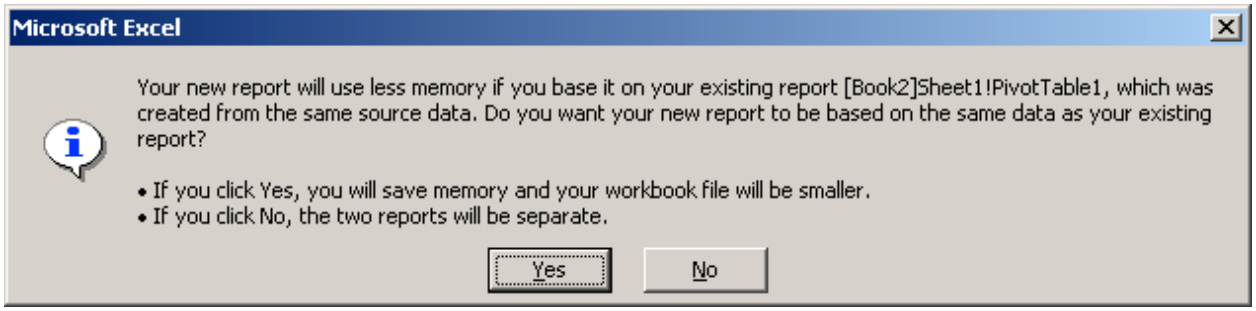
Once the pivot table is finished, it is possible to re-structure the pivot tables in different ways. For example, it is possible to move fields from one location to another. The 'type' field can be moved into the row field with the variables, etc. It is also possible to change some options for specific fields. For example, it is possible to hide certain variables or years, or to have sums calculated or suppressed for specific fields. Double clicking on any one of the field names will bring up the following screen:



From within the pivot table it is possible to re-arrange the ordering of rows and columns by dragging and moving specific labels.

It is also possible for the same Excel file to contain more than one pivot table. For example, one worksheet may contain macro data only. A second worksheet may be loaded with a pivot table, which contains all of the sectoral data, etc. Creating a second pivot table simply requires going

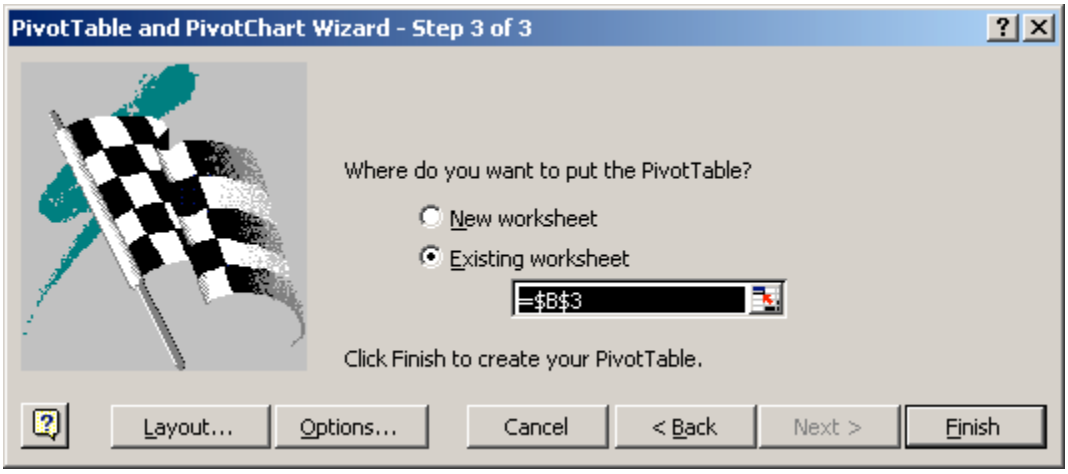
through all of the same steps as defined above, but using different criteria to provide a different lens for analyzing the output. (The PivotTable Wizard will at some point bring up the following dialogue when creating multiple pivot tables in the same Excel file:



It is best to answer no since it relies on the criteria specified in Microsoft Query. Unless you desire to see the same data (with the same criteria) with a different view, the correct answer is no. Alternatively, the initial pivot table can contain all data from the external file and subsequent pivot tables can be linked to the original pivot table. The distinct advantage of this is that hitting the refresh key to update one pivot table, will refresh all linked pivot tables.

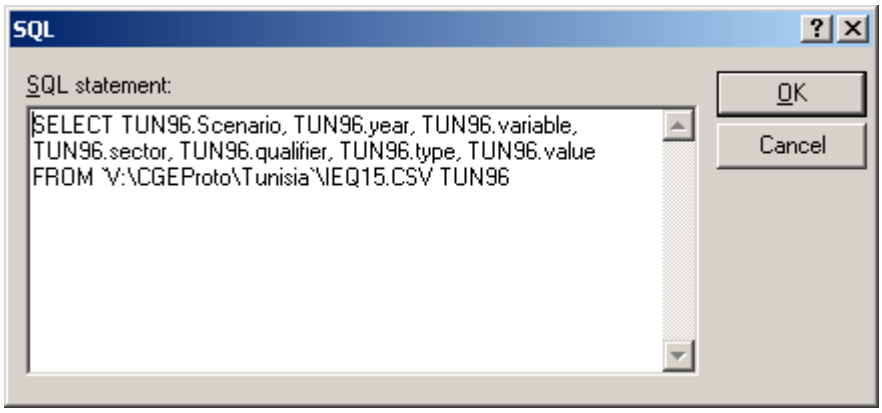
One potential problem with pivot tables is that the link to the external file is hard coded in the Excel file—including the full path name. If the external data file is moved to a different directory or the directory structure on another machine differs, then the link is broken and the refresh data operation will fail.

One way to repair this is to go back to Step 2 of the Wizard. From any cell in a pivot table, right click the mouse button and click on the Wizard… option. The following screen will appear:



Click on the < Back button to get the next screen:

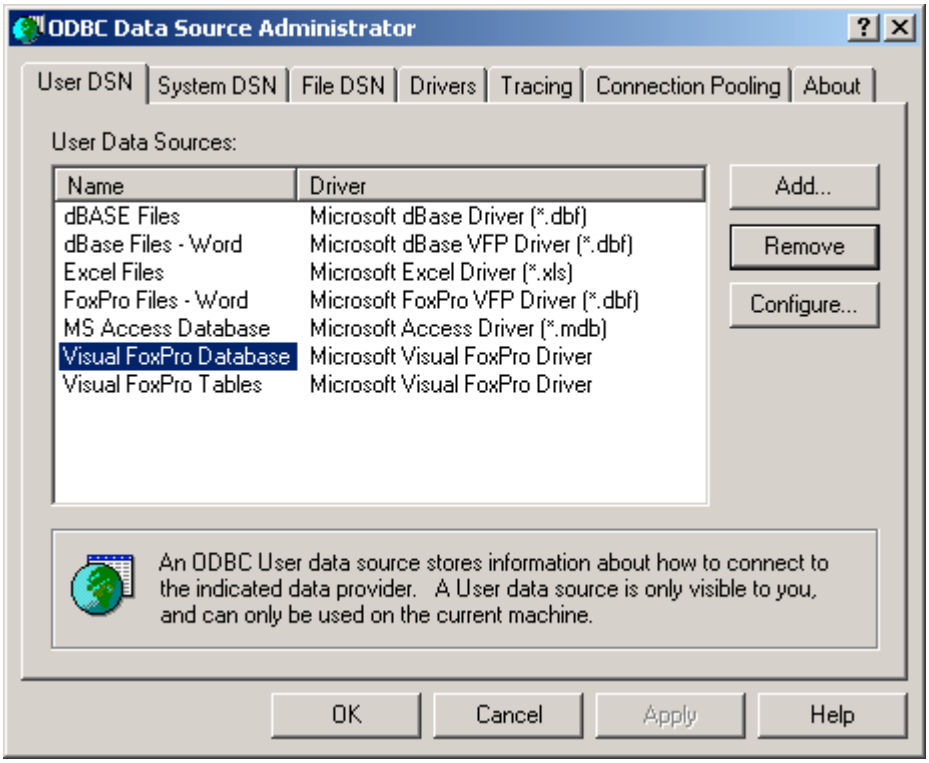
Click on the Get Data… button to restart Microsoft Query. Since the link has been broken, the data part of the Microsoft Query window will be blank. Go to the View/SQL… option from the Microsoft Query Menu. This will pop open a text window similar to the following:



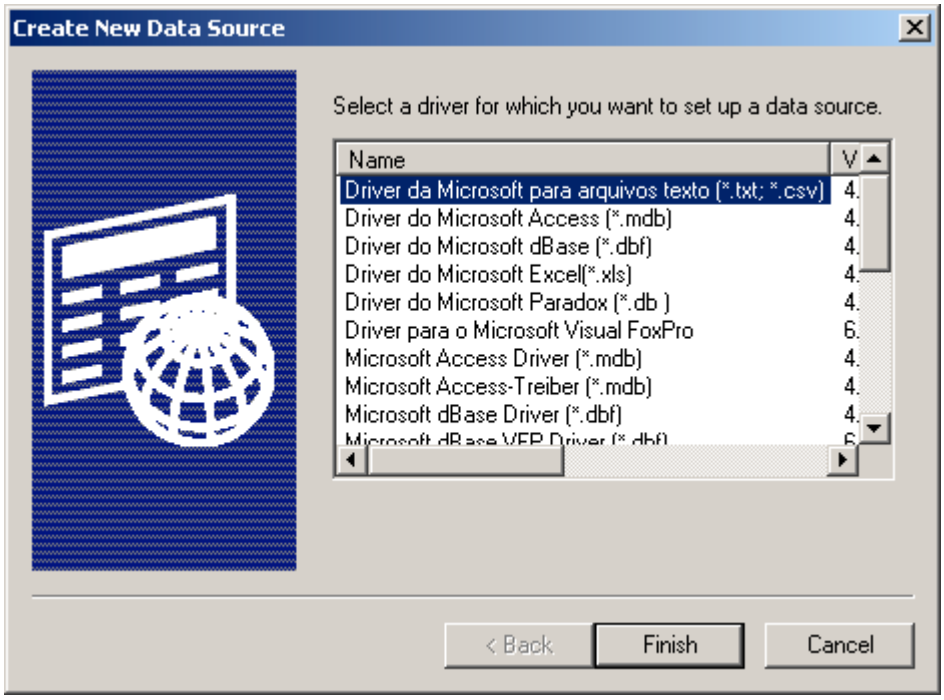
The full path name is provided as an argument to the 'FROM' part of the SQL statement. Simply replace the invalid path name with a valid path name. Be careful with the backward single quotes that are required by the SQL statement. The file name can also be replaced. (N.B. This is one way to replicate a standard report for various different scenario files—they could even be in the same path.) Click on OK. This should automatically fill the data part of the Microsoft Query window. Then simply hit the File/Return Data to Microsoft Excel menu item. The data in the pivot table will then be refreshed.

Adding drivers for Microsoft Query

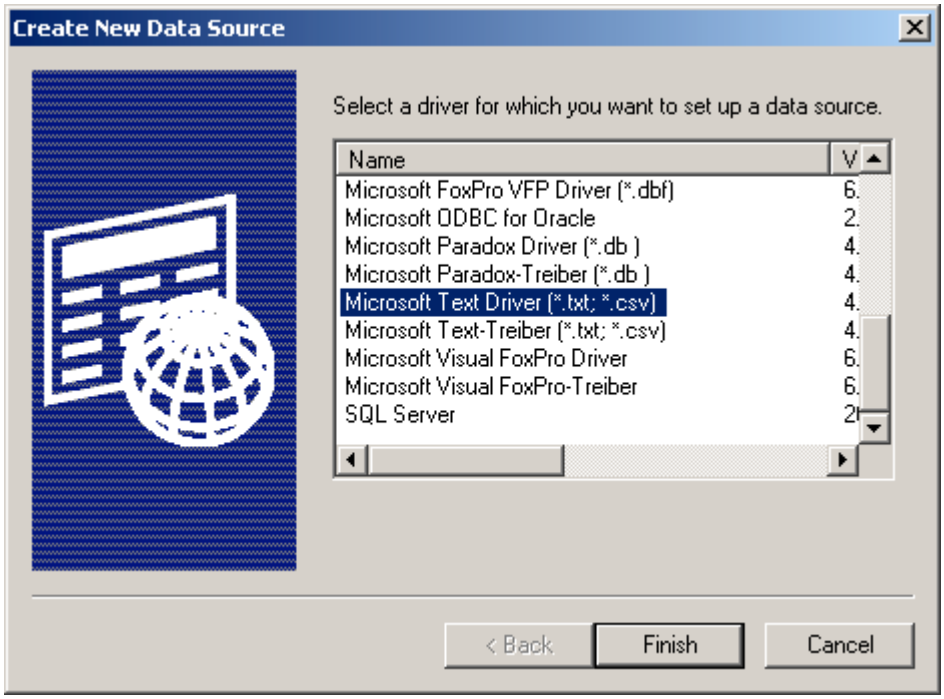
The pivot tables created by the GAMS package are all text-based CSV files. Windows comes packaged with a number of database drivers, but not all are installed by default, and typically the text driver is not installed. The following instructions show how to install the text driver if it is not already available. This needs to be done only once. First, enter the control panel and open the folder called 'Administrative Tools'. Click on the file called 'Data Sources (ODBC)'. This will open the application to load a database driver. A screen similar to the following will appear:



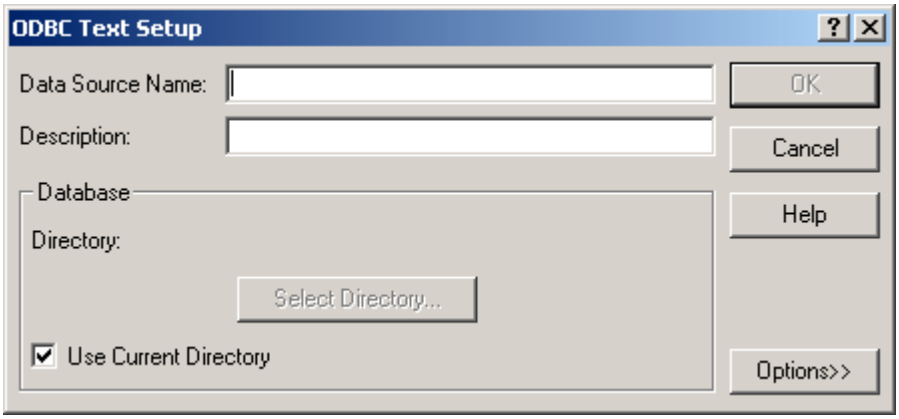
Click on Add… to see the following:



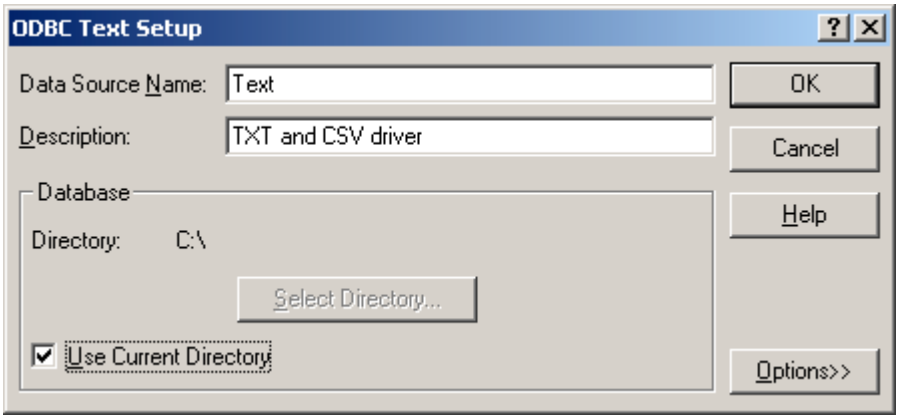
This provides a list of all drivers available with the current operating system. Scroll down to the Microsoft text driver:



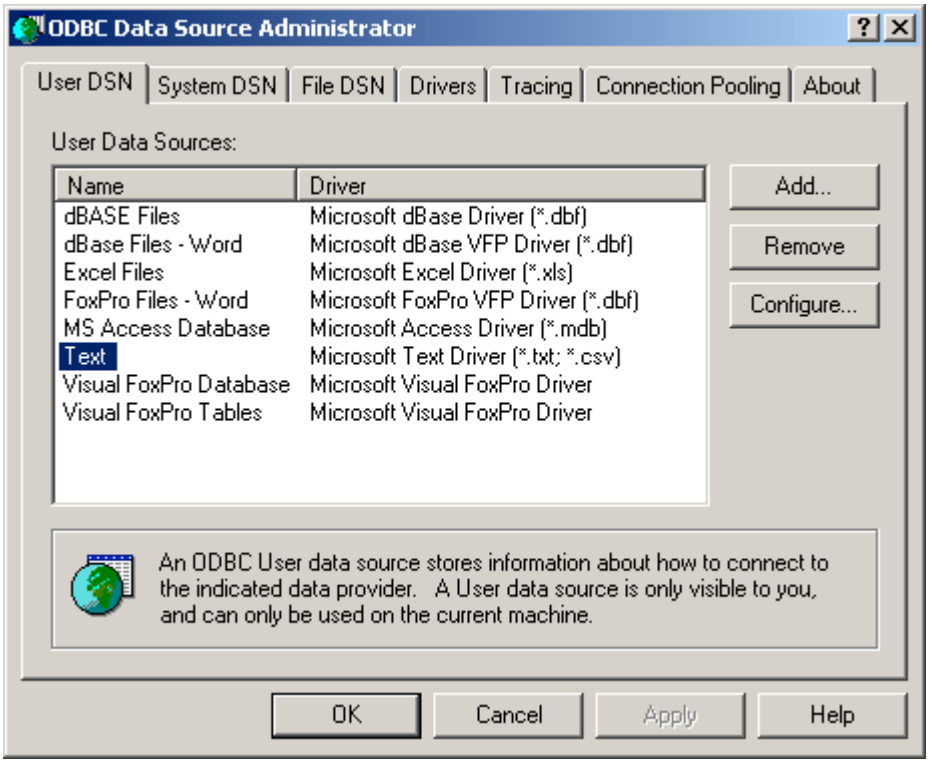
Then click on finish. The following screen appears:



Fill in the text boxes with the following information:



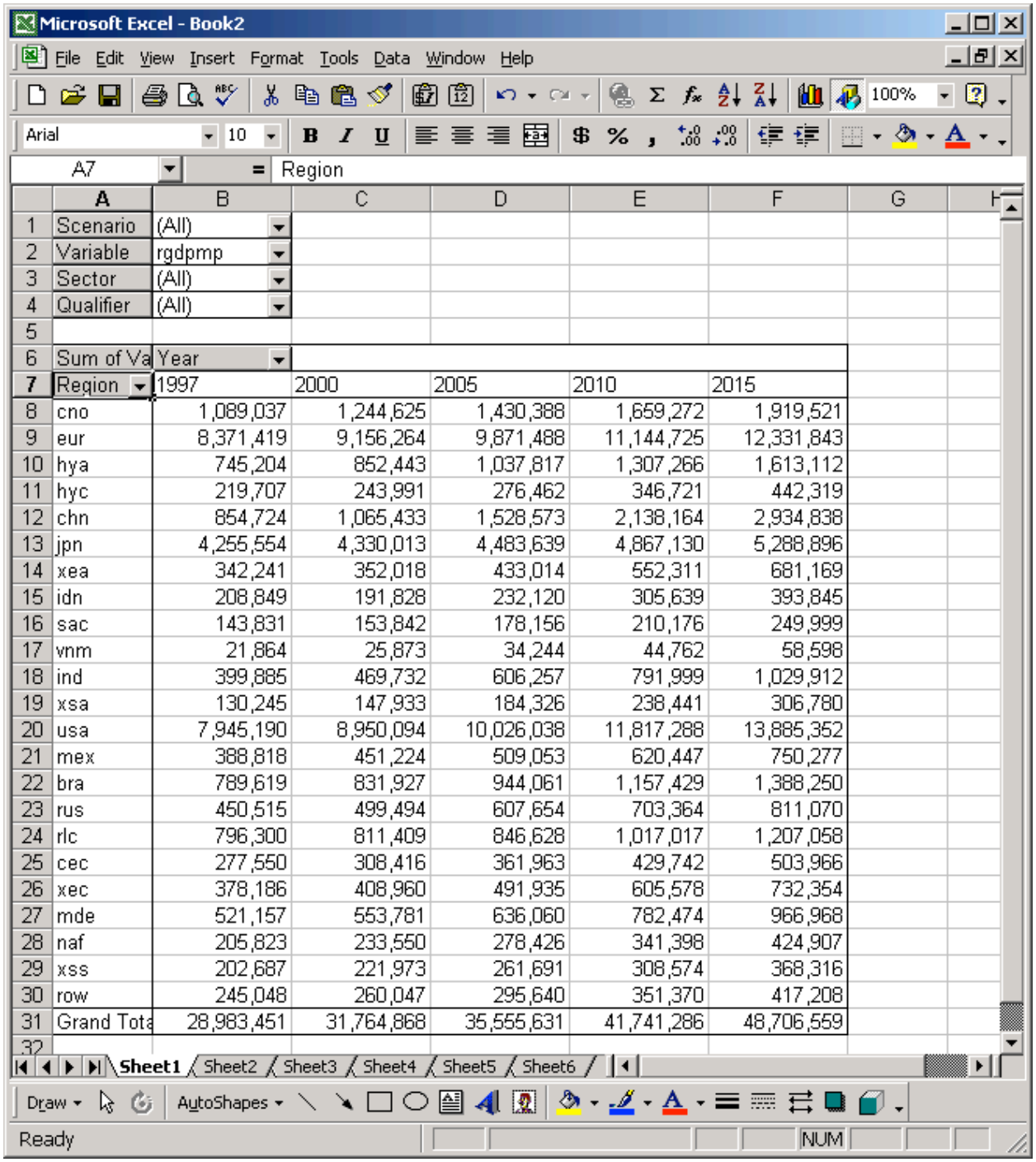
Then click on OK. This will return control back to the main dialogue box:



The text driver has been added. Click on OK once more to complete the installation.

Custom lists and sorting in Excel

It is easy to sort data in pivot tables, particularly if you have created a custom list. The following shows an unordered Pivot table:



Microsoft Excel - Book2

File Edit View Insert Format Tools Data Window Help

Arial 10 B I U

A7 = Region

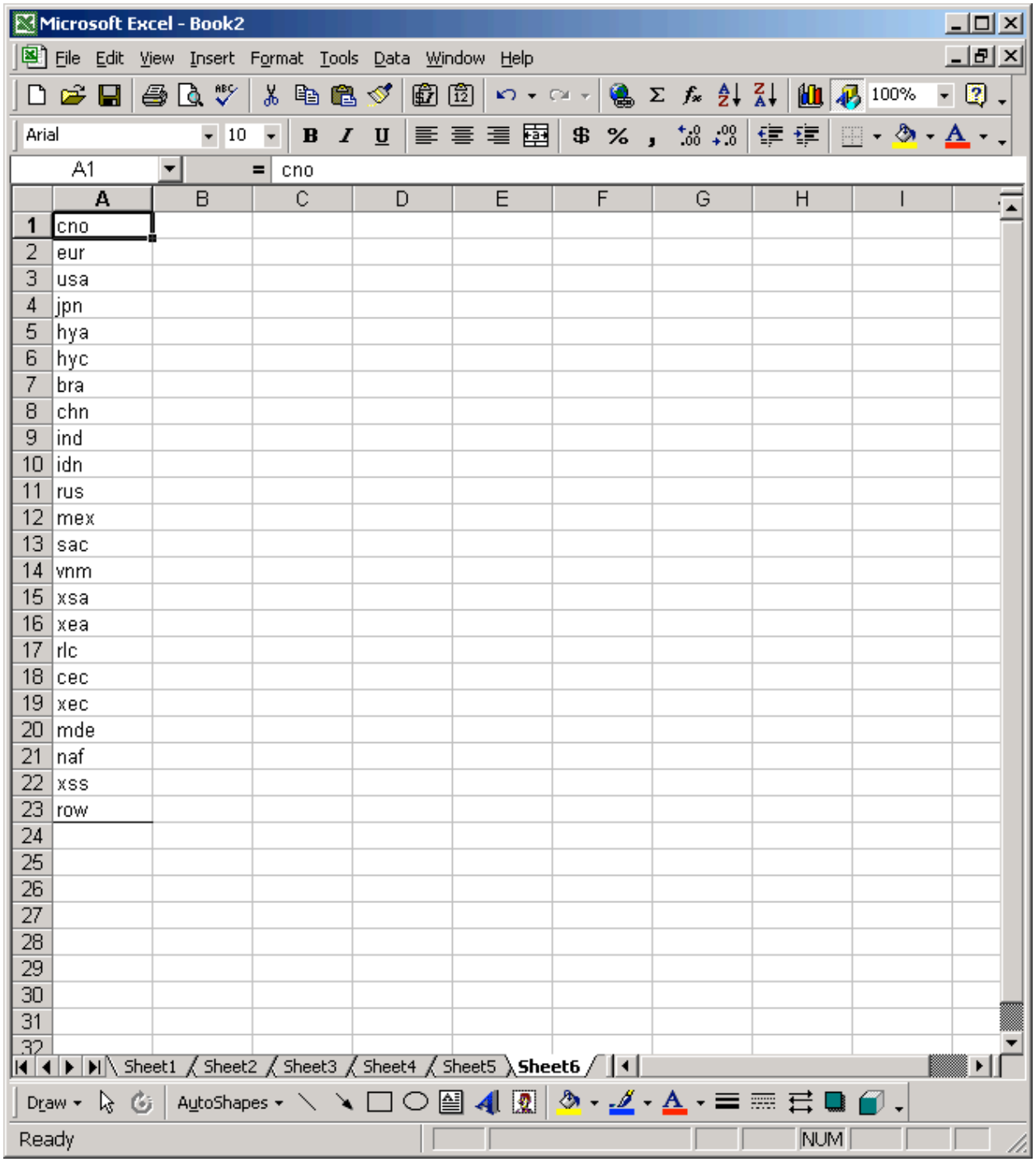
	A	B	C	D	E	F	G
1	Scenario	(All)					
2	Variable	rgdpmp					
3	Sector	(All)					
4	Qualifier	(All)					
5							
6	Sum of Va	Year					
7	Region	1997	2000	2005	2010	2015	
8	cno	1,089,037	1,244,625	1,430,388	1,659,272	1,919,521	
9	eur	8,371,419	9,156,264	9,871,488	11,144,725	12,331,843	
10	hya	745,204	852,443	1,037,817	1,307,266	1,613,112	
11	hyc	219,707	243,991	276,462	346,721	442,319	
12	chn	854,724	1,065,433	1,528,573	2,138,164	2,934,838	
13	jpn	4,255,554	4,330,013	4,483,639	4,867,130	5,288,896	
14	xea	342,241	352,018	433,014	552,311	681,169	
15	idn	208,849	191,828	232,120	305,639	393,845	
16	sac	143,831	153,842	178,156	210,176	249,999	
17	vnm	21,864	25,873	34,244	44,762	58,598	
18	ind	399,885	469,732	606,257	791,999	1,029,912	
19	xsa	130,245	147,933	184,326	238,441	306,780	
20	usa	7,945,190	8,950,094	10,026,038	11,817,288	13,885,352	
21	mex	388,818	451,224	509,053	620,447	750,277	
22	bra	789,619	831,927	944,061	1,157,429	1,388,250	
23	rus	450,515	499,494	607,654	703,364	811,070	
24	rlc	796,300	811,409	846,628	1,017,017	1,207,058	
25	cec	277,550	308,416	361,963	429,742	503,966	
26	xec	378,186	408,960	491,935	605,578	732,354	
27	mde	521,157	553,781	636,060	782,474	966,968	
28	naf	205,823	233,550	278,426	341,398	424,907	
29	xss	202,687	221,973	261,691	308,574	368,316	
30	row	245,048	260,047	295,640	351,370	417,208	
31	Grand Tota	28,983,451	31,764,868	35,555,631	41,741,286	48,706,559	
32							

Sheet1 Sheet2 Sheet3 Sheet4 Sheet5 Sheet6

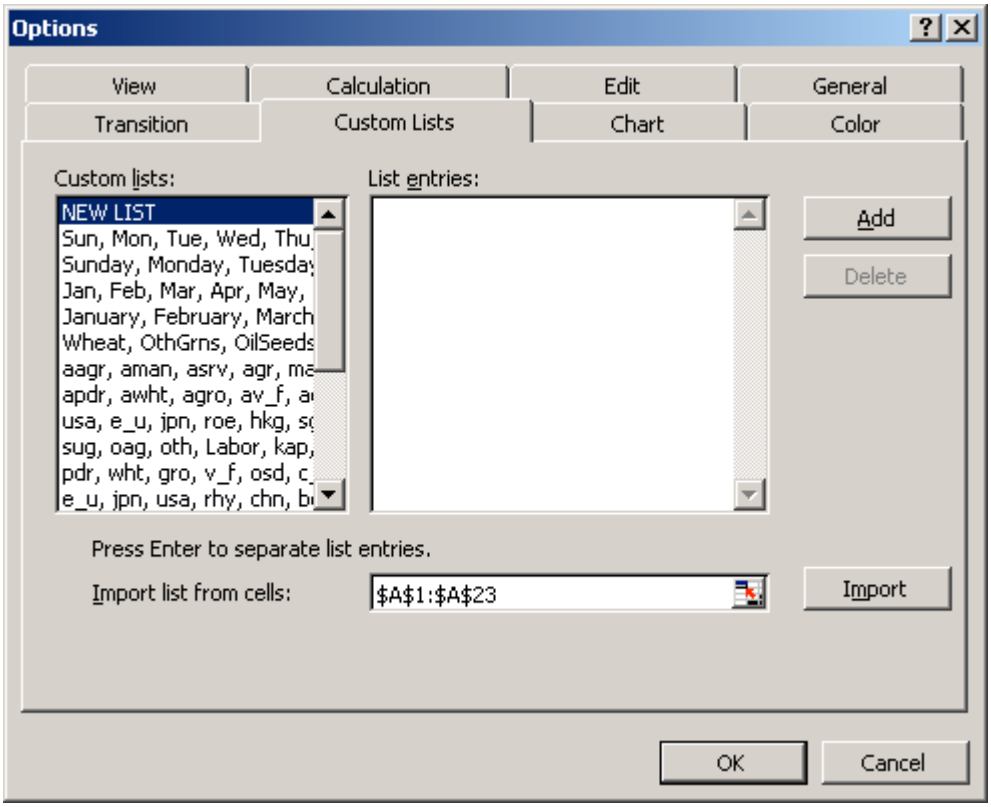
Draw AutoShapes

Ready NUM

Say the desired order is the following:

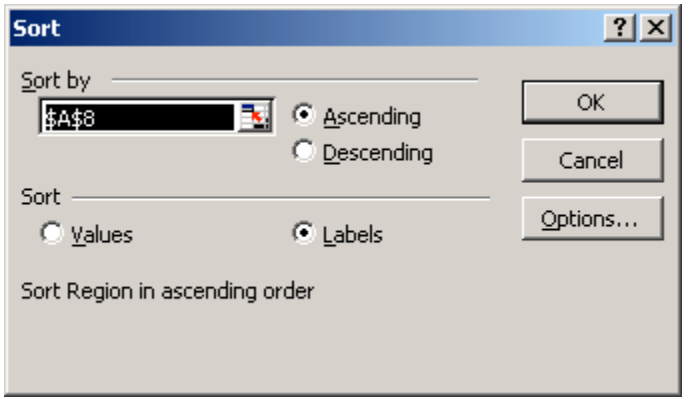


The labels are entered in the right order and as values. (This procedure cannot be used on text formulas.) From the Tools menu, choose, Options, and then Custom lists. Click on the Import box and select the labels. The dialogue box should look like the following:

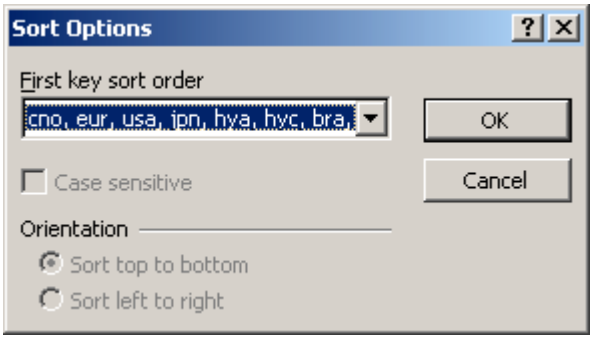


Click on the OK button.

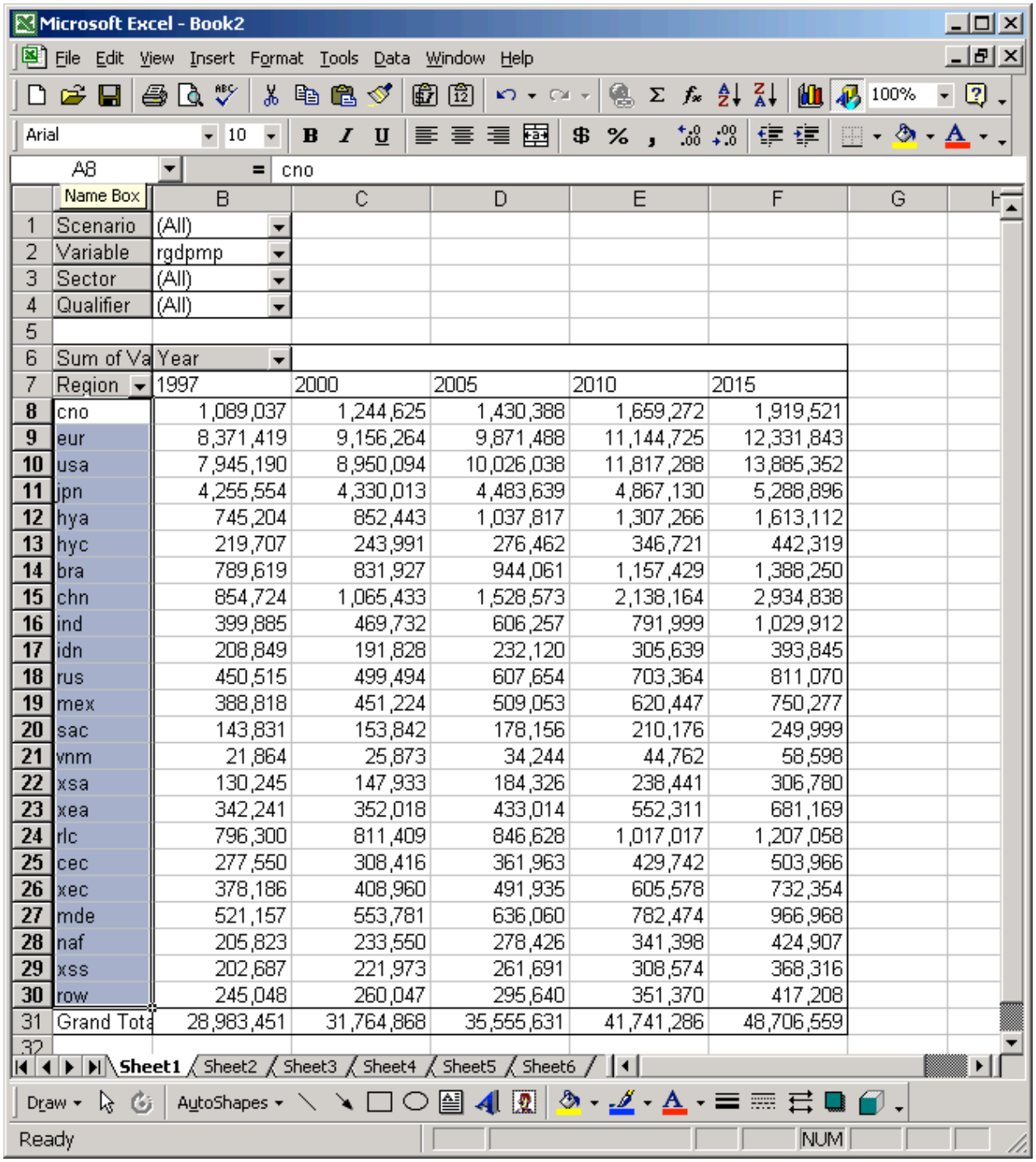
To sort the data table by region, click once near the top of the region tab, until the cursor turns into a downward pointing arrow. This should highlight the region labels. From the Data menu, select the Sort option. To bring up the following screen:



Click on Options, which brings up the following:



It may be necessary to click on the tab to select the right sort key. Then click on OK. Click on OK once more in the main Sort dialogue. This will implement the sort. The final pivot table will look like:



Microsoft Excel - Book2

File Edit View Insert Format Tools Data Window Help

Arial 10

A8 = cno

	Name Box	B	C	D	E	F	G
1	Scenario	(All)					
2	Variable	rgdpmp					
3	Sector	(All)					
4	Qualifier	(All)					
5							
6	Sum of Va	Year					
7	Region	1997	2000	2005	2010	2015	
8	cno	1,089,037	1,244,625	1,430,388	1,659,272	1,919,521	
9	eur	8,371,419	9,156,264	9,871,488	11,144,725	12,331,843	
10	usa	7,945,190	8,950,094	10,026,038	11,817,288	13,885,352	
11	jpn	4,255,554	4,330,013	4,483,639	4,867,130	5,288,896	
12	hya	745,204	852,443	1,037,817	1,307,266	1,613,112	
13	hyc	219,707	243,991	276,462	346,721	442,319	
14	bra	789,619	831,927	944,061	1,157,429	1,388,250	
15	chn	854,724	1,065,433	1,528,573	2,138,164	2,934,838	
16	ind	399,885	469,732	606,257	791,999	1,029,912	
17	idn	208,849	191,828	232,120	305,639	393,845	
18	rus	450,515	499,494	607,654	703,364	811,070	
19	mex	388,818	451,224	509,053	620,447	750,277	
20	sac	143,831	153,842	178,156	210,176	249,999	
21	vnm	21,864	25,873	34,244	44,762	58,598	
22	xsa	130,245	147,933	184,326	238,441	306,780	
23	xea	342,241	352,018	433,014	552,311	681,169	
24	rlc	796,300	811,409	846,628	1,017,017	1,207,058	
25	cec	277,550	308,416	361,963	429,742	503,966	
26	xec	378,186	408,960	491,935	605,578	732,354	
27	mde	521,157	553,781	636,060	782,474	966,968	
28	naf	205,823	233,550	278,426	341,398	424,907	
29	xss	202,687	221,973	261,691	308,574	368,316	
30	row	245,048	260,047	295,640	351,370	417,208	
31	Grand Tota	28,983,451	31,764,868	35,555,631	41,741,286	48,706,559	
32							

Sheet1 Sheet2 Sheet3 Sheet4 Sheet5 Sheet6

Draw AutoShapes

Ready NUM

The custom lists and sort fields feature is particularly useful for structuring the SAM pivot sheets. It helps to create the custom lists before creating the pivot tables. In most cases, if Excel recognizes the data in the external files, it will automatically sort the data to a pre-existing custom list.