EViews® 10

User's Guide I

IHS Markit

EViews 10 User's Guide I

EViews 10 User's Guide I

Table of Contents

EVIEWS 10 USER'S GUIDE II 1

Preface

The EViews documentation is divided into four volumes.

The first two *User's Guide* volumes provide basic documentation on using EViews. *User's Guide I*, describes EViews fundamentals and describes using EViews to perform basic data analysis and display results. The second volume, *User's Guide II*, offers a description of EViews' more advanced statistical and estimation features.

The remaining volumes document the use of commands and programming in EViews. The *Command and Programming Reference* offers details on specific commands and functions, while the *Object Reference* documents the commands for working with views and procedures of objects in EViews.

The material in the *User's Guide I* may be divided into three distinct parts:

- Part I. "EViews Fundamentals," beginning on page 3 introduces you to the basics of using EViews. In addition to a discussion of basic Windows operations, we explain how to use EViews to work with your data.

- Part II. "Basic Data Analysis," beginning on page 399 describes the use of EViews to perform basic analysis of data and to draw graphs and display tables describing the results of analysis of your data.

- Part III. "Customizing Output," beginning on page 749 documents the graph object, table and text objects, and the spool object, all of which are used to produce presentation output.

You need not read the manuals from cover-to-cover in order to use EViews. Once you gain a basic familiarity with the program you should be able to perform most operations without consulting the documentation. We do recommend, however, that you glance at most of Part I. "EViews Fundamentals" to gain familiarity with the basic concepts and operation of the program.

Part I. EViews Fundamentals

The following chapters document the fundamentals of working with EViews:

- Chapter 1. "Introduction" describes the basics of installing EViews and offers basic information to get you started (including tips on places to look for help).

- Chapter 2. "A Demonstration" guides you through a typical EViews session, introducing you to the basics of working with EViews.

- Chapter 3. "Workfile Basics" describes working with workfiles (the containers for your data in EViews).

- Chapter 4. "Object Basics" provides an overview of EViews objects, which are the building blocks for all analysis in EViews.

- Chapter 5. "Basic Data Handling" and Chapter 6. "Working with Data" provide background on the basics of working with numeric data. We describe methods of getting your data into EViews, manipulating and managing your data held in series and group objects, and exporting your data into spreadsheets, text files and other Windows applications.

We recommend that you browse through most of the material in the above sections before beginning serious work with EViews.

The remaining material is somewhat more advanced and may be ignored until needed:

- Chapter 7. "Working with Data (Advanced)," Chapter 8. "Series Links," and Chapter 9. "Advanced Workfiles" describe advanced tools for working with numeric data, and tools for working with different kinds of data (alphanumeric and date series, irregular and panel workfiles).

- Chapter 10. "EViews Databases" describes the EViews database features and advanced data handling features.

This material is relevant only if you wish to work with the advanced tools.

Chapter 1. Introduction

What is EViews?

EViews provides sophisticated data analysis, regression, and forecasting tools on Windows-based computers. With EViews you can quickly develop a statistical relation from your data and then use the relation to forecast future values of the data. Areas where EViews has proven to be useful include: scientific data analysis and evaluation, financial analysis, macroeconomic forecasting, simulation, sales forecasting, and cost analysis.

EViews is a new version of a set of tools for manipulating time series data originally developed in the Time Series Processor software for large computers. The predecessor of EViews was MicroTSP, first released in 1981. Though EViews was developed by economists with an emphasis on time series analysis, there is nothing in its design that limits its usefulness to economic time series.

EViews provides convenient visual ways to enter data series from the keyboard or from files, to create new series from existing ones, to display and print series, and to carry out statistical analysis of the relationships among series.

EViews takes advantage of the visual features of modern Windows software. You can use your mouse to guide the operation with standard Windows menus and dialogs, and capture those operations for later use. Results appear in windows and can be manipulated with standard techniques.

Alternatively, you may use EViews' powerful command and batch processing language. You can enter and edit commands in the command window. You can create and store the commands in programs that document your research project for later execution.

The EViews Window

If the program is installed correctly, you should see the EViews window when you launch the program.

Title bar   Main menu   Command window

EViews

File Edit Object View Proc Quick Options Add-ins Window Help

Command

Path = c:\data   DB = eia   WF = none

Status line   Work area
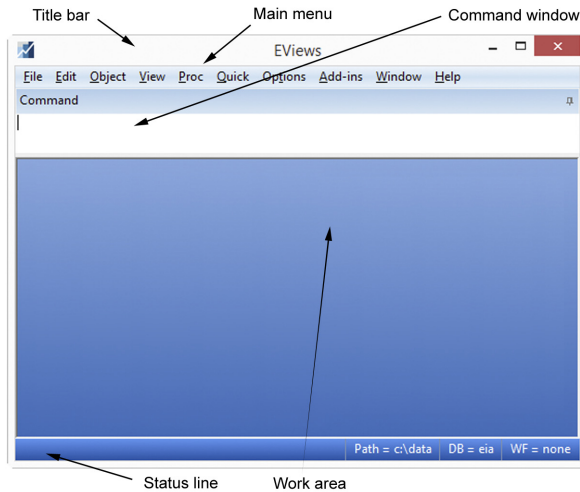
You should familiarize yourself with the following main areas in the EViews window.

The Title Bar

The *title bar*, labeled **EViews**, is at the very top of the main window. When EViews is the active program in Windows, the title bar has a color and intensity that differs from the other windows (generally it is darker). When another program is active, the EViews title bar will be lighter. If another program is active, EViews may be made active by clicking anywhere in the EViews window or by using ALT-TAB to cycle between applications until the EViews window is active.

The Main Menu

Just below the title bar is the *main menu*. If you move the cursor to an entry in the main menu and click on the left mouse button, a *drop-down menu* will appear. Clicking on an entry in the drop-down menu selects the highlighted item.

For example, here we click on the **Object** entry in the main menu to reveal a drop-down menu. Notice that some of the items in the drop-down menu are listed in black and others are in gray. In menus, black items may be executed while the gray items are not available. In this example, you cannot create a **New Object** or **Store** an object, but you can **Print** and **View Options**. We will explain this behavior in our discussion of "The Object Window" on page 112.
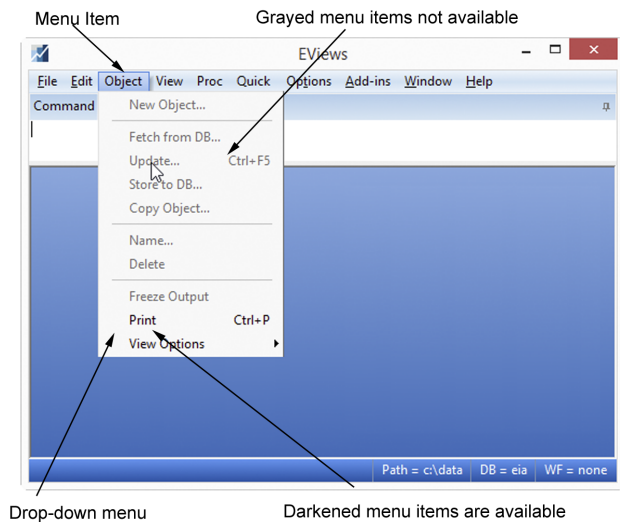
The Command Window

Below the menu bar is an area called the *command window*. EViews commands may be typed in this window. The command is executed as soon as you hit ENTER.

The vertical bar in the command window is called the *insertion point*. It shows where the letters that you type on the keyboard will be placed. As with standard word processors, if you have typed something in the command area, you can move the insertion point by pointing to the new location and clicking the mouse. If the insertion point is not visible or your keystrokes are not appearing in the window, it probably means that the command window is not active (not receiving keyboard focus); simply click anywhere in the command window to tell EViews that you wish to enter commands.
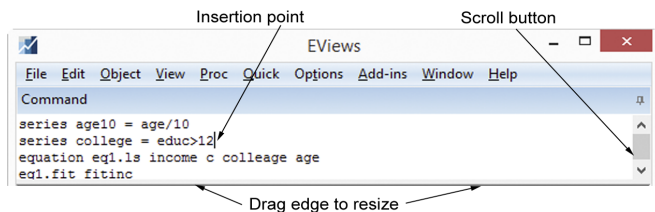
To toggle between the active window and the command window, press F5.

See "Window Behavior" on page 861 of *User's Guide I* for a discussion of focus settings which affect the use of the command window.

Command Editing

The command window supports Windows cut-and-paste so that you can easily move text between the command window, other EViews text windows, and other Windows programs. The contents of the command area may also be saved directly into a text file for later use: make certain that the command window is active by clicking anywhere in the window, and then select **File/Save As…** from the main menu.

If you have entered more commands than will fit in your command window, EViews turns the window into a standard scrollable window. Simply use the scroll bar or up and down arrows on the right-hand side of the window to see various parts of the list of previously executed commands. You may move the insertion point to previously executed commands, edit the existing command, and then press ENTER to execute the edited version of the command.

Alternately, to display a list of previous commands in the order in which they were entered, press the Control key and the UP arrow (CTRL + UP). The last command will be entered into the command window. Holding down the CTRL key and pressing UP repeatedly will display the next prior commands. Repeat until the desired command is recalled.

To examine a history of the last 30 commands, press the Control Key and the J key (CTRL + J). In the popup window you may use the UP and DOWN arrows to select the desired command and then press the ENTER key to add it to the command window, or simply double click on the command. To close the history window without selecting a command, click elsewhere in the command window or press the Escape (ESC) key.
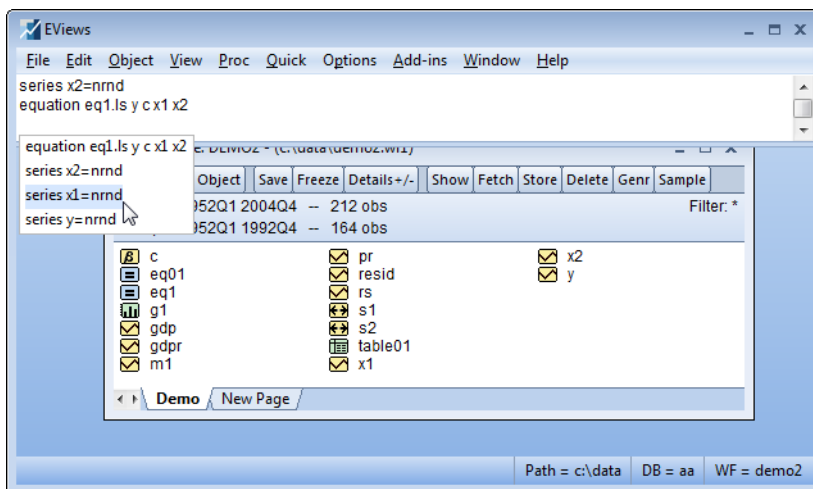
To execute the retrieved command, simply press ENTER again. You may first edit the command if you wish to do so.

You may find that the default size of the command window is too large or small for your needs. You can resize the command window by placing the cursor at the bottom of the command window, holding down the mouse button and dragging the window up or down. Release the mouse button when the command window is the desired size.

Undocking the Command Window

You may undock and drag the command window to anywhere on your windows desktop.
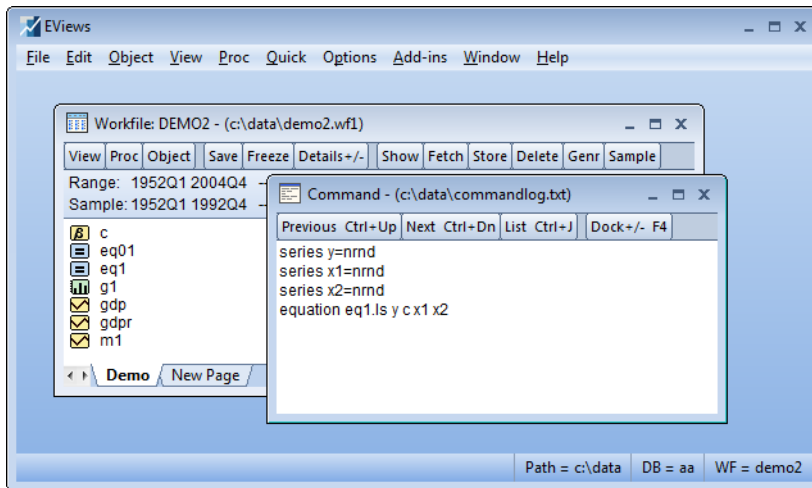
See "Command and Capture Window Docking" on page 6 of *Command and Programming Reference* for more on docking.

The Status Line

At the very bottom of the window is a *status line* which is divided into several sections.

The left section will sometimes contain status messages sent to you by EViews. These status messages can be cleared manu-

ally by clicking on the box at the far left of the status line. The next section shows the *default directory* that EViews will use to look for data and programs. The last two sections display the names of the default database and workfile. In later chapters, we will show you how to change both defaults.
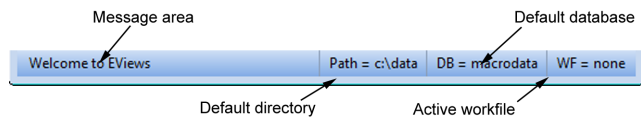
The Work Area

The area in the middle of the window is the work area where EViews will display the various object windows that it creates. Think of these windows as similar to the sheets of paper you might place on your desk as you work. The windows will overlap each other with the foremost window being in *focus* or *active*. Only the active window has a darkened titlebar.

When a window is partly covered, you can bring it to the top by clicking on its titlebar or on a visible portion of the window. You can also cycle through the displayed windows by pressing the F6 or CTRL-TAB keys.

Alternatively, you may select a window by clicking on the **Window** menu item, and selecting the desired name.

You can move a window by clicking on its title bar and dragging the window to a new location. You can change the size of a window by clicking on any corner and dragging the corner to a new location.

Custom Edit Fields in EViews

Most of your interactive work in EViews will consist of entering information in dialogs. Dialogs in EViews work much like dialogs in any other program, a window will open prompting you to manipulate controls and to enter information in edit fields.

EViews offers custom edit fields that provide smart auto-complete and automatic expansion, simplifying the task of entering the required information.

Smart Auto-Complete

Many dialogs in EViews require the user to enter the names of one or more objects in the workfile (see Chapter 4. "Object Basics," beginning on page 101 for a detailed description of objects). You may, for example be prompted for the name of explanatory variables to use in a regression, or for the name of a vector to use in initializing a procedure.

Smart auto-complete allows you to easily enter the required object names in edit fields, generally by bringing up a list of available objects workfile from which you may select. Where possible, EViews restricts the list of objects to include only those that make sense given the context of the edit field.

- If you press CTRL-L, EViews will display a list of objects for you to make a selection. The list display will centered on the closest match to what you have already typed.

- If you type CTRL-space, EViews will add a unique matching object name to the edit field or display the list.

By default, smart auto-complete will display the list so that the list pops up automatically after a delay in typing, but you may configure so that list is only displayed when you press CTRL-L (or CTRL-space when there is no unique match).

For example, when you specify an equation in EViews you will be prompted the names of variables (series objects) or groups of series (a group object) to be used in estimation in an edit field. In the example below, we started typing "MSFT", and EViews brought up a list of all series and groups in the workfile, and centered the list on the first object whose name start with "MSFT".
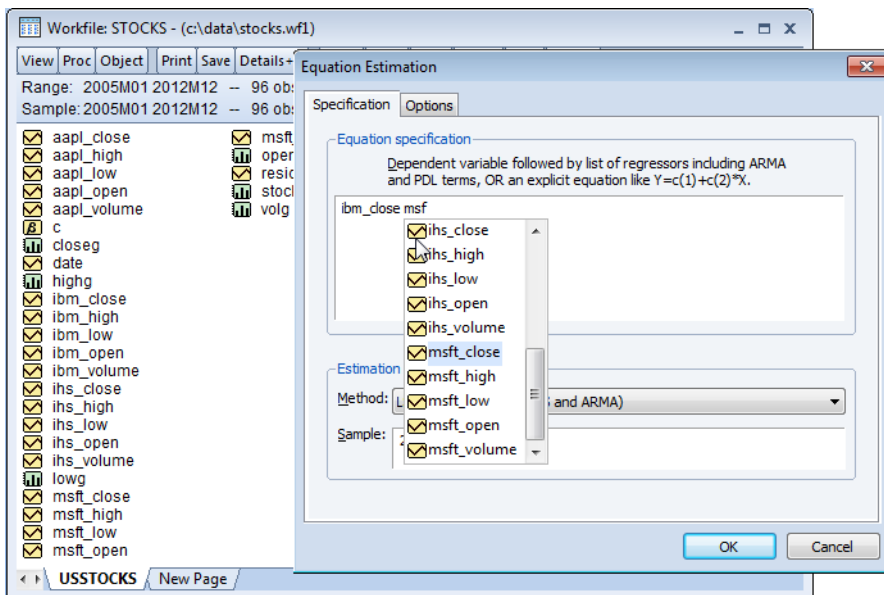
Continuing to type will change the list centering. You may also use the scroll bar to display a different portion of the list.

If EViews finds at least one object in the workfile that matches your text, the first such name will be selected, and the name will be shaded to indicate this selection. You may continue to type to refine the selection, or you may use the scroll bar and mouse to select an alternate object.

Single clicking on a name will display a description of the selected object, if one is available. Double-clicking, hitting ENTER, or pressing the TAB key will add the name of the selected object to the edit field. Pressing ESC closes the list and returns you to the dialog without altering the edit field.

If there are no objects that match your text, the list will be centered on the nearest match, but the object will not be selected. The lack of selection will be indicated by an open box around the name:
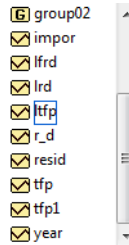
When there is no object selected, pressing TAB will add the name of the nearest match (indicated by the open box) to the edit field. Pressing ESC will close the list and return you to the dialog, without making additional changes. Hitting RETURN does not add the nearest match name to the edit field, but instead closes the list and enters a RETURN keystroke in the original edit field.

You may modify the automatic pop-up behavior of the smart auto-complete using the EViews global keyboard options. See "Keyboard," on page 862.

Expanded Edit fields

In more complex dialogs, size is at a premium so that individual edit fields may be too small to enter information without horizontal or vertical scrolling. Edit field expansion lets you increase the size of edit fields in EViews so that you may more easily see and enter information. To expand an edit field, simply double click on the white space in the box, or right click and select **Expand**.
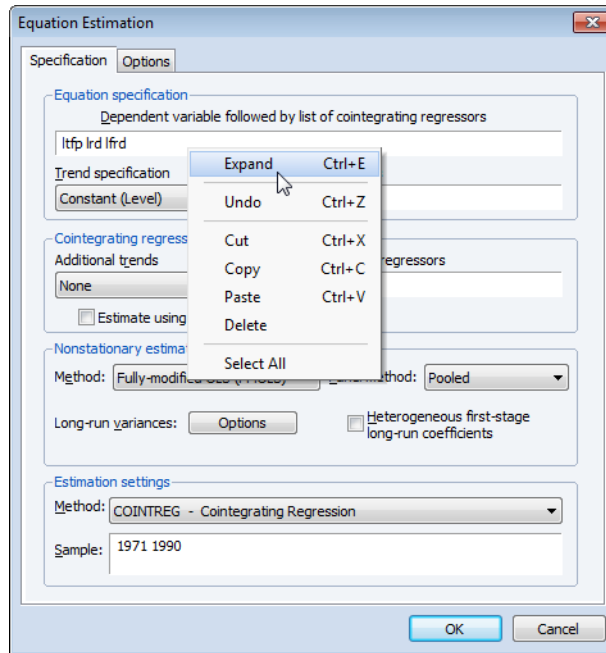
EViews will bring up a new edit box, filled with the contents of the original edit field. You may type, copy-and-paste and do all of the standard operations with this edit field. The edit field may also be resized by dragging the lower right-hand corner.
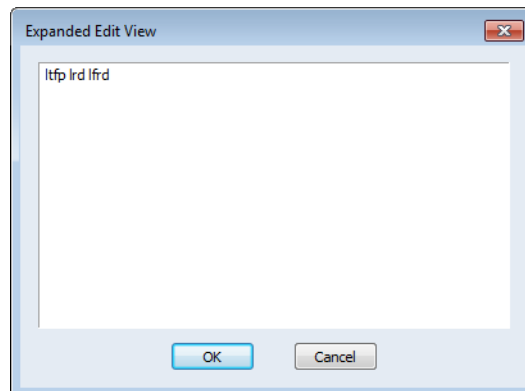
After you modify the edit field contents and click on **OK** EViews will copy your entered text into the original edit field and close the expanded edit field. If you click on **Cancel**, the box closes, but the edited contents are discarded. The contents of the original edit field will be retained.

Breaking or Canceling in EViews

EViews follows the Windows standard in using the ESC key as the break key. If you wish to cancel the current task or ongoing operation, simply press ESC.

Closing EViews

There are a number of ways to close EViews. You can always select **File/Exit** from the main menu, or you can press ALT-F4. Alternatively, you can click on the close box in the upper right-hand corner of the EViews window, or double click on the EViews icon in the upper left-hand corner of the window. If necessary, EViews will warn you and provide you with the opportunity to save any unsaved work.

Where to Go For Help

The EViews Manuals

This *User's Guide* describes how to use EViews to carry out your research. The earlier chapters deal with basic operations, the middle chapters cover basic econometric methods, and the later chapters describe more advanced methods.

Though we have tried to be complete, it is not possible to document every aspect of EViews. There are almost always several ways to do the same thing in EViews, and we cannot describe them all. In fact, one of the strengths of the program is that you will undoubtedly discover alternative, and perhaps more efficient, ways to get your work done.

Most of the *User's Guide* explains the visual approach to using EViews. It describes how you can use your mouse to perform operations in EViews. To keep the explanations simple, we do not tell you about alternative ways to get your work done. For example, we will not remind you about the ALT- keyboard alternatives to using the mouse.

When we get to the discussion of the substantive statistical methods available in EViews, we will provide some technical information about the methods, and references to econometrics textbooks and other sources for additional information.

The Help System

Almost all of the EViews documentation may be viewed from within EViews by using the help system. To access the EViews help system, simply go to the main menu and select **Help**.

In addition, the Help system will contain updates to the documentation that were made after the manuals went to press.

The EViews Forum

To supplement the information provided in the manuals and the help system, we encourage you to visit the EViews Forum where you can find answers to common questions about installing, using, and getting the most out of EViews. The Forum is an ideal place to ask questions of and share information with other EViews users. The forum address is:

> http://forums.eviews.com.

EViews Updates

IHS Global Inc. occasionally releases patches to the EViews program, and we strongly recommend that you keep your copy of EViews as up-to-date as possible.

EViews includes an automatic updating feature that can check for new updates every day, and install any update that may be available. The automatic update feature can be disabled from the **Options/EViews Auto-Update from Web** item in
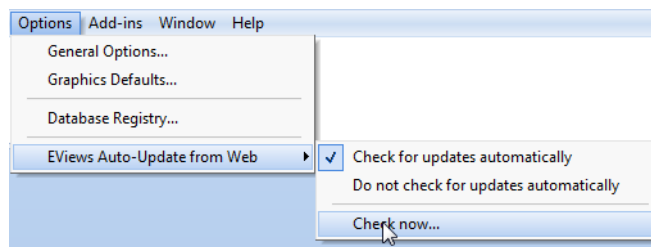
your main EViews menu. You can also manually check for updates from within EViews at any time by selecting **Check now...** under the **EViews Auto-Update from Web** menu item, or by selecting **EViews Update** from the **Help** menu.

You may also check for updates to the EViews program or for updates to supplementary components such as the EViews documentation or sample data and programs manually by visiting the Download section of the EViews website (www.eviews.com).

Chapter 2. A Demonstration

In this chapter, we provide a demonstration of some basic features of EViews. The demonstration is meant to be a brief introduction to EViews; not a comprehensive description of the program. A full description of the program begins in Chapter 4. "Object Basics," on page 101.

This demo takes you through the following steps:

- getting data into EViews from an Excel spreadsheet
- examining your data and performing simple statistical analyses
- using regression analysis to model and forecast a statistical relationship
- performing specification and hypothesis testing
- plotting results

Getting Data into EViews

The first step in most projects will be to read your data into an EViews workfile. EViews provides sophisticated tools for reading from a variety of common data formats, making it extremely easy to get started.

Before we describe the process of reading a foreign data file, note that the data for this demonstration have been included in both Excel spreadsheet and EViews workfile formats in your EViews installation directory (".\Example Files\EV8 Manual Data\Chapter 2 - A Demonstration").

If you wish to skip the discussion of opening foreign files, going directly to the analysis part of the demonstration, you may load the EViews workfile by selecting **File/Open/ EViews Workfile…** and opening "Demo.WF1", or by navigating to the directory and drag-and-dropping "Demo.WF1" onto your open EViews application window or icon.

The easiest way to read the data from Excel file "Demo.XLS" is to drag-and-drop the file into an open EViews application window. You may also drag-and-drop the file onto the EViews icon. In the latter case, Windows will first start the EViews application and will then open the demonstration Excel workfile.

Alternately, you may use the **File/Open/Foreign Data as Workfile…** dialog, selecting **Files of type** Excel and selecting the desired file.
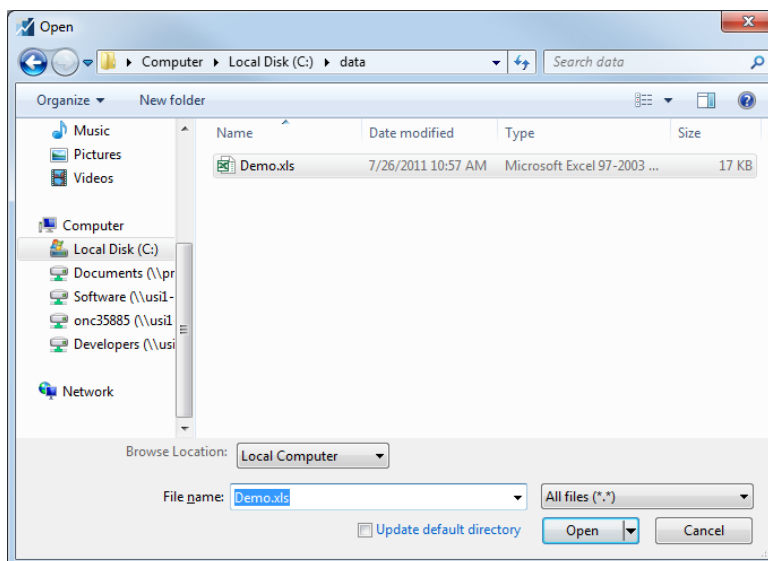
When EViews opens "Demo.XLS", it determines that the file is in Excel file format, analyzes the contents, and opens the **Excel Read** wizard.

The first page of the wizard includes a preview of the data found in the spreadsheet. In most cases, you need not worry about any of the options on this page. In more complicated cases, you may use the options on this page to provide a custom range of cells to read, or to select a different sheet in the workbook.
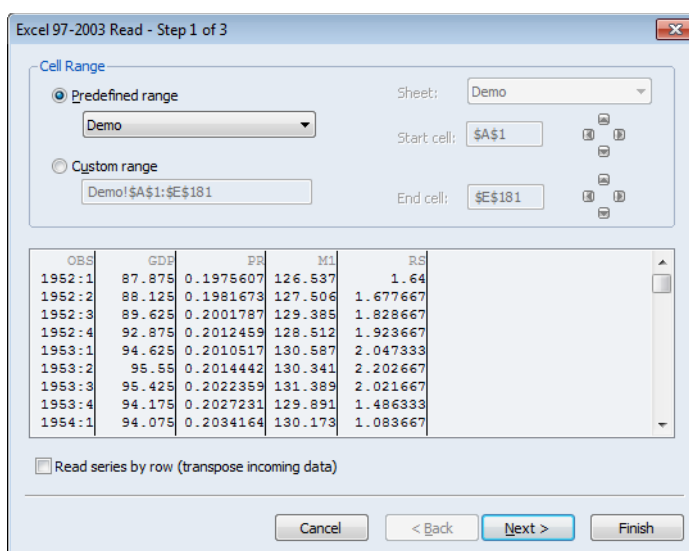
The second page of the wizard contains various options for reading the Excel data. These options are set at the most likely choices given the EViews analysis of the contents of your workbook. In most cases, you should simply click on **Finish** to accept the default settings. In other cases where the preview window

does not correctly display the desired data, you may click on **Next** and adjust the options that appear on the second page of the wizard. In our example, the data appear to be correct, so we simply click on **Finish** to accept the default settings.

When you accept the settings, EViews automatically creates a workfile that is sized to hold the data, and imports the series into the workfile. The workfile ranges from 1952 quarter 1 to 1996 quarter 4, and contains five series (GDP, M1, OBS, PR, and RS) that you have read from the Excel file. There are also two objects, the coefficient vector C and the series RESID, that are found in all EViews workfiles.
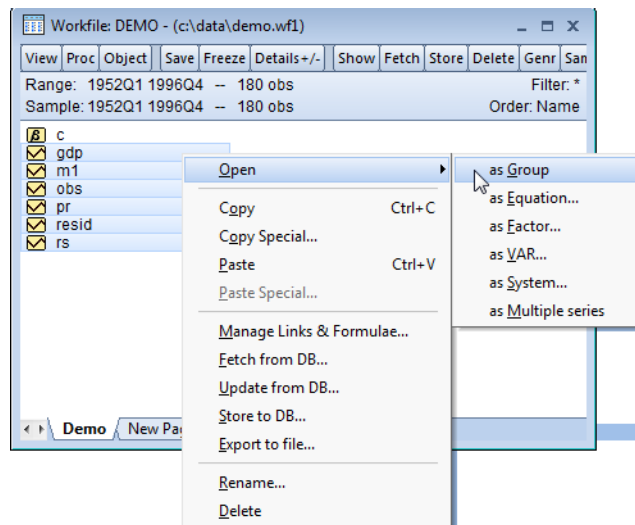
Select all of the series, right-click, and select **Open/as Group**. EViews will open the selected series in spreadsheet view.

You can use the scroll bars and scroll arrows on the right side of the window to view and verify the reminder of the data.
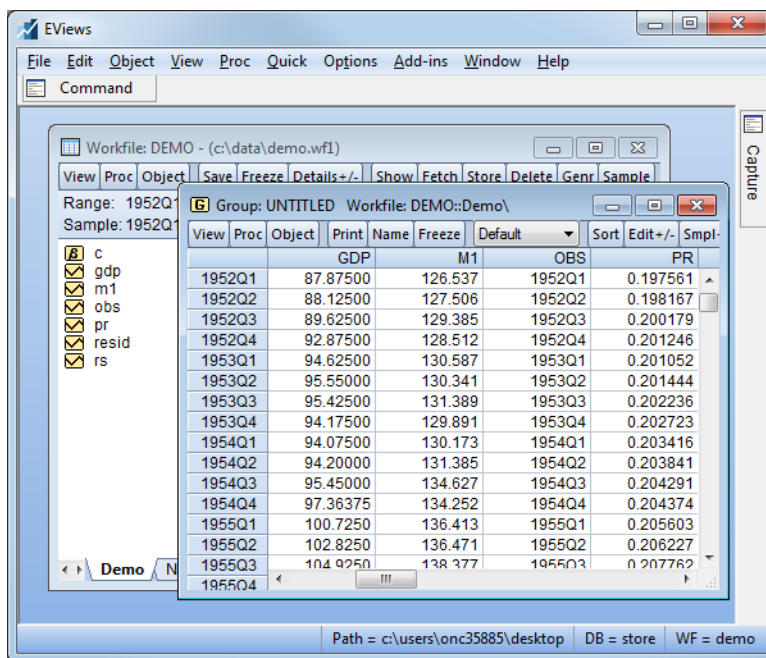
You may wish to click on the **Name** button in the group toolbar to provide a name for your UNTITLED group. Enter the name ORIGINAL, and click on **OK** to accept the name.
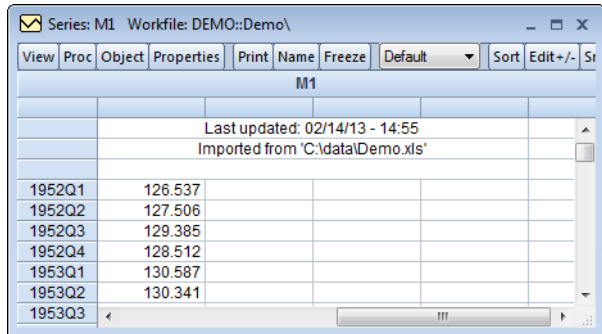
Once you are satisfied that the data are correct, you should save the workfile by clicking on the **Save** button in the workfile window. A saved dialog will open, prompting you for a workfile name and location. You should enter "Demo2.WF1", and then click **OK**. A second dialog may be displayed prompting you to set storage options. Click **OK** to accept the defaults. EViews will save the workfile in the specified directory with the name "Demo2.WF1". A saved workfile may be opened later by selecting **File/Open/Workfile….** from the main menu.

Examining the Data

Now that you have your data in an EViews workfile, you may use basic EViews tools to examine the data in your series and groups in a variety of ways.

First, we examine the character-istics of individual series. To see the contents of the M1 series, simply double click on the M1 icon in the workfile window, or select **Quick/Show…** in the main menu, enter m1, and click on **OK**.
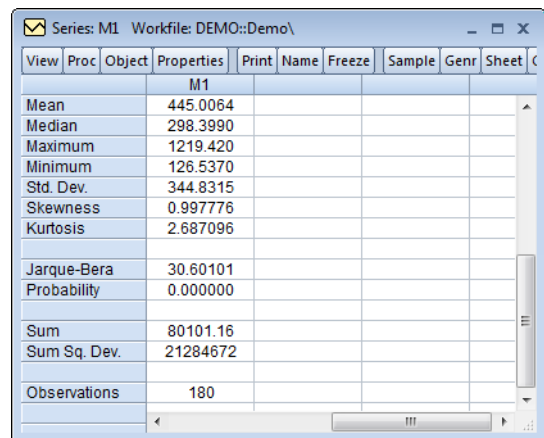
EViews will open the M1 series object and will display the default spreadsheet view of the series. Note the description of the contents of the series ("Series: M1") in the upper leftmost corner of the series window toolbar, indicating that you are working with the M1 series.

You will use the entries in the **View** and **Proc** menus to examine various characteristics of the series. Simply click on the buttons on the toolbar to access these menu entries, or equiva-lently, select **View** or **Proc** from the main menu.

To compute, for example, a table of basic descriptive statistics for M1, simply click on the **View** button, then select **Descriptive Statistics & Tests/ Stats Table**. EViews will compute descriptive statistics for M1 and change the series view to display a table of results.

Similarly, to examine a line graph of the series, simply select **View/Graph…** to bring up the **Graph Options** dialog, and select **Line & Symbol** from the list of graph types on the left-hand side. EViews will change the M1 series window to display a line graph of the data in the M1 series.
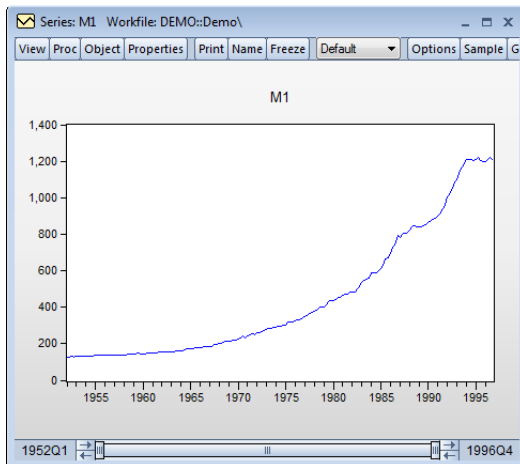
Note the presence of the slider bar in the bottom of the window which you may drag to show a sub-sample of the observations.

At this point, you may wish to explore the contents of the **View** and **Proc** menus in the M1 series window to see the various tools for examining and working with series data. You may always return to the spreadsheet view of your series by selecting **View/Spreadsheet** from the toolbar or main menu.

Since our ultimate goal is to perform regression analysis with our data expressed in natural logarithms, we may instead wish to work with the log of M1. Fortunately, EViews allows you to work with expressions involving series as easily as you work with the series themselves. To open a series containing this expression, select **Quick/Show…** from the main menu, enter the text for the expression, `log(m1)`, and click **OK**. EViews will open a series window for containing LOG(M1). Note that the titlebar for the series shows that we are working with the desired expression.

You may work with this auto-series in exactly the same way you worked with M1 above. For example, clicking on **View** in the series toolbar and selecting **Descriptive Statistics & Tests/ Histogram and Stats** displays a view containing a histogram and descriptive statistics for LOG(M1):
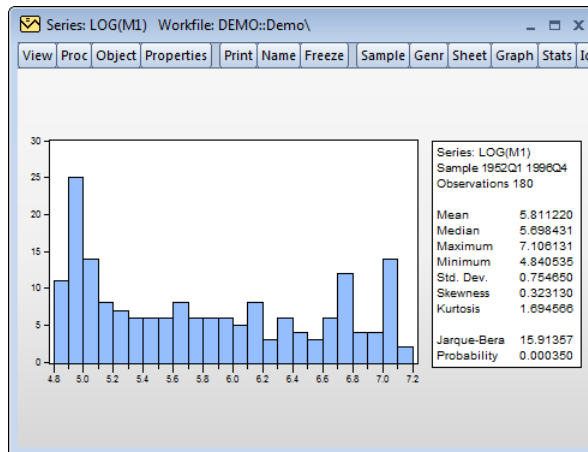
Alternately, we may display a smoothed version of the histogram by selecting **View/ Graph…**, choosing **Distribution** from the list on the left and **Kernel Density** from the drop-down on the right, and clicking on **OK** to accept the default options:
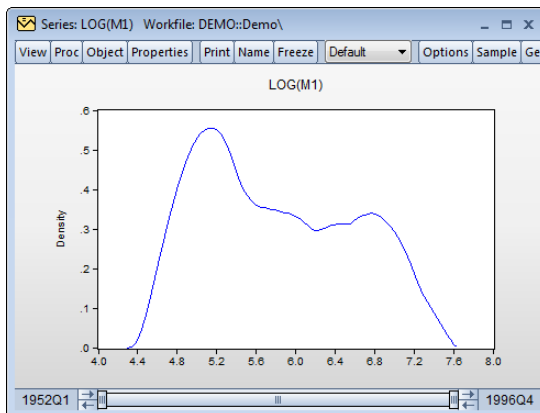
Suppose that you wish to examine multiple series or series expressions. To do so, you will need to construct a group object that contains the series of interest.

Earlier, you worked with an EViews created group object containing all of the series read from your Excel file. Here, we will construct a group object containing expressions involving a subset of those series. We wish to create a group object containing the logarithms of the series M1 and GDP, the level of RS, and the first difference of the logarithm of the series PR. Simply select **Quick/Show...** from the main EViews menu, and enter the list of expressions and series names:

```
log(m1) log(gdp) rs dlog(pr)
```

Click on **OK** to accept the input. EViews will open a group window containing a spreadsheet view of the series and expressions of interest.

As with the series object, you will use the **View** and **Proc** menus of the group to examine various characteristics of the group of series. Simply click on the buttons on the toolbar to access these menu entries or select **View** or **Proc** from the main menu to call up the relevant entries. Note that the entries for a group object will differ from those for a series object since the kinds of operations you may perform with multiple series differ from the types of operations available when working with a single series.

For example, you may select **View/Graph...** from the group object toolbar, and then select **Line & Symbol** from the list on the left side of the dialog to display a single graph containing line plots of each of the series in the group:
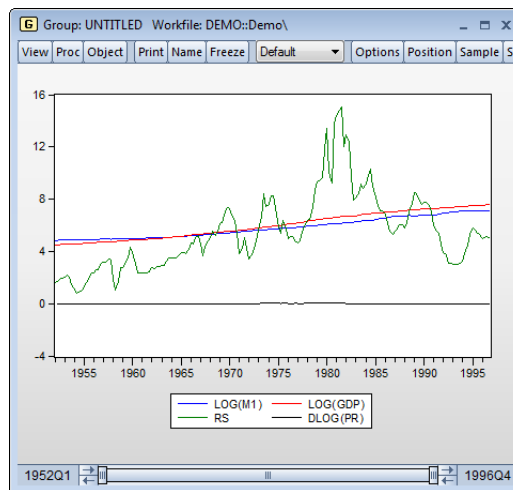
Alternately, you may select **View/Graph...** and choose **Multiple graphs** from the Multiple series drop-down on the right side of the dialog to display the same information, but with each series expression plotted in an individual graph:
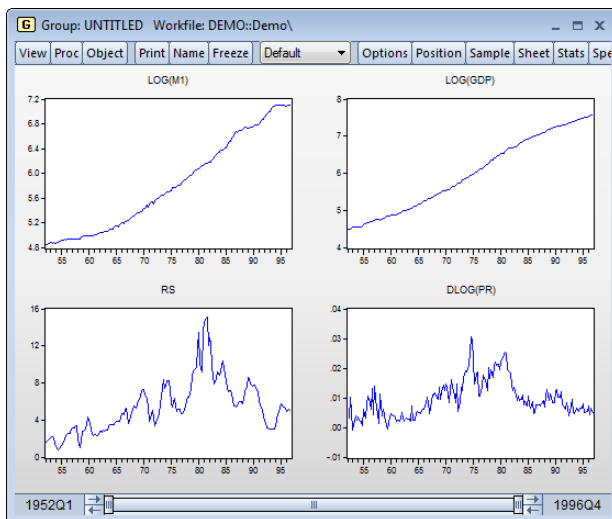
Likewise, you may select **View/Descriptive Stats/Individual Samples** to display a table of descriptive statistics computed for each of the series in the group:

Note that the number of observations used for computing descriptive statistics for DLOG(PR) is one less than the number used to compute the statistics for the other expressions. By electing to compute our statistics using "Individual Samples", we informed EViews that we wished to use the series specific samples in each computation, so that the loss of an observation in DLOG(PR) to differencing should not affect the samples used in calculations for the remaining expressions.

We may instead choose to use "Common Samples" so that observations are only used if the data are available for all of the series in the group. Click on **View/Covariance Analysis...**

and select only the **Correlation** checkbox to display the correlation matrix of the four series for the 179 common observations:

	LOG(M1)	LOG(GDP)	RS	DLOG(PR)
LOG(M1)	1.000000	0.992173	0.469921	0.191597
LOG(GDP)	0.992173	1.000000	0.554978	0.267920
RS	0.469921	0.554978	1.000000	0.681349
DLOG(PR)	0.191597	0.267920	0.681349	1.000000

Once again, we suggest that you may wish to explore the contents of the **View** and **Proc** menus for this group to see the various tools for examining and working with sets of series You can always return to the spreadsheet view of the group by selecting **View/Spreadsheet**.

Estimating a Regression Model

We now estimate a regression model for M1 using data over the period from 1952Q1–1992Q4 and use this estimated regression to construct forecasts over the period 1993Q1–2003Q4. The model specification is given by:

$$\log(M1_t) = \beta_1 + \beta_2\log(\text{GDP}_t) + \beta_3\text{RS}_t + \beta_4\Delta\log(\text{PR}_t) + \epsilon_t \tag{2.1}$$

where log(M1) is the logarithm of the money supply, log(GDP) is the log of income, RS is the short term interest rate, and $\Delta\log(\text{PR})$ is the log first difference of the price level (the approximate rate of inflation).

To estimate the model, we will create an equation object. Select **Quick** from the main menu and choose **Estimate Equation…** to open the estimation dialog. Enter the following equation specification:
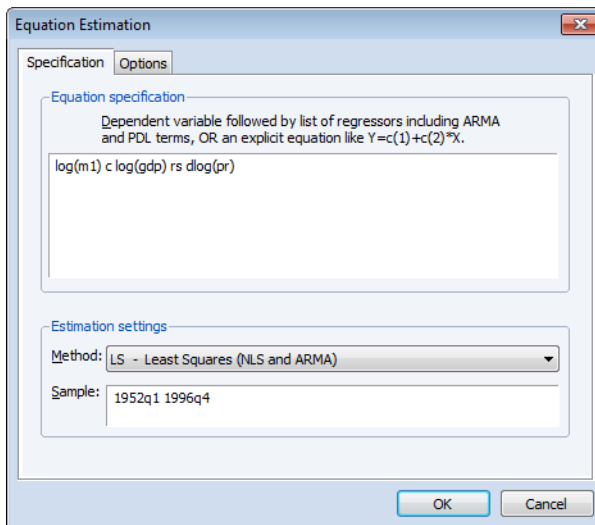
Here we list the expression for the dependent variable, followed by the expressions for each of the regressors, separated by spaces. The built-in series name C stands for the constant in the regression.

The dialog is initialized to estimate the equation using the **LS - Least Squares** method for the sample 1952Q1 1996Q4. You should change text in the **Sample** edit box to "1952Q1 1992Q4" or equivalently "1952 1992" to estimate the equation for the subsample of observations.

Click **OK** to estimate the equation using least squares and to display the regression results:

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 08/18/09   Time: 14:14
Sample (adjusted): 1952Q2 1992Q4
Included observations: 163 after adjustments
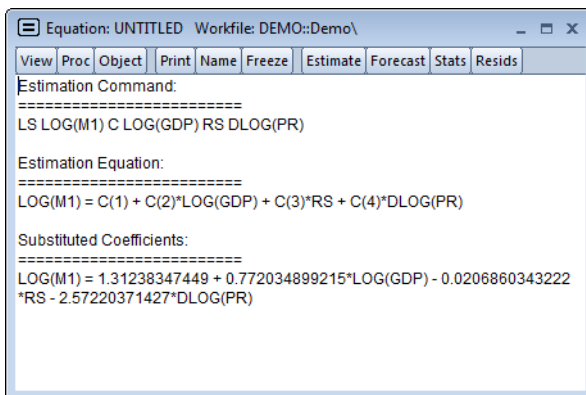
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.312383	0.032199	40.75850	0.0000
LOG(GDP)	0.772035	0.006537	118.1092	0.0000
RS	-0.020686	0.002516	-8.221196	0.0000
DLOG(PR)	-2.572204	0.942556	-2.728967	0.0071

R-squared	0.993274	Mean dependent var	5.692279
Adjusted R-squared	0.993147	S.D. dependent var	0.670253
S.E. of regression	0.055485	Akaike info criterion	-2.921176
Sum squared resid	0.489494	Schwarz criterion	-2.845256
Log likelihood	242.0759	Hannan-Quinn criter.	-2.890354
F-statistic	7826.904	Durbin-Watson stat	0.140967
Prob(F-statistic)	0.000000		

Note that the equation is estimated from 1952Q2 to 1992Q4 since one observation is dropped from the beginning of the estimation sample to account for the DLOG difference term. The estimated coefficients are statistically significant, with t-statistic values well in excess of 2. The overall regression fit, as measured by the R^2 value, indicates a very tight fit. You can select **View/Actual, Fitted, Residual/Actual, Fitted, Residual Graph** in the equation toolbar to display a graph of the actual and fitted values for the dependent variable, along with the residuals:
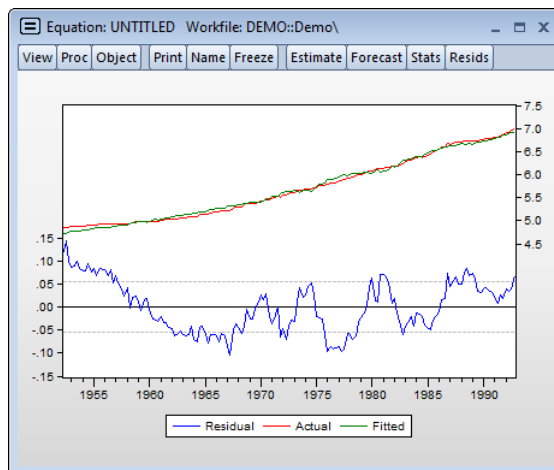
Specification and Hypothesis Tests

We can use the estimated equation to perform hypothesis tests on the coefficients of the model. For example, to test the hypothesis that the coefficient on the price term is equal to 2, we will perform a Wald test. First, determine the coefficient of interest by selecting **View/ Representations** from the equation toolbar:

```
Equation: UNTITLED   Workfile: DEMO::Demo\                    _ □ X
View Proc Object  Print Name Freeze  Estimate Forecast Stats Resids
Estimation Command:
========================
LS LOG(M1) C LOG(GDP) RS DLOG(PR)

Estimation Equation:
========================
LOG(M1) = C(1) + C(2)*LOG(GDP) + C(3)*RS + C(4)*DLOG(PR)

Substituted Coefficients:
========================
LOG(M1) = 1.31238347449 + 0.772034899215*LOG(GDP) - 0.0206860343222
*RS - 2.57220371427*DLOG(PR)
```

Note that the coefficients are assigned in the order that the variables appear in the specification so that the coefficient for the PR term is labeled C(4). To test the restriction on C(4) you should select **View/Coefficient Diagnostics/Wald Test–Coefficient Restrictions…**, and enter the restriction "c(4) = 2". EViews will report the results of the Wald test:

Wald Test:
Equation: UNITITLED
Null Hypothesis: C(4)=2

Test Statistic	Value	df	Probability
t-statistic	-4.850856	159	0.0000
F-statistic	23.53081	(1, 159)	0.0000
Chi-square	23.53081	1	0.0000

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
-2 + C(4)	-4.572204	0.942556

Restrictions are linear in coefficients.

The low probability values indicate that the null hypothesis that C(4) = 2 is strongly rejected.

We should, however, be somewhat cautious of accepting this result without additional analysis. The low value of the Durbin-Watson statistic reported above is indicative of the pres-

ence of serial correlation in the residuals of the estimated equation. If uncorrected, serial correlation in the residuals will lead to incorrect estimates of the standard errors, and invalid statistical inference for the coefficients of the equation.

The Durbin-Watson statistic can be difficult to interpret. To perform a more general Breusch-Godfrey test for serial correlation in the residuals, select **View/Residual Diagnostics/Serial Correlation LM Test…** from the equation toolbar, and specify an order of serial correlation to test against. Entering "1" yields a test against first-order serial correlation:

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	813.0060	Prob. F(1,158)	0.0000
Obs*R-squared	136.4770	Prob. Chi-Square(1)	0.0000

Test Equation:
Dependent Variable: RESID
Method: Least Squares
Date: 08/18/09   Time: 14:10
Sample: 1952Q2 1992Q4
Included observations: 163
Presample missing value lagged residuals set to zero.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.006355	0.013031	-0.487683	0.6265
LOG(GDP)	0.000997	0.002645	0.376929	0.7067
RS	-0.000567	0.001018	-0.556748	0.5785
DLOG(PR)	0.404143	0.381676	1.058864	0.2913
RESID(-1)	0.920306	0.032276	28.51326	0.0000

R-squared	0.837282	Mean dependent var	1.23E-15
Adjusted R-squared	0.833163	S.D. dependent var	0.054969
S.E. of regression	0.022452	Akaike info criterion	-4.724644
Sum squared resid	0.079649	Schwarz criterion	-4.629744
Log likelihood	390.0585	Hannan-Quinn criter.	-4.686116
F-statistic	203.2515	Durbin-Watson stat	1.770965
Prob(F-statistic)	0.000000		

The top part of the output presents the test statistics and associated probability values. The test regression used to carry out the test is reported below the statistics.

The statistic labeled "Obs*R-squared" is the LM test statistic for the null hypothesis of no serial correlation. The (effectively) zero probability value strongly indicates the presence of serial correlation in the residuals.

Modifying the Equation

The test results suggest that we need to modify our original specification to take account of the serial correlation.

One approach is to include lags of the independent variables. To add variables to the existing equation, click on the **Estimate** button in the equation toolbar and edit the specification to include lags for each of the original explanatory variables:

```
log(m1) c log(gdp) rs dlog(pr) log(m1(-1)) log(gdp(-1)) rs(-1)
    dlog(pr(-1))
```

Note that lags are specified by including a negative number, enclosed in parentheses, following the series name. Click on **OK** to estimate the new specification and to display the results:

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 08/18/09   Time: 14:19
Sample (adjusted): 1952Q3 1992Q4
Included observations: 162 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.071297	0.028248	2.523949	0.0126
LOG(GDP)	0.320338	0.118186	2.710453	0.0075
RS	-0.005222	0.001469	-3.554801	0.0005
DLOG(PR)	0.038615	0.341619	0.113036	0.9101
LOG(M1(-1))	0.926640	0.020319	45.60375	0.0000
LOG(GDP(-1))	-0.257364	0.123264	-2.087910	0.0385
RS(-1)	0.002604	0.001574	1.654429	0.1001
DLOG(PR(-1))	-0.071650	0.347403	-0.206246	0.8369

R-squared	0.999604	Mean dependent var	5.697490
Adjusted R-squared	0.999586	S.D. dependent var	0.669011
S.E. of regression	0.013611	Akaike info criterion	-5.707729
Sum squared resid	0.028531	Schwarz criterion	-5.555255
Log likelihood	470.3261	Hannan-Quinn criter.	-5.645823
F-statistic	55543.30	Durbin-Watson stat	2.393764
Prob(F-statistic)	0.000000		

Note that EViews has automatically adjusted the estimation sample to accommodate the additional lagged variables. We will save this equation in the workfile for later use. Press the **Name** button in the toolbar and name the equation EQLAGS.
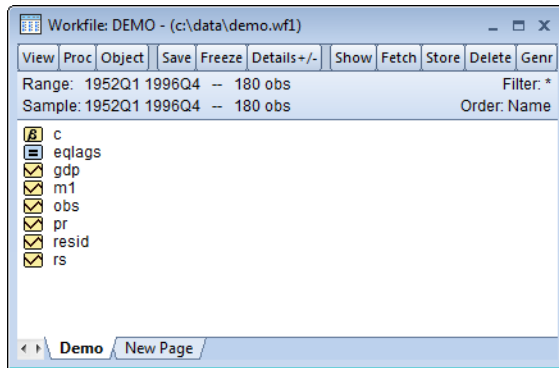
The EQLAGS equation object will be placed in the workfile.

One common method of accounting for serial correlation is to include autoregressive (AR) and/or moving average (MA) terms in the equation. To estimate the model with an AR(1) error specification, you should make a copy of the EQLAGS equation by clicking **Object/ Copy Object…** in the EQLAGS window. EViews will create a new untitled equation containing all of the information from the previous equation. Press **Estimate** on the toolbar of the copy and modify the specification to read

```
log(m1) c log(gdp) rs dlog(pr) ar(1)
```

This specification removes the lagged terms, replacing them with an AR(1) specification:

$$\log(M1_t) = \beta_1 + \beta_2 \log(\text{GDP}_t) + \beta_3 \text{RS}_t + \beta_4 \Delta \log(\text{PR}_t) + u_t$$
$$u_t = \rho\, u_{t-1} + \epsilon_t$$

$$(2.2)$$

Click **OK** to accept the new specification. EViews will estimate the equation and will report the estimation results, including the estimated first-order autoregressive coefficient of the error term:

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 08/18/09   Time: 14:12
Sample (adjusted): 1952Q3 1992Q4
Included observations: 162 after adjustments
Convergence achieved after 32 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.050341	0.328260	3.199724	0.0017
LOG(GDP)	0.794928	0.049324	16.11633	0.0000
RS	-0.007395	0.001457	-5.075118	0.0000
DLOG(PR)	-0.008020	0.348689	-0.023002	0.9817
AR(1)	0.968109	0.018189	53.22359	0.0000

R-squared	0.999526	Mean dependent var		5.697490
Adjusted R-squared	0.999514	S.D. dependent var		0.669011
S.E. of regression	0.014751	Akaike info criterion		-5.564584
Sum squared resid	0.034164	Schwarz criterion		-5.469288
Log likelihood	455.7313	Hannan-Quinn criter.		-5.525892
F-statistic	82748.93	Durbin-Watson stat		2.164283
Prob(F-statistic)	0.000000			

Inverted AR Roots	.97	

The fit of the AR(1) model is roughly comparable to the lag model, but its somewhat higher values for both the Akaike and the Schwarz information criteria indicate that the previous lag model may be preferred. Accordingly, we will work with the lag model in EQLAGS for the remainder of the demonstration.

Forecasting from an Estimated Equation

We have been working with a subset of our data, so that we may compare forecasts based upon this model with the actual data for the post-estimation sample 1993Q1–1996Q4.

Click on the **Forecast** button in the EQLAGS equation toolbar to open the forecast dialog:

We set the forecast sample to 1993Q1–1996Q4 and provide names for both the forecasts and forecast standard errors so both will be saved as series in the workfile. The forecasted values will be saved in M1_F and the forecast standard errors will be saved in M1_SE.

Note also that we have elected to forecast the log of M1, not the level, and that we request both graphical and forecast evaluation output. The **Dynamic** option constructs the forecast for the sample period using only information available at the beginning of 1993Q1. When you click **OK**, EViews displays both a graph of the forecasts, and statistics evaluating the quality of the fit to the actual data:
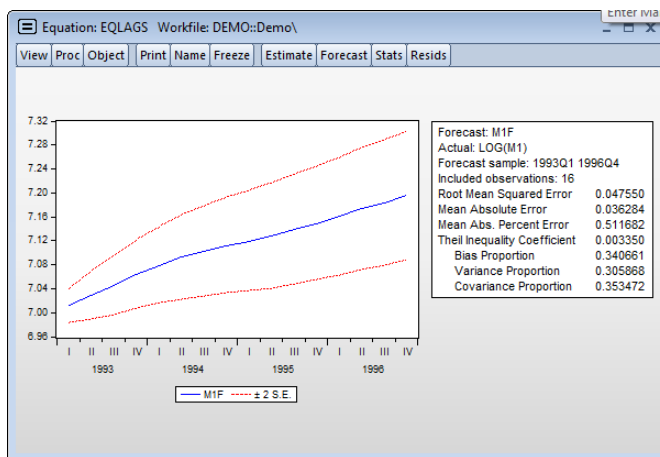
Alternately, we may also choose to examine forecasts of the level of M1. Click on the **Forecast** button in the EQLAGS toolbar to open the forecast dialog, and select **M1** under the

Series to forecast option. Enter a new name to hold the forecasts and standard errors, say M1LEVEL_F and M1LEVEL_SE, and click **OK**.
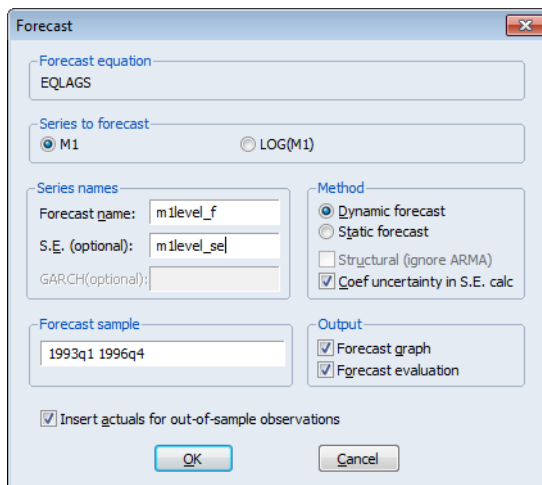
EViews will present a graph of the forecast of the level of M1, along with the asymmetric confidence intervals for this forecast:
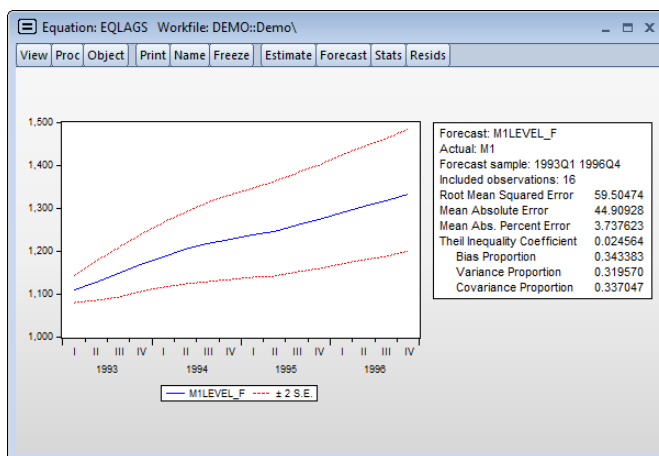
The series that the forecast procedure generates are ordinary EViews series that you may work with in the usual ways. For example, we may use the forecasted series for LOG(M1) and the standard errors of the forecast to plot actuals against forecasted values with (approximate) 95% confidence intervals for the forecasts.

We will first create a new group object containing these values. Select **Quick/Show...** from the main menu, and enter the expressions:

```
m1_f+2*m1_se m1_f-2*m1_se log(m1)
```

to create a group containing the confidence intervals for the forecast of LOG(M1) and the actual values of LOG(M1):
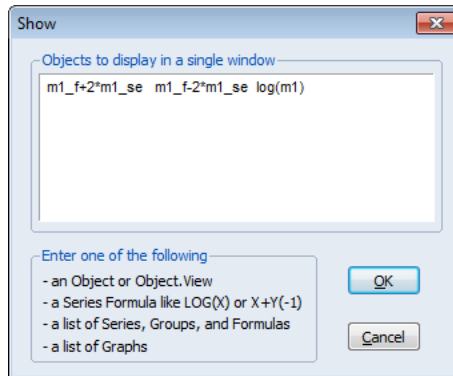
There are three expressions in the dialog. The first two represent the upper and lower bounds of the (approximate) 95% forecast interval as computed by evaluating the values of the point forecasts plus and minus two times the standard errors. The last expression represents the actual values of the dependent variable.

When you click **OK**, EViews opens an untitled group window containing a spreadsheet view of the data. Before plotting the data, we will change the sample of observations so that we only plot data for the forecast sample. Select **Quick/Sample…** or click on the **Sample** button in the group toolbar, and change the sample to include only the forecast period:
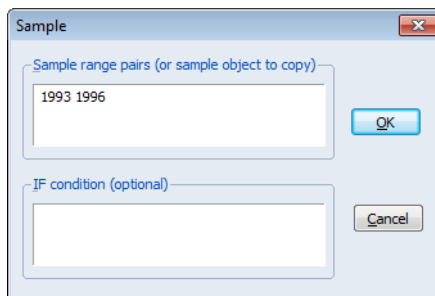
To plot the data for the forecast period, select **View/Graph…** from the group window and choose **Line & Symbol** from the list on the left of the **Graph Options** dialog:
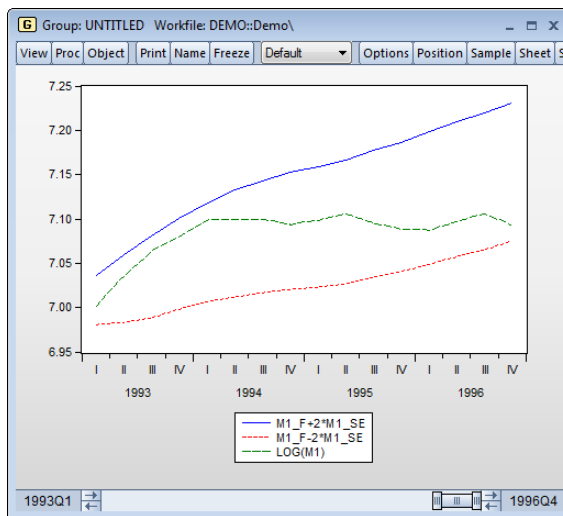
The actual values of log(M1) are within the forecast interval for most of the forecast period, but fall below the lower bound of the 95% confidence interval beginning in 1996:1.

(Note that the slider bar at the bottom of the graph indicates that we are viewing only a subset of the workfile range).

For an alternate view of these data, you can select **View/Graph... and Error Bar from the list in the dialog**, which displays the graph as follows:
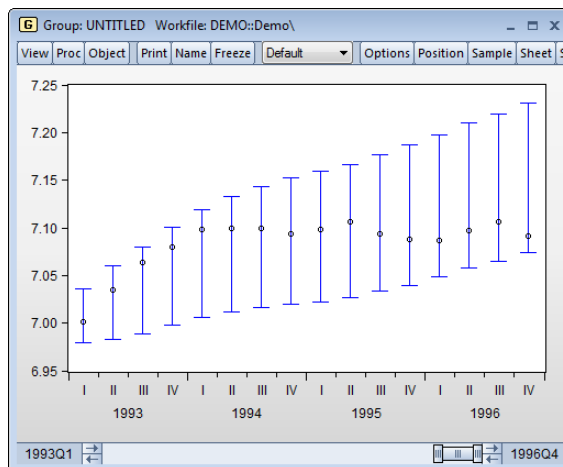
This graph shows clearly that the forecasts of LOG(M1) over-predict the actual values in the last four quarters of the forecast period.

Additional Testing

Note that the above specification has been selected for illustration purposes only. Indeed, performing various specification tests on EQLAGS suggests that there may be a number of problems with the existing specification.

For one, there is quite a bit of serial correlation remaining even after estimating the lag specification. A test of serial correlation in the EQLAGS equation (by selecting **View/Residual Diagnostics/Serial Correlation LM Test…**, and entering "1" for the number of lags) rejects the null hypothesis of no serial correlation in the reformulated equation. The top portion of the output is:

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	7.880369	Prob. F(1,153)	0.0056
Obs*R-squared	7.935212	Prob. Chi-Square(1)	0.0048

Moreover, there is strong evidence of autoregressive conditional heteroskedasticity (ARCH) in the residuals. Select **View/Residual Diagnostics/Heteroskedasticity Tests…** to bring up the **Heteroskedasticity Tests** dialog. Then select **ARCH** from the listbox and accept the default of "1". The top portion of the ARCH test results shows that the data suggest the presence of ARCH in the residuals:

Heteroskedasticity Test: ARCH

F-statistic	11.21965	Prob. F(1,159)	0.0010
Obs*R-squared	10.61196	Prob. Chi-Square(1)	0.0011

In addition to serial correlation and ARCH, there is an even more fundamental problem with the above specification since, as the graphs attest, LOG(M1) exhibits a pronounced upward trend, suggesting that we should perform a unit root in this series. The presence of a unit root will indicate the need for further analysis.

We once again display the LOG(M1) series window by clicking on **Window** and selecting the LOG(M1) series window from the menu. If the series window for LOG(M1) is not present (if you previously closed the window), you may again open a new window by selecting **Quick/Show…**, entering "log(m1)", and clicking **OK**.

Before computing the test statistic, we will reset the workfile sample to all of the observations by clicking on Quick/Sample... and entering "@all" in the dialog.

Next, to perform an Augmented Dickey-Fuller (ADF) test for nonstationarity of this series, select **View/Unit Root Test…** and click on **OK** to accept the default options. EViews will perform an ADF test and display the test results. The top portion of the output reads:

Null Hypothesis: LOG(M1) has a unit root
Exogenous: Constant
Lag Length: 0 (Automatic - based on SIC, maxlag=13)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		3.797182	1.0000
Test critical values:	1% level	-3.470679	
	5% level	-2.879155	
	10% level	-2.576241	

*MacKinnon (1996) one-sided p-values.

EViews performs the ADF test statistic with the number of lagged difference terms in the test equation (here, four) determined by automatic selection. The ADF test statistic value has a probability value of 0.9911, providing little evidence that we may reject the null hypothesis of a unit root.

If a unit root were present in our data, we may wish to adopt more sophisticated statistical models. These techniques are discussed in Chapter 22. "Time Series Regression", Chapter 26. "Cointegrating Regression", and Chapter 40. "Vector Autoregression and Error Correction Models" (all in *User's Guide II*) which deal with basic time series, cointegrating regression models, and vector error correction specifications, respectively).

Chapter 3. Workfile Basics

Managing the variety of tasks associated with your work can be a complex and time-consuming process. Fortunately, EViews' innovative design takes much of the effort out of organizing your work, allowing you to concentrate on the substance of your project. EViews provides sophisticated features that allow you to work with various types of data in an intuitive and convenient fashion.

Before describing these features, we begin by outlining the basic concepts underlying the EViews approach to working with datasets using workfiles, and describing simple methods to get you started on creating and working with workfiles in EViews.

What is a Workfile?

At a basic level, a *workfile* is simply a container for EViews objects (see Chapter 4. "Object Basics," on page 101). Most of your work in EViews will involve objects that are contained in a workfile, so your first step in any project will be to create a new workfile or to load an existing workfile into memory.

Every workfile contains one or more *workfile pages*, each with its own objects. A workfile page may be thought of as a subworkfile or subdirectory that allows you to organize the data within the workfile.

For most purposes, you may treat a workfile page as though it were a workfile (just as a subdirectory is also a directory) since there is often no practical distinction between the two. Indeed, in the most common setting where a workfile contains only a single page, the two are completely synonymous. Where there is no possibility of confusion, we will use the terms "workfile" and "workfile page" interchangeably.

Workfiles and Datasets

While workfiles and workfile pages are designed to hold a variety of EViews objects, such as equations, graphs, and matrices, their primary purpose is to hold the contents of *datasets*. A dataset is defined here as a data rectangle, consisting of a set of *observations* on one or more *variables*—for example, a time series of observations on the variables GDP, investment, and interest rates, or perhaps a random sample of observations containing individual incomes and tax liabilities.

Key to the notion of a dataset is the idea that each observation in the dataset has a unique *identifier*, or *ID*. Identifiers usually contain important information about the observation, such as a date, a name, or perhaps an identifying code. For example, annual time series data typically use year identifiers ("1990", "1991", ...), while cross-

sectional state data generally use state names or abbreviations ("AL", "AK", ..., "WY"). More complicated identifiers are associated with longitudinal data, where one typically uses *both* an individual ID and a date ID to identify each observation.

Observation IDs are often, but not always, included as a part of the dataset. Annual datasets, for example, usually include a variable containing the year associated with each observation. Similarly, large cross-sectional survey data typically include an interview number used to identify individuals.

In other cases, observation IDs are not provided in the dataset, but external information is available. You may know, for example, that the 21 otherwise unidentified observations in a dataset are for consecutive years beginning in 1990 and continuing to 2010.

In the rare case were there is no additional identifying information, one may simply use a set of default integer identifiers that enumerate the observations in the dataset ("1", "2", "3", ...).

Since the primary purpose of every workfile page is to hold the contents of a single dataset, each page must contain information about observation identifiers. Once identifier information is provided, the workfile page provides context for working with observations in the associated dataset, allowing you to use dates, handle lags, or work with longitudinal data structures.

Creating a Workfile

There are several ways to create and set up a new workfile. The first task you will face in setting up a workfile (or workfile page) is to specify the structure of your workfile. We focus here on three distinct approaches:

First, you may simply *describe the structure of your workfile* ("Creating a Workfile by Describing its Structure" on page 43). EViews will create a new workfile for you to enter or import your data).

Describing the workfile is the simplest method, requiring only that you answer a few simple questions—it works best when the identifiers follow a simple pattern that is easily described (for example, "annual data from 1950 to 2000" or "quarterly data from 1970Q1 to 2002Q4"). This approach must be employed if you plan to enter data into EViews by typing or copy-and-pasting data.

In the second approach, you simply *open and read data from a foreign data source* ("Creating a Workfile by Reading from a Foreign Data Source" on page 47). EViews will analyze the data source, create a workfile, and then automatically import your data.

The final approach, which should be reserved for more complex settings, involves two distinct steps. In the first, you create a new workfile using one of the first two approaches (by

describing the structure of the workfile, or by opening and reading from a foreign data source). Next, you will s*tructure the workfile*, by instructing EViews to construct unique identifiers, in some cases using values of variables contained in the dataset.

We begin by describing the first two methods. The third approach, involving the more complicated task of structuring a workfile, will be taken up in "Structuring a Workfile" on page 263.

Creating a Workfile by Describing its Structure

To describe the structure of your workfile, you will need to provide EViews with external information about your observations and their associated identifiers. For example, you might tell EViews that your dataset consists of a time series of observations for each quarter from 1990Q1 to 2003Q4, or that you have information for every day from the beginning of 1997 to the end of 2001. At the very least, you might indicate that you have a dataset with 500 observations and no additional identifier information.

To create a new workfile, select **File/New/Workfile...** from the main menu to open the **Workfile Create** dialog.

On the left side of the dialog is a dropdown menu for describing the underlying structure of your dataset. You will choose between the **Dated - regular frequency**, the **Unstructured**, and the **Balanced Panel** settings. Generally speaking, you should use **Dated - regular frequency** if you have a simple time series dataset, for a simple panel dataset you should use **Balanced Panel**, and in all other cases, you should select **Unstructured**. Additional detail to aid you in making a selection is provided in the description of each category.

Describing a Dated Regular Frequency Workfile

When you select **Dated - regular frequency**, EViews will prompt you to select a frequency for your data. You may choose between the standard EViews supported date frequencies (**Multi-year, Annual**, **Semi-annual**, **Quarterly**, **Monthly**, **Bimonthly, Fortnight, Ten-day, Weekly**, **Daily - 5 day week**, **Daily - 7 day week, Daily - custom week, Intraday**), and a special frequency (**Integer date**) which is a generalization of a simple enumeration.

In selecting a frequency, you set intervals between observations in your data (whether they are annual, semi-annual, quarterly, monthly, weekly, etc.), which allows EViews to use all available calendar information to organize and manage your data. For example, when moving between daily and weekly or annual data, EViews knows that some years contain days in each of 53 weeks, and that some years have 366 days, and will use this information when working with your data.

```
Multi-year
Annual
Semi-annual
Quarterly
Monthly
Bimonthly
Fortnight
Ten-day (Trimonthly)
Weekly
Daily - 5 day week
Daily - 7 day week
Daily - custom week
Intraday
Integer date
```

As the name suggests, *regular frequency* data arrive at regular intervals, defined by the specified frequency (*e.g.*, monthly). In contrast, *irregular frequency* data do not arrive in regular intervals. An important example of irregular data is found in stock and bond prices where the presence of holidays and other market closures ensures that data are observed only irregularly, and not in a regular 5-day daily frequency. Standard macroeconomic data such as quarterly GDP or monthly housing starts are examples of regular data.

EViews also prompts you to enter a **Start date** and **End date** for your workfile. For **Daily - custom week** or **Intraday**, you will also need to enter **Day** and/or **Time** information as described below. When you click on **OK**, EViews will create a regular frequency workfile with the specified number of observations and the associated identifiers.

Suppose, for example, that you wish to create a quarterly workfile that begins with the first quarter of 1970 and ends in the last quarter of 2020.

- First, select **Dated - regular frequency** for the workfile structure, and then choose the **Quarterly** frequency.

- Next, enter the **Start date** and **End date**. There are a number of ways to fill in the dates. EViews will use the largest set of observations consistent with those dates, so if you enter "1970" and "2020", your quarterly workfile will begin in the first quarter of 1970, and end in the last quarter of 2020. Entering the date pair "Mar 1970" and "Nov 2020", or the start-end pair "3/2/1970" and "11/15/2020" would have generated a workfile with the same structure, since the implicit start and end quarters are the same in all three cases.

This latter example illustrates a fundamental principle regarding the use of date information in EViews. Once you specify a date frequency for a workfile, EViews will use all available calendar information when interpreting date information. For example, given a quarterly frequency workfile, EViews knows that the date "3/2/1990" is in the first quarter of 1990 (see "Dates" on page 94 of the *Command and Programming Reference* for details).

For some choices of frequency, you will be prompted for additional information. If, for example, you select the **Daily - custom week** frequency, the dialog will prompt you to provide a range for the days of the week in your custom week. You should use the **Intraweek/Intraday range** dropdown menus to define a the start and end of the week. Here, we indicate that we want to define a custom 4-day (Monday-Thursday) week.

If you choose the **Intraday** frequency you will be prompted to provide additional information about the structure of your observed days. You must provide information about the frequency of observations within the day, the days of the week for which you observe data, and the time, and the time range of observations within a day.

The intraday settings are best illustrated through a simple example. To create an hourly workfile that begins at 8AM Monday and ends at 5PM Friday for every day in 2005, you should also specify the **Day** and **Time** parameters.

- Select **Dated - regular frequency** for the workfile structure, and then choose the **Intraday** frequency. Select **Hour** from the dropdown that appears.

- Enter the **Start date** and **End date**. In this case, this will be 2005 for both fields.

- After selecting an **Intraday** frequency, the **Intraweek/ Intraday range** options were enabled at the bottom of the dialog. Select **1-Mon** through **5-Fri** from the **Days** dropdowns and **8:00:00** through **17:00:00** from the **Time** dropdowns.

Note that the workfile created will have ten observations per day, the first at 8 a.m. and the last at 5 p.m. Though you may have intended to create a workfile to record nine hours of data, specifying an end time of 17:00:00 defines the period from 5 p.m. to 6 p.m., adding an extra hour to the day specification. This may be desirable for recording discrete measurements on the hour from 8 a.m. to 5 p.m., but if you wish to measure the hours as intervals (from 8 a.m. to 9 a.m., 9 a.m. to 10 a.m., ..., and 4 p.m. to 5 p.m.), with one observation for each interval, you should specify an end time of 16:00:00, or 4 p.m. An end time of 16:00:00 defines the last interval as 4 p.m. to 5 p.m (see "Specifying Start and End Times," on page 275 for a thorough discussion of how start and end times are defined).

Lastly, you may optionally provide a name to be given to your workfile and a name to be given to the workfile page.

After creating a workfile, you can see EViews' interpretation of your workfile specification by looking at the **Workfile structure** dialog. Select **Proc/Structure/Resize Current Page...** or click on the **Range** area at the top of the new workfile to bring up the **Workfile structure** dialog (see "Applying a Structure to a Workfile," on page 273 for detailed discussion of structuring a workfile and the settings in this dialog).

For the previous example, you may take note of two things in the dialog. First, the **Start date** is displayed as "1/03/2005 08:00", which corresponds to 8 a.m. this is the first Monday in 2005. Similarly, the **End date** is "12/30/2005 17:00", the last Friday in 2005. Second, notice that the time in the **End date** field is different from the time displayed in the **Time** end field. This illustrates that because the **End date** is specified in hours, an end time of 17:00 defines the period from 5 p.m. to 6 p.m. Alternately, the **Time** field denotes the last second that is part of the specification, 17:59:59.

Describing an Unstructured Workfile

Unstructured data are simply data which use the default integer identifiers. You should choose the **Unstructured** type if your data are not a **Dated - regular frequency** or **Balanced Panel** (*i.e.,* you wish to create an undated workfile that uses the default identifiers or you will eventually define an advanced structure your data).

When you select this structure in the dropdown menu, the remainder of the dialog will change, displaying a single field prompting you for the number of observations. Enter the number of observations, and click on **OK** to proceed. In the example depicted here, EViews will create a 500 observation workfile containing integer identifiers ranging from 1 to 500.
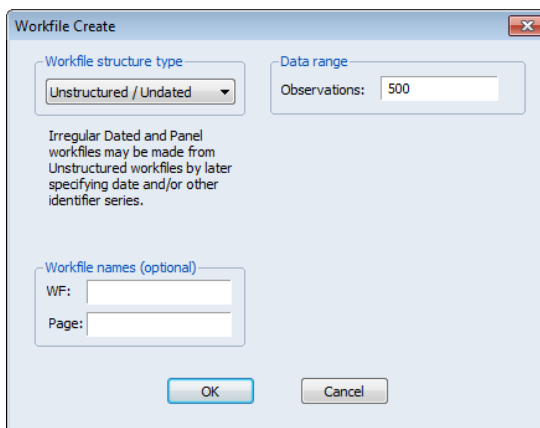
In many cases, the integer identifiers will be sufficient for you to work with your data. In more complicated settings, you may wish to further refine your identifiers. We describe this process in "Applying a Structure to a Workfile" on page 273.

Describing a Balanced Panel Workfile

The **Balanced Panel** entry provides a simple method of describing a regular frequency *panel data* structure. Panel data is the term that we use to refer to data containing observations with both a group (cross-section) and time (within-group) identifiers.

This entry may be used when you wish to create a balanced structure in which every cross-section follows the same regular frequency with the same date observations. Only the barest outlines of the procedure are provided here since a proper discussion requires a full description of panel data and the creation of the advanced workfile structures. Panel data and structured workfiles are discussed at length in "Structuring a Workfile" on page 263.

To create a balanced panel, select **Balanced Panel** in the dropdown menu, specify the desired **Frequency**, and enter the **Start date** and **End date**, and **Number of cross sections**. You may optionally name the workfile and the workfile page. Click on **OK**. EViews will create a balanced panel workfile of the given frequency, using the specified start and end dates and number of cross-sections.

Here, EViews creates a 200 cross-section, regular frequency, quarterly panel workfile with observations beginning in 1970Q1 and ending in 2020Q4.

Unbalanced panel workfiles or workfiles involving more complex panel structures should be created by first defining an unstructured workfile, and then applying a panel workfile structure.

Creating a Workfile by Reading from a Foreign Data Source

A second method of creating an EViews workfile is to open a foreign (non-EViews format) data source and to read the data into an new EViews workfile.

One way to read foreign data into a new workfile is to copy the foreign data source to the Windows clipboard, right click on the gray area in your EViews window, and select **Paste as new Workfile**. EViews will automatically create a new workfile containing the contents of the clipboard. Such an approach, while convenient, is only practical for small amounts of data.

Alternately, you may open a foreign data source as an EViews workfile. There are two ways to proceed. First, you may select **File-Import/Import from File…** to bring up the standard file **Open** dialog. Clicking on the **Files of type** dropdown menu brings up a list of the file types that EViews currently supports for opening a workfile. Select a file type to narrow the list of files in the dialog, then choose the file you wish to open. Note that the file can be web-based, in which case you may provide a URL for a file or HTML page, or it can be located on your major provider cloud drive ("Cloud Drive Support" on page 96).

EViews also offers drag-and-drop support for the file open operation. Simply select the file you wish to open in Windows, then drag-and-drop the file on the gray area in your EViews window.

Whichever your approach, EViews will open the specified foreign data as an EViews workfile.

If you select a time series database file (*e.g.*, Aremos TSD, GiveWin/Pc-Give, Rats 4.*x*, Rats Portable, TSP Portable), EViews will create a new, regular frequency workfile containing the contents of the entire file. If there are mixed frequencies in the database, EViews will select the lowest frequency, and convert all of the series to that frequency using the default conversion settings (we emphasize here that all of these database formats may also be opened as databases by selecting **File/Open/Database...** and filling out the dialogs, allowing for additional control over the series to be read, the new workfile frequency, and any frequency conversion).

If you choose one of the remaining source types, EViews will create a new unstructured workfile. First, EViews will open a series of dialogs prompting you to describe and select data to be read. The data will be read into the new workfile, which will be resized to fit. If possible, EViews will attempt to structure the workfile using information in the file. If this is not possible but you still wish to apply a specific workfile structure with these data, you will have to define a structured workfile using the advanced workfile structure tools (see "Structuring a Workfile" on page 263).

The import as workfile interface is available for Microsoft Access files, Gauss Dataset files, ODBC Dsn files, ODBC Query files, SAS Transport files, native SPSS files (using the SPSS Input/output .DLL that should be installed on your system), SPSS Portable files, Stata files, Excel files, raw ASCII or binary files, or ODBC Databases and queries (using an ODBC driver already present on your system).

For many of these files, you can link the data in the workfile to the original data source, allowing the EViews workfile to update its data to the data contained in the source file. At the end of the open process, EViews will ask if the new series objects should be linked back to the source. Click **Yes** to mark the new series objects as externally linked. Linked objects can be refreshed with updated data when the workfile is next opened, or on demand by right-clicking on the workfile window and selecting **Update from DB...** or by clicking on **Objects/Manage Links & Formulae...** in the main menu to display the link management dialog.

When you next open your existing workfile containing linked data, EViews will prompt you to refresh the data, and optionally, to display a comparison of the data before and after the refresh:
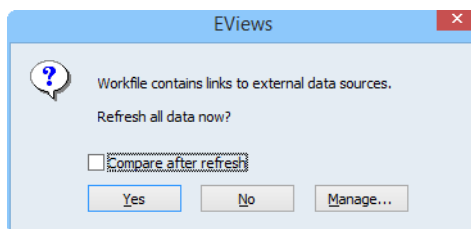
An Illustration

Perhaps the most common form of foreign data files are Excel files, followed by text files (ASCII). We offer examples of reading Excel, ASCII, and Stata files to illustrate the basic process of creating a new workfile (or a workfile page) by opening a foreign source file.

(The `wfopen` command offers command line support for all of these methods. See `wfopen` (p. 526) in *Command and Programming Reference*.)
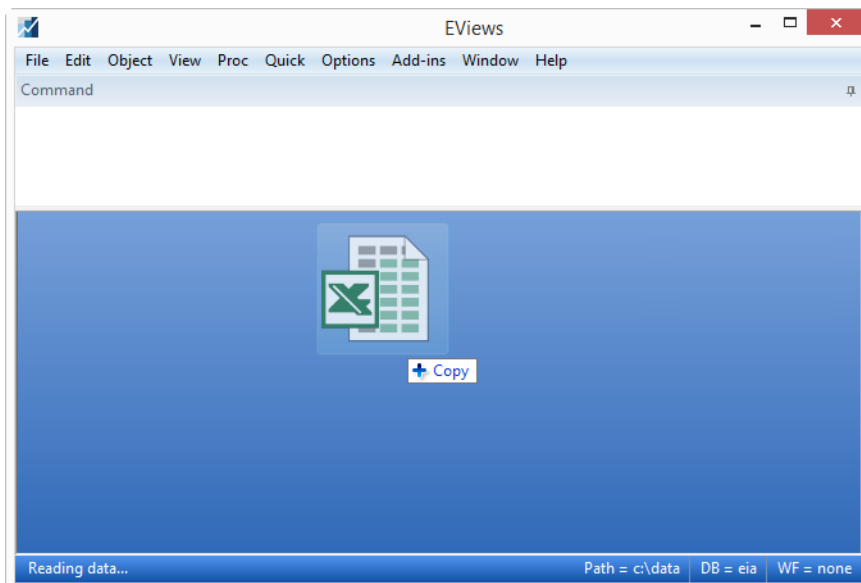
The easiest way to specify the foreign file to be opened is to drag-and-drop the file into background area of the main EViews window.

Here we have dragged an Excel file onto the EViews desktop. A plus (" + ") sign will appear when the file is over an appropriate area. Drop the file and EViews will proceed with the foreign file open process, first opening the selected file, validating its type, then displaying a tabbed dialog allowing you to select the specific data that you wish to read into your new

workfile. If you wish to read all of your data using the default settings, click on **OK** to proceed. Otherwise you may use each of the tabs to change the read behavior.
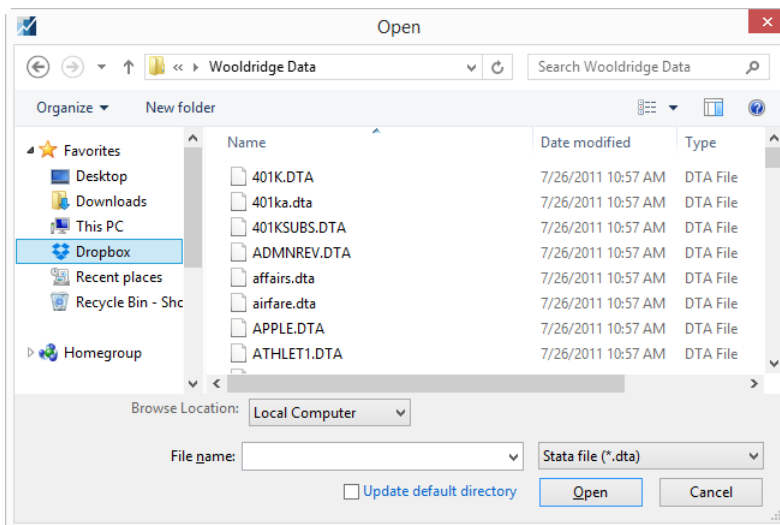
Alternately, you may specify the foreign file using the **File-Open/Foreign data as workfile...** dialog. First navigate to the appropriate directory and select the **Files of type** entry to match the file you wish to open. In this example, we see that if we select **Stata files (*.dta)** the dialog displays a list of Stata files that may be opened.

Double-click on the name to select and open the file, or enter the filename in the dialog and click on **Open** to accept the selection. Note that the **Open** dialog displayed here is the one for the Windows 8 operating system. The dialog may look different in other versions of Windows.

Opening an Excel file

If you have chosen an Excel file, the first dialog that will open is **Step 1** of the **Excel 97-2003 Read** dialog. This page lets you select which data in the Excel file you wish to read into EViews. You can choose from either one of the **Predefined range** options (a list of Excel ranges that EViews has automatically selected), or you can specify a **Custom range**, and enter the **Sheet** name and the **Start Cell** and **End Cell** of the data. The bottom half of this dialog presents a preview of the incoming data, based upon the range you have selected. In this example we have opened the file "academic salaries by discipline.XLS". Here the predefined range chosen by EViews is exactly the data we wish to read.

If you wish to transpose the incoming data, which will arrange the data by row rather than by column, check the **Read series by row** checkbox at the bottom of the dialog.

If you are happy with the way the previewed data looks, you can click **Finish**, and EViews will open up a new workfile containing the Excel data. For further adjustments to the data import, you can click **Next** rather than **Finish** and open up **Step 2** of the **Excel 97-2003 Read** dialog.

The **Column headers** section of **Step 2** allows you to specify the number of header lines (lines containing information about each column of the data, such as column names or descriptions, rather than the data itself) contained in the Excel file, and tell EViews a little bit about what information is stored in the header lines, using the **Header type** dropdown box.

If the data in your Excel file has special coding for missing values, you can set the **Text representing NA** box to whatever value your data uses.

EViews will fill out the **Column info** box automatically, based upon your choices in the **Column headers** box, and by analyzing the data in your file. If you would like to change the name, description or data type (**Character**, **Number** or **Date**) of a specific column, select that column in the preview box at the bottom of the dialog, and then change the information.

Again, if you would like to transpose your data, you may elect to do so by checking the **Read series by row** checkbox at the bottom of the dialog.

Once you have described the file and data, you may click **Finish**. Alternately, to further specify details on how you wish to perform the import of data, click **Next** to open up **Step 3** of the **Excel 97-2003 Read** dialog.
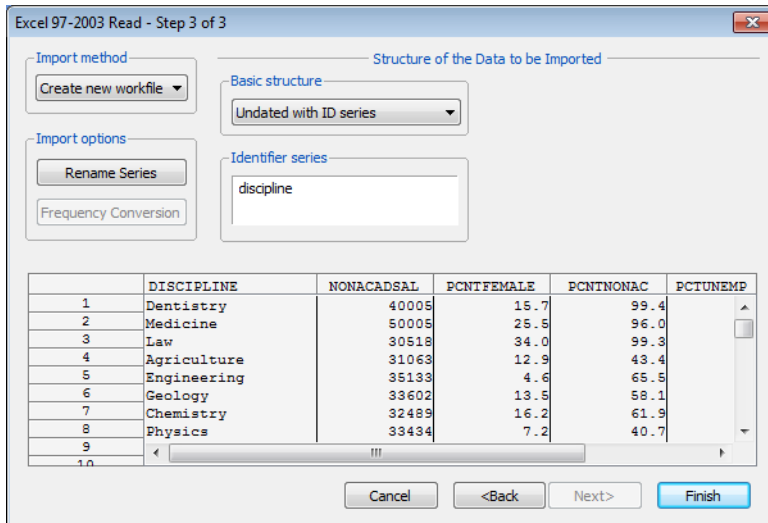
Some options in this dialog are simplified because we are reading into a new workfile. As such, we don't need to worry about how our data fits into the structure of an existing workfile. For instance, the **Import method** dropdown in the top-left portion of the dialog will be set to **Create new workfile** in this case. In the **Import options** section, the **Frequency Conversion** button will be disabled, as we have no need to define frequency conversion methods when importing into a new workfile. See "Importing Data" on page 146 for information on reading into an existing workfile.

The **Import options** section offers access to settings for renaming series. Renaming the series on import allows you to handle illegal input series names, and to avoid name conflicts with existing series.

The **Rename Series** button brings up a dialog allowing you to perform multiple series renaming. Renaming should be done by entering an existing series name followed by the new series name in pairs. For example, entering

```
series01 gdp series02 unemp series03 m1
```

will rename the series SERIES01 to GDP, SERIES02 to UNEMP, and SERIES03 to M1.
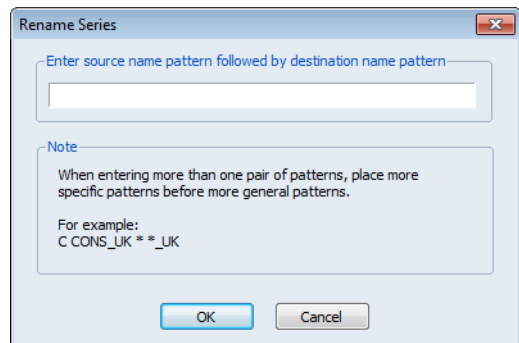
You may use the "*" and "?" wildcards as part of the renaming specification to rename multiple series at once. For example,

```
series* gdp*
```

changes the names of all series with an existing prefix of "SERIES" to "GDP". In other words, SERIES01 will become GDP01 and SERIES_USA will become GDP_USA.

The area to the right of the **Import options** dropdown may be used to define the structure of the new workfile.

The bottom portion of the dialog consists of a preview display, which shows a sample of destination observations along with corresponding data from the source file. The preview will change along with your selected import method specification.

When the source file is first opened, EViews will attempt to identify the frequency and potentially a date series in the file (*i.e.* a column of data containing dates). If a source date series is located, the **Basic structure** dropdown will be set to **Dated - specified by date series** and the source name will be entered in the **Date series** edit field.

To aid you in visualizing the effect of the **Basic structure** setting, the left-hand column of the data preview display updates with changes in the structure to show you the current date settings.

In most cases, EViews will have correctly identified the date structure so that the default settings will require no change. Click on the **Finish** button complete the import specification and instruct EViews to begin the import procedure.

You may, if desired, override the default EViews date specification settings by changing the **Basic structure** dropdown menu, and then using the dialog to specify the data frequency explicitly.

For example, if you select **Dated - regular frequency** in the **Basic structure** dropdown, the dialog changes to reflect the new settings:

Notice that EViews no longer shows the date series under **Basic structure**, instead prompting you to provide a **Frequency/date specification** using the **Frequency** dropdown menu and the **Start date** edit field. EViews will assume that the source file has sequential observations of the given frequency beginning in the specified date.

Since we have not yet specified a **Start date**, the "?"s in the observation preview at the bottom of the dialog indicate that the observation matching is indeterminate. Once a start date is specified, the display will be updated to show how the observations will be imported.

When **Dated Panel** is selected in the **Basic structure** dropdown, the import structure section of the dialog provides different options, prompting you to specify a **Cross section ID series** and a **Date series**.
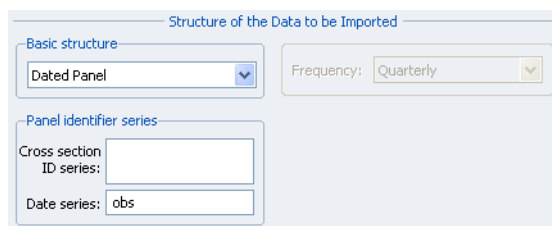
When you are pleased with the setup shown at the bottom of the dialog, click on the **Finish** button.

The final step of the import process involves deciding whether to link the EViews file back to the original Excel file. EViews will ask if the new series objects should be linked back to the source. Linked data remembers the specification of its source and can be updated or refreshed to re-import the data. Click **Yes** to mark the new series objects as externally linked. Linked objects can be refreshed with updated data when the workfile is next opened, or on demand by right-clicking on the workfile window and selecting **Update from DB…** or by

clicking on **Objects/Manage Links & Formulae...** in the main menu to display the link management dialog.

When you next open your existing workfile containing linked data, EViews will prompt you to refresh the data, and optionally, to display a comparison of the data before and after the refresh:
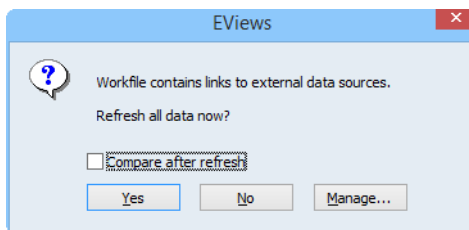
Opening an ASCII text file

If you have chosen an ASCII text file, EViews will open **Step 1** of the **Text Read** dialog. This page lets you specify column and row information for your file. Generally EViews will analyze your file and fill this data in correctly, so you can just click **Finish**. If EViews has not analyzed your data correctly, you can make changes during the four steps of the **Text Read** dialog.
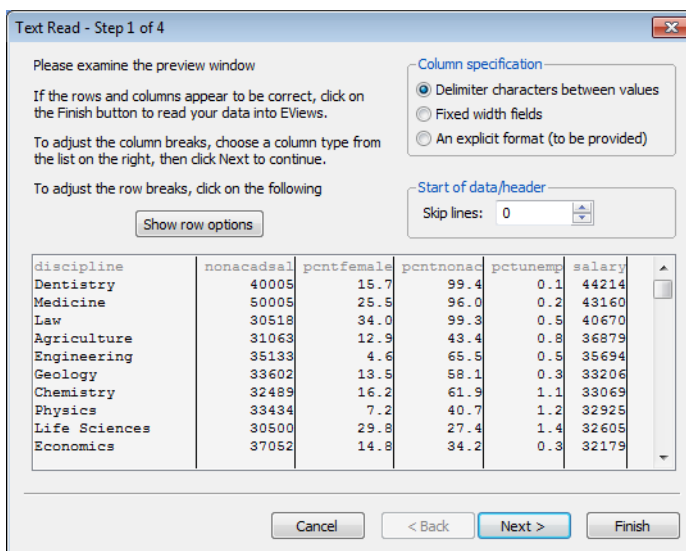
If your text file has lines of text at the start of the file that come before the actual columns of data (such as text explaining what the data is, or where it came from), you can tell EViews to skip these lines as part of the data read in the **Start of data/header** section.

Most text files containing data will have a row of text for each observation, and EViews, by default, will read in the data assuming this is the case. For more complicated structures, you should press the **Show row options** button, which will then display a set of options that allow you to specify how each row of your file is formatted.

The **Column specification** box lets you specify how each column of data is separated. The most common method of separating columns in a text file is through the use of a delimiter character between each column, such as a comma, space, or tab. A second form of separation is through fixed width columns, in which each column has the same number of characters in it for every row. If your data has a fixed width, you should select **Fixed width fields**. If your text file uses neither delimiters nor fixed widths, you should select **An explicit format (to be provided)**.

Step 2 of the **Text Read** dialog will change depending on the choices you made in the **Column specification** box during **Step 1**. If you chose **Delimiter characters between values**, then **Step 2** lets you specify information about the delimiters. You can either choose from one of the preset **Standard delimiters**, or specify a **Custom delimiter**. You can also choose how

EViews treats delimiters that are part of text enclosed in quotes using the **Other options** section.

If you chose **Fixed width fields** as your column separator, **Step 2** will let you specify the width of each column. Finally, if you chose **An explicit format**, then **Step 2** lets you specify the format string.

Steps 3-4 of the **Text Read** dialog are exactly the same as **Steps 2-3** of the **Excel Read** dialog, as explained above beginning on . **Step 3** allows you set information regarding the data columns of your file, while in **Step 4** you can specify the new workfile structure and rename series.
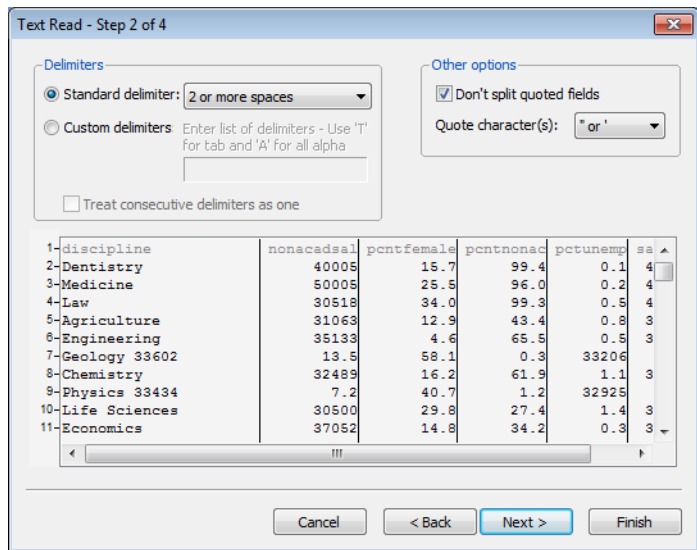
Opening a Stata file

If you had chosen a Stata file (or other file containing predefined variable information) to be read, the **Table read specification** dialog will open, letting you select which variables to read.

The **Select variables** tab of the dialog should be used to choose the series data to be included. The upper list box shows the names of the variables that can be read into EViews series, along with the variable data type, and if available, a description of the data. The variables are first listed in the order in which they appear in the file. You may choose to sort the data by clicking on the header for the column. The display will be toggled between three states: the original order, sorted (ascending), and sorted (descending). In the latter two cases, EViews will display a small arrow on the header column indicating the sort type. Here, the data are sorted by variable name in ascending order.
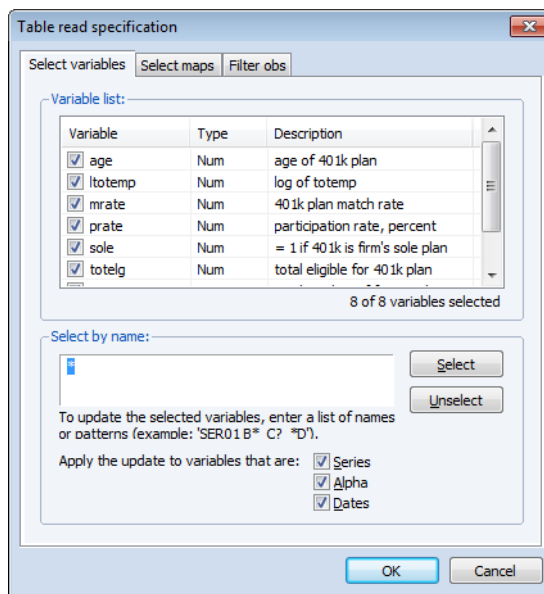
When the dialog first opens, all variables are selected for reading. You can change the current state of any variable by checking or unchecking the corresponding checkbox. The number of variables selected is displayed at the bottom right of the list.

There may be times when checking and unchecking individual variables is inconvenient (*e.g.*, when there are thousands of variable names). The bottom portion of the dialog provides you with a control that allows you to select or unselect variables by name. Simply enter the names of variables using wildcard characters if desired, choose the types of interest, and click on the appropriate button.

For example, entering "A* B?" in the selection edit box, selecting only the **Numeric** checkbox, and clicking on **Unselect** will uncheck all numeric series beginning with the letter "A" and all numeric series with two character names beginning in "B". To select *only* series beginning with the letter "A", enter "*" and click on **Unselect**, then enter "A*" and click on **Select**.

When opening datasets that contain value labels, EViews will display a second tabbed dialog page labeled **Select maps**, which controls the importing of value maps. On this page, you will specify how you wish EViews to handle these value labels. You should bear in mind that when opening datasets which do not contain value labels, EViews will not display the value map tab.

The upper portion of the dialog contains a dropdown menu where you specify which labels to read. You may choose between the default **Attached to selected series**, **None**, **All**, or **Selected from list**.
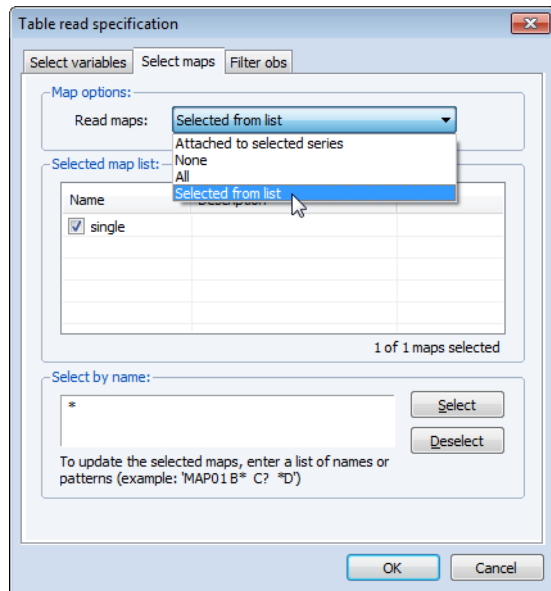
The selections should be self-explanatory—**Attached to selected series** will only load maps that are used by the series that you have selected for inclusion; **Selected from list** (depicted) displays a map selection list in which you may check and uncheck individual label names along with a control to facilitate selecting and deselecting labels by name.

Lastly, the **Filter obs** page brings up an observation filter specification where you may enter a condition on your data that must be met for a given observation to be read.

When reading the dataset, EViews will discard any observation that does not meet the specified criteria. Here we tell EViews that we only wish to keep observations where AGE > 10.

Once you have specified the characteristics of your table read, click on **OK** to begin the procedure.

EViews will open the foreign dataset, validate the type, create a workfile, read the selected data, and if possible, apply a workfile structure. When the procedure is completed, EViews will display an untitled group containing the series, and will display relevant information in the status line. In this example, EViews will report that after applying the observation filter it has retained 636 of the 1534 observations in the original dataset.
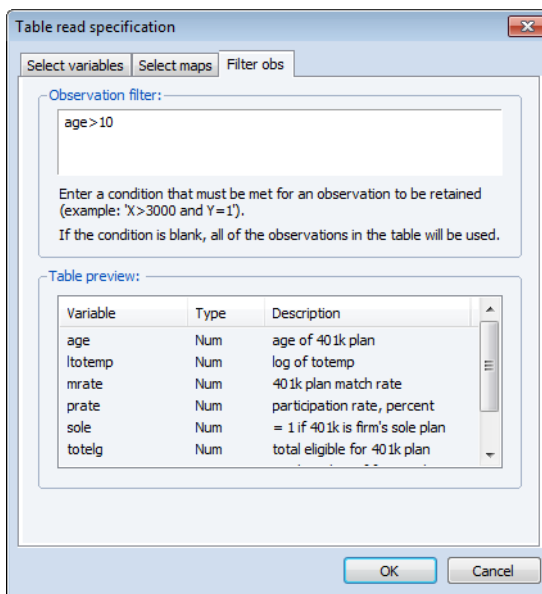
The Workfile Window

Probably the most important windows in EViews are those for workfiles. Since open workfiles contain the EViews objects that you are working with, it is the workfile window that provides you with access to all of your data. Roughly speaking, the workfile window provides you with a directory for the objects in a given workfile or workfile page. When open, the workfile window also provides you with access to tools for working with workfiles and their pages.

Workfile Directory Display

The standard workfile window view will look something like this:

Workfile Range and Structure
(click to structure or resize)
Title Bar
Button Bar
Current
/Filter
(click to change)

Current
Sample
(click to
change)

Click to
scroll
page tabs

Workfile
Directory

Workfile page tabs (click to set active page)

Click here to create
a new workfile page
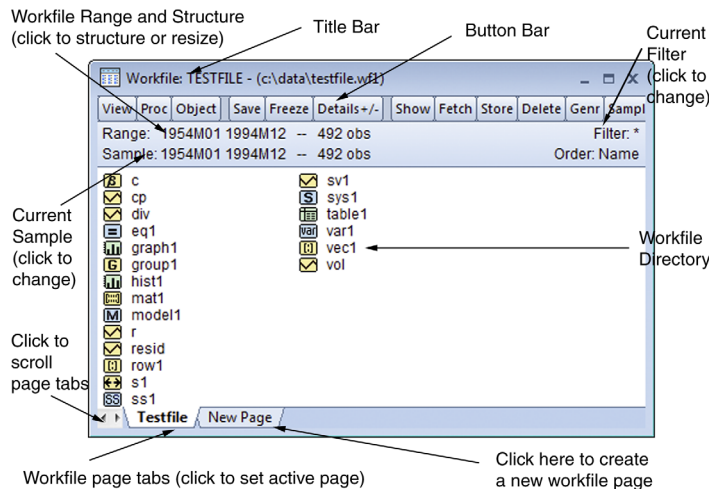
In the title bar of the workfile window you will see the "**Workfile**" designation followed by the workfile name. If the workfile has been saved to disk, you will see the name and the full disk path. Here, the name of the workfile is "TESTFILE", and it is located in the "c:\eviews\data" directory on disk. If the workfile has not been saved, it will be designated "UNTITLED".

Just below the titlebar is a button bar that provides you with easy access to useful workfile operations. Note that the buttons are simply shortcuts to items that may be accessed from the main EViews menu. For example, the clicking on the **Fetch** button is equivalent to selecting **Object/Fetch from DB...** from the main menu.

Below the toolbar are two lines of status information where EViews displays the *range* (and optionally, the structure) of the workfile, the current *sample* of the workfile (the range of observations that are to be used in calculations and statistical operations), the display filter (rule used in choosing a subset of objects to display in the workfile window), and an indicator for the order in which the objects in the workfile are displayed. Here we see that the objects are currently in alphabetical order. You may change the range, sample, and filter by double clicking on these labels and entering the relevant information in the dialog boxes.

Lastly, in the main portion of the window, you will see the contents of your workfile page in the workfile directory. In normal display mode, all named objects are listed in the directory, sorted by name, with an icon showing the object type. The different types of objects and their icons are described in detail in "Object Types" on page 103. You may also show a subset of the objects in your workfile page, as described below in "Filtering the Workfile Display" on page 73, or change the display order as outlined in

You may choose **View/Name Display** in the workfile toolbar to specify whether EViews should use upper or lower case letters when it displays the workfile directory. The default is lower case.

It is worth keeping in mind that the workfile window is a specific example of an object window. Object windows are discussed in "The Object Window" on page 112.

Workfile Details Display

You may change the default workfile display to show additional information about your objects. If you select **View/Details + /–**, or click on the **Details + /-** button on the toolbar, EViews will toggle between the standard workfile display format, and a display which provides additional information drawn from the label of each object. (See "Adding Custom Attributes in the Label View," on page 117 for details on modifying label information in your objects.)
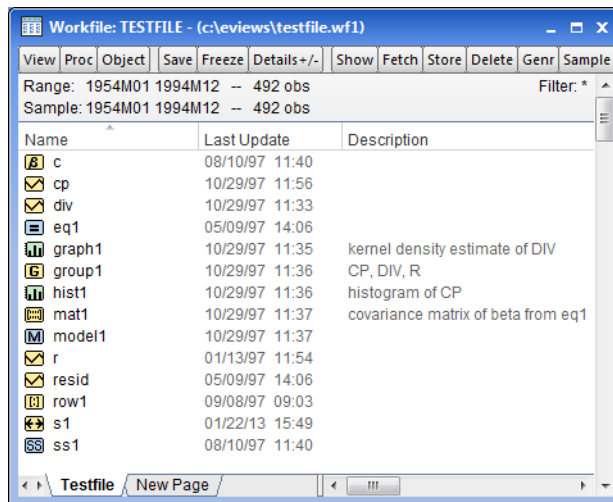
Working with Built-in Attributes

From this view, you can sort the objects in the workfile using the values in any of the columns. Simply click on a column header to sort by that column, and to alternate between an ascending or descending sort order. A small gray triangle pointing up indicates an ascending sort, while a triangle pointing down indicates a descending sort.

With the exception of the NAME column, which must remain in the first column, you may reorder the columns by dragging each to your desired location. The columns can be resized by placing the cursor over the gray dividing line between column headers (the cursor will switch to a sizing image) and dragging to the left or right.

Size Column to Fit
Size All Columns to Fit

Clear Attribute Values
Replace Attribute Values...
Rename/Merge Attribute...
Import Attributes...

Add Column...
Remove Column
Edit Columns...
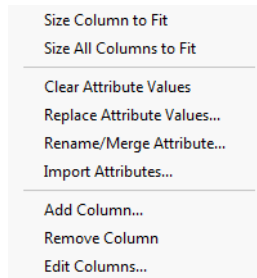
To size a column to the width of its longest text string, right-click on the column header and select **Size Column to Fit** from the menu that appears. If you'd like to size all the columns in the workfile view to accommodate the text within each, select **Size All Columns to Fit** from the right-click menu.

By default, the **Details** view shows NAME, LAST UPDATE, and DESCRIPTION for each object. You can customize this view to include any of the attributes in your objects. Right-click on the column header and select **Edit Columns...** from the menu. You will see a dialog displaying all available attributes in your work-file. These items are a collection of the individual label attributes within each object. While each object in the workfile may not have the same attributes, the dialog presents all possible attributes that can be displayed as columns in the workfile **Details** view.

Workfile columns

Displayed attributes:
- ☑ Name
- ☑ Last Update
- ☑ Start
- ☑ End
- ☑ Description
- ☐ Type
- ☐ Freq
- ☐ Obs
- ☐ Convert Lohi
- ☐ Convert Hilo
- ☐ Display Name

OK   Cancel
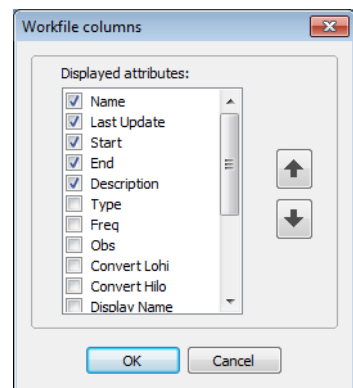
You will notice that initially, NAME, LAST UPDATE, and DESCRIPTION are checked, indicating they are visible in the workfile **Details** view. To add additional columns, click on the checkbox in front of the attribute you wish to see. You can reorder the columns from within the dialog by selecting an attribute and clicking the up and down arrows on the right-hand side. Again, the NAME column is locked in the first position and cannot be moved or hidden. The attribute columns will be displayed in the workfile in the order you specify after you click **OK** in the dialog.

All available information for each attribute will be displayed. However, some objects may not contain label information for an attribute, in which case you will see a blank cell in the attribute column.

Adding columns to the workfile **Details** view provides you with a convenient way to look at information across the objects in your workfile. Instead of opening each object to view its observation count, for instance, you can add OBS to the workfile and look at the number of non-NA observations in every series in your workfile at once. Additionally, there are many more attributes available via the **Edit Columns** dialog than are visible in the **Label** view of

an object. For instance, you can view the **Start** and **End** dates of your objects in the workfile **Details** view by adding them as columns.

The sort function provides a method for grouping objects in your workfile. For example, you could sort by the LAST UPDATE column to view objects that have been created or modified recently, or by the SOURCE column to group objects from a particular data source.

The rules EViews uses to sort the workfile window are intuitive, but it is worth mentioning a couple of them. First, empty values are placed at the end of the list, whether the column is sorted in ascending or descending order. Additionally, if you add the FREQ attribute to display the frequency of each object and sort by this column, you will see that frequency is sorted from low to high (or high to low), not by the letter representation in the column. (You could have multiple frequencies if you have a series link to another page of a different frequency, for instance.)

At the end of the list in the default **Edit Columns** dialog, you will see the items "Deptype" and "Depspec". These are the dependency type and dependency spec, respectively. The dependency type provides information about the source of the data, when appropriate. It may be empty or one of three values: "dblink", "pagelink", or "formula". The dependency spec gives additional information about the dependency.

For instance, if a series is based on a formula, its dependency type would be "formula" and its dependency spec might be "log(x)". If a series is a link from another workfile page, its dependency type would be "pagelink", and its dependency spec might provide information about its source workfile page. Finally, a series that has been pasted from a database would have the dependency type "dblink", and the source database name and series could be stored in the dependency spec.

To remove a column from view, unselect the checkbox in front of the item in the **Edit Columns...** dialog. Alternately, right-click on the column you wish to remove, and select **Remove Column** from the menu.

If you'd like to add a column without going through the **Edit Columns...** dialog, right-click on any column header and choose **Add Column...**. The **Add Column** dialog will appear, into which you may enter the name of the attribute you'd like to add as a column in the workfile **Details** view. This field is not case sensitive, but otherwise must match the attribute name exactly. You can also click the arrow on the dropdown menu to select from a list of all available attributes. (This is the same list that is used to fill the **Edit Columns** dialog.)
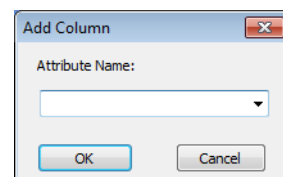
Note that **Add Column...**, **Remove Column**, and **Edit Columns...** modify the appearance of the workfile **Details** view. They do not add or change attributes within the objects. As such,

you can add a column that doesn't exist as an attribute for any object in your workfile. EViews will display the column anyway, showing all blank values.
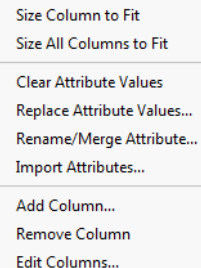
You may customize each workfile page independently, choosing to display different columns and grouping your objects differently according to the sort order.

When you save the workfile, the columns you have chosen to show will be stored, along with their widths, the display order and the sort method. Your settings will be saved whether the workfile is currently in **Details** view or not. You may also freeze the workfile window in this view, providing you with a detailed table that can be copied or saved. To save the table, right-click in the table and select **Save table to disk...**.

Adding Your Own Label Attributes

While EViews provides a variety of attributes by default, you may like to add your own. You can add them directly to individual objects, or globally from the workfile window. To learn more about adding custom attributes to individual objects, which will then appear in the list of attributes for the workfile **Details** view, see "Adding Custom Attributes in the Label View," on page 117. Alternately, to add attributes to every object in the workfile, you may use the tools provided in the workfile window.

From within a column header in the workfile **Details** window, right-click to bring up the menu. We will turn our attention to the items in the middle of the menu (**Clear Attribute Values**, **Replace Attribute Values...**, **Rename/Merge Attribute...**, and **Import Attributes...**). These menu items modify the actual attributes within your objects, while the items we have already discussed (**Add Column...**, **Remove Column**, and **Edit Columns...**) simply change the appearance of the workfile **Details** view.

We will illustrate the use of these options with an example. Say we have a workfile containing information on the population and output for a number of countries, from 1950 to 2000. For each country, there is a series with population data and one with output data. The series are named such that a prefix indicates the series type (POP for population and Y for output), and the last three characters specify the country. For instance, the population series for Canada is POPCAN. We would
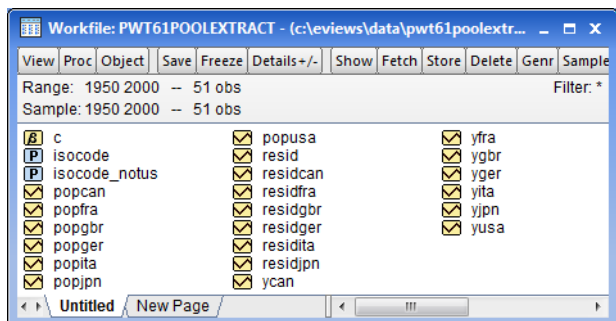
like to flag each series, so that we can easily see which contain population data and which contain output data.

From the workfile **Details** view, we add a column named SERTYPE (select **Add Column...** from the right-click menu, and enter "Sertype"). At this point, the attribute does not exist in any of our objects, and so the column is blank. We would like to add the attribute to the objects and populate it with information.
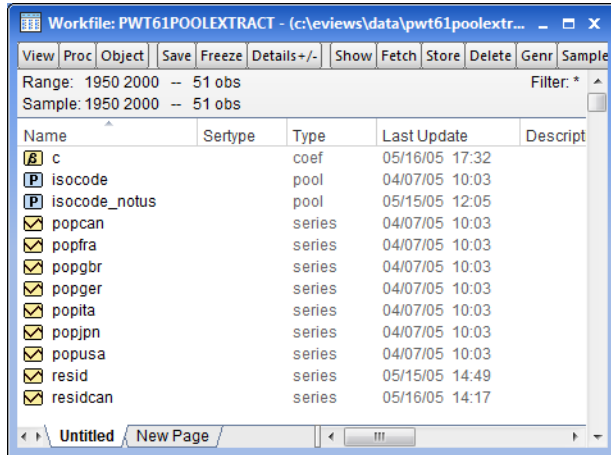
The values of an attribute can be set much easier if we first filter the workfile to show only the objects we'd like to edit. Double-click on the word **Filter** in the top right-hand corner of the workfile window to bring up the **Object Filter** dialog.
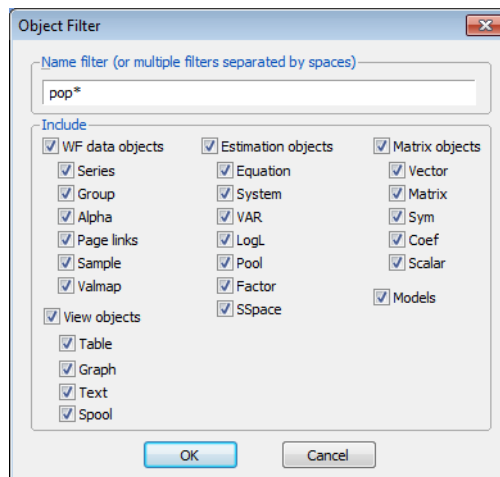
First, let's label the series with population data. Enter "pop*" in the **Name filter** edit field to filter the workfile to show only objects whose name begins with the letters "pop", and click **OK**. (See for more information on filtering your workfile.)

The workfile now shows only the series related to population, for each of the seven countries included in our data.
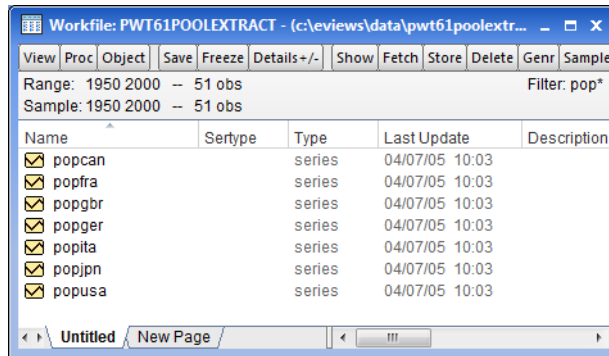
Now, we go into the label view of the series named POPCAN, add the SERTYPE attribute (see "Adding Custom Attributes in the Label View," on page 117), and set the value to "population". When we return to the workfile **Details** view, we will see "population" under the SERTYPE column for the POPCAN series. The other series do not have a SERTYPE attribute. You might imagine that adding the attribute and entering a value for every series could be tedious, especially if our workfile contained series for more countries. Let's explore an easier method of globally modifying your attributes.
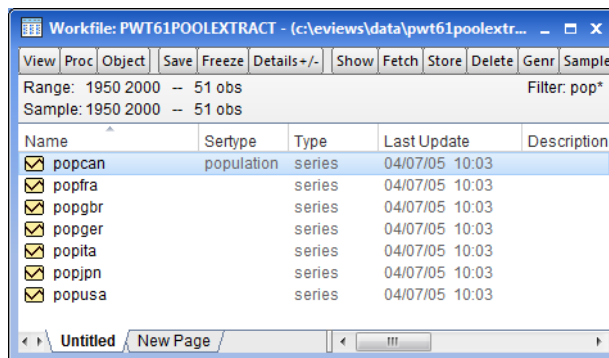
Right-click on the SERTYPE column to bring up the attribute modification dialog again, and select **Replace Attribute Values…**
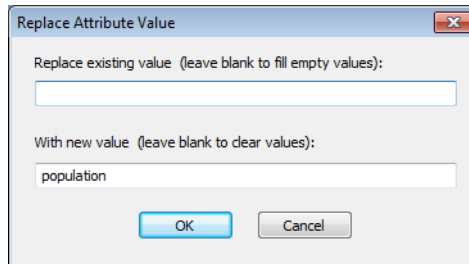
This dialog combines a few operations into one, by allowing you to enter a value into one field and leave the other blank, or by entering data into both fields. In turn, you can fill empty attributes with a value, replace a specific existing value with another value, or reset attributes to blank. We will discuss each as we continue our example.

First, let's add the SERTYPE attribute and fill it with the value "population" for the set of objects we have filtered. In the **Replace Attribute Value** dialog, leave the first edit field blank, enter "population" in the second edit field, and click **OK**. Note that this dialog is case-sensitive. Since we brought up the dialog by right-clicking on the SERTYPE column, this operation will replace all blank values of SERTYPE with "population", for all visible objects in the workfile. The SERTYPE attribute is new for all objects except the POPCAN series. Since the SERTYPE value is already set in this series, this operation will not modify its value. Click **OK** in the confirmation dialog to complete the process. We will see that every series in our current workfile view has been given the value "population" for its SERTYPE attribute.
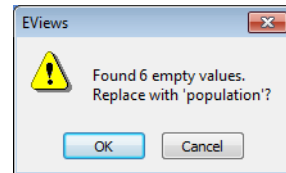
Now we can add and populate the SERTYPE attribute for objects with output data. Double-click the **Filter** text in the top right corner of the workfile window, bringing up the **Object Filter** dialog again. This time, enter "y*" in the edit field to filter the workfile to show only objects beginning with "y", or those related to output, and click **OK**.
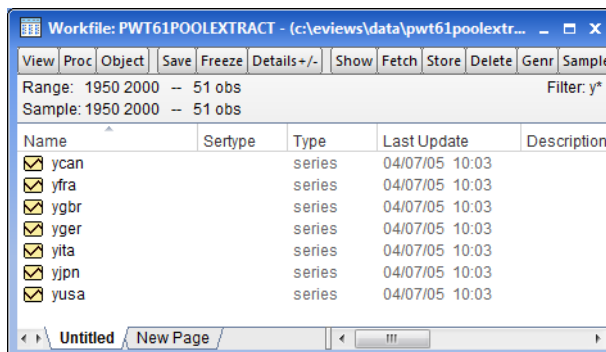
You may now right-click on the SERTYPE column, select **Replace Attribute Values...**, and repeat the process for the second set of filtered objects. Leave the first edit field blank, enter "output" in the second edit field, and click **OK**. The SERTYPE attribute will be populated with the text "output" for all the filtered objects.

When we remove the workfile filter (open the **Object Filter** dialog and enter "*"), we will see all our objects with the appropriate SERTYPE coding, indicating which series provide population data and which provide data about the country's output.

Now that we have populated all the objects, what if we notice that we typed "outPUT" instead of "output" in the case-sensitive **Replace Attribute Value** dialog. This can be easily fixed. Bring up the dialog again, and enter "outPUT" in the **Replace existing value** field, and "output" in the **With new value** field.
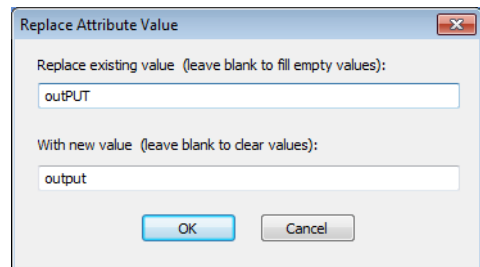
When you click **OK**, you will be prompted by a dialog confirming that you wish to replace all SERTYPE attributes containing the text "outPUT" with the text "output". Click **OK** to confirm. By using the **Replace Attribute Value** dialog, there was no need to go into each object individually; the label information was updated globally. You also didn't need to refilter the workfile. Only attributes with the text "outPUT" were modified.

Finally, the **Replace Attribute Value** dialog can also be used to clear specific attributes based on their value. For instance, now that all our objects related to a country's output are coded with "output", we could clear all SERTYPE attributes for these objects by entering "output" in the first edit field, and leaving the second edit field blank. All SERTYPE attributes with the value "output" will be removed from their objects.
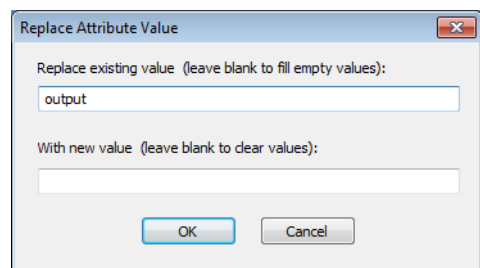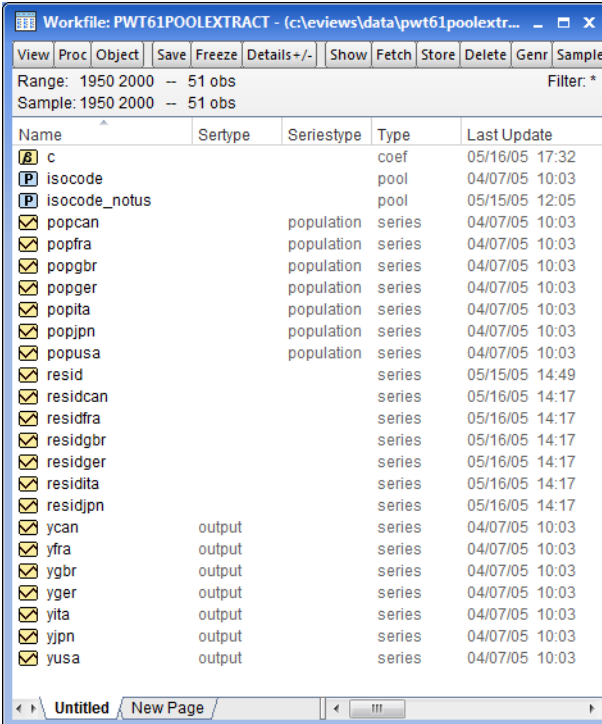
Another item in the right-click menu that can be used to edit the underlying attributes is **Rename/Merge Attribute...**. If you decide you would like your attribute to be named SERIESTYPE instead of SERTYPE, you may use this dialog to globally edit the attribute in all objects which are currently visible in your workfile. Right-click on the SERTYPE column, and select
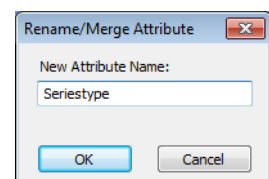
Rename/Merge Attribute... to bring up the dialog. Enter "Seriestype" and click **OK**. The SERTYPE attribute will be renamed SERIESTYPE in all objects currently in your workfile filter.

Say you accidentally have your workfile filter set to show only objects related to population ("pop*") when you rename the SERTYPE attribute. When you reset the filter to show all ("*"), you will see that the column has been renamed to SERIES-TYPE, and only objects with population data are showing their "population" attribute value. The series with output data still have an attribute named SERTYPE, which is not shown, as you no longer have a SERTYPE column in your workfile. If you now add back a column for the SER-TYPE attribute, (select **Add Column...** from the right-click menu, and enter "Sertype"), you will see that all series are labeled, but half of the values are in the SERTYPE attribute, and half are in the SERIESTYPE attribute. You can remedy this using the **Rename/Merge Attribute** dialog.

With the filter set to all ("*"), right-click in the SERTYPE column and select **Rename/Merge Attribute...**. Type "Seriestype" in the edit field, then click **OK** in the dialog and for the confirmation message. Now, there will no longer be a SERTYPE attribute, and all objects contain an attribute named SERIESTYPE.

We have described a rename operation with no conflicts. When you rename an attribute to one that already exists, EViews attempts to merge the attribute values. The attribute that you right-clicked on to bring up the dialog is considered the primary attribute. First, the primary attribute will be renamed to the name you entered in the edit field. Then, in preparing to delete the conflicting attribute, EViews tries to merge the values for each object. If both attributes have values for a given object, the text of the primary attribute is retained. If one attribute is blank, then the text is preserved, whether it be from the primary or conflicting attribute. Once the values have been used to populate the renamed attribute, the conflicting

attribute and the contents of its label values will be removed from all objects in the current workfile filter.

Another item in this section of the right-click menu is **Clear Attribute Values**. This option clears the contents of an attribute and removes the attribute from every object, provided it is not one of the fundamental locked types. Again, this is actually deleting the label information and removing the attribute across the entire workfile; it is not simply modifying the column in the workfile. You will see a warning dialog before the clear operation is performed, indicating how many instances of the attribute were found.

If we want to clear the SERIESTYPE attribute in our example, select **Clear Attribute Values** from the right-click menu while the mouse is over the SERIESTYPE column. EViews scans the workfile for objects containing the attribute SERIESTYPE, and displays the number of objects found in the warning dialog. Click **OK** to remove the label information and attribute from all the series we had labeled in our workfile.
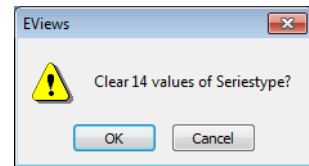
The SERIESTYPE column will remain in the workfile **Details** view, even though the attribute does not exist in any of our current objects. This is useful if we wish to add and repopulate the attribute. Alternately, we might have had a filter on the workfile at the time the clear operation was performed, such that when we return to a full view of all objects, some will have the attribute and can display values in the column. (See "Filtering the Workfile Display" on page 73.)

Importing Attributes

The previous example used a set of series which were named in such a way as to make workfile filtering effective. We were able to isolate the series related to population and those related to output quite easily. If your series are not named so conveniently, the **Import Attributes** feature may be a better way to create and populate your attribute.

Additionally, if you have the values of your attribute in some text format, you can setup a workfile page containing the values, and then import them into your series.

To illustrate, say we have a workfile with the series objects D2, S1, and S2 and an equation EQ1. We have received the data from different employees, and we'd like to add an attribute named CREATOR to keep track of who contributed each series.

We need something to hold your attribute values and to provide information about which value relates to which series. To accomplish this and get the information into the EViews world, we use another workfile page.
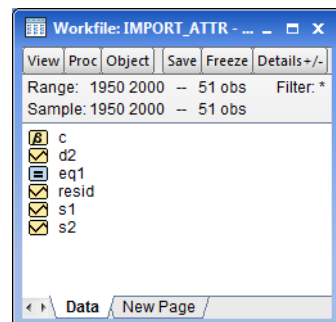
Click on the **New Page** tab at the bottom of the workfile window and select **Specify by Frequency/Range...** to bring up the **Workfile Create** dialog. It doesn't matter what frequency you use to define the new page; dates are completely irrelevant. What you should pay attention to is the number of observations, as each observation will be used to tie an attribute value to a series. In other words, you should have at least as many observations as the number of series you wish to populate with an attribute value.

We have four objects in our current workfile page, so if we have an attribute value for each, our start and end dates should define at least four observations. The default options create a new workfile page from 1950 to 2000, matching our current workfile. While this provides more observations than necessary, we accept the default and click **OK**. (We have also renamed the page to ATTRINFO by right-clicking on the new tab after it is created.)

The method EViews uses to populate an attribute requires two things: the attribute values and a list of the objects to receive these values. These are each held in an alpha series. As such, we create two alpha series, named CREATOR and OBJNAME. (This can be done by typing `alpha creator` and `alpha objname` into the command window.)

The CREATOR series will hold the values for whichever objects we'd like to provide information, and it is also the name of the attribute we will be defining. The OBJNAME series indicates which objects will be receiving attribute information. The series are aligned by observation number, such that the value of CREATOR for 1950 will be assigned to the CREATOR attribute of the object specified in OBJNAME for 1950.

Here, we have filled SERNAME with a list of object names in our first workfile page. Note that R5 does not exist, but this will not cause us problems. When entering the object names, remember that copying from a workfile page copies the names of the objects. You can then paste the names directly into an alpha series to get a list of your objects.

Since we entered the object names as each employee submitted their data, they are not in any particular order. This is fine, as long as the values in CREATOR are in the same order.

We have entered names into the CREATOR series to indicate that Andy provided data for the series S2, Mike provided D2, and so on.

If you have a source from which to copy and paste your text, this is clearly the easiest method. However, even if you must enter the information manually, this importing method is still superior to opening every object's label view to enter each value individually.

Filtering the Workfile Display

When working with workfiles containing a large number of objects, it may become difficult to locate specific objects in the workfile directory display. You can solve this problem by using the workfile display filter to instruct EViews to display only a subset of objects in the workfile window. This subset can be defined on the basis of object name as well as object type.
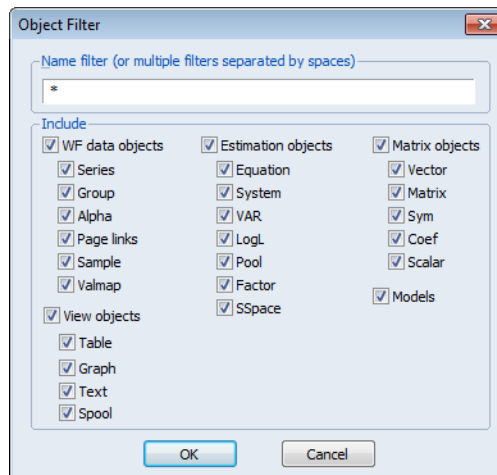
Select **View/Display Filter…** or double click on the Filter description in the workfile window. The following dialog box will appear:

There are two parts to this dialog. In the edit field (blank space) of this dialog, you may place one or several name descriptions that include the standard wildcard characters: "*" (match any number of characters) and "?" (match any single character). Below the edit field are a series of check boxes corresponding to various types of EViews objects. EViews will display only objects of the specified types whose names match those in the edit field list.

The default string is "*", which will display all objects of the specified types. However, if you enter the string:

    x*

only objects with names beginning with X will be displayed in the workfile window. Entering:

    x?y

displays all objects that begin with the letter X, followed by any single character and then ending with the letter Y. If you enter:

    x* y* *z

all objects with names beginning with X or Y and all objects with names ending in Z will be displayed. Similarly, the more complicated expression:

    ??y* *z*

tells EViews to display all objects that begin with any two characters followed by a Y and any or no characters, and all objects that contain the letter Z. Wildcards may also be used in more general settings—a complete description of the use of wildcards in EViews is provided in Appendix A. "Wildcards," on page 771 of the *Command and Programming Reference*.

When you specify a display filter, the **Filter** description in the workfile window changes to reflect your request. EViews always displays the current string used in matching names. Additionally, if you have chosen to display a subset of EViews object types, a "–" will be displayed in the **Display Filter** description at the top of the workfile window.

Changing the Display Order

You may change the order in which objects in the workfile are displayed by selecting View/ **Display Order...** or clicking on the **Order** in the upper right hand portion of the workfile window.

EViews will display the **Object Order** dialog, prompting you for the attribute you wish to use for sorting, and whether to use ascending or descending order.
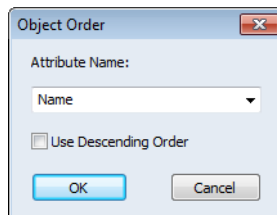
There is a long is of attributes that you may use to determine order, including, among others, Name, Last Update, Source, Description:
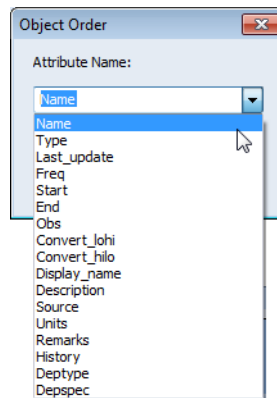
Workfile Summary View

In place of the directory display, you can display a summary view for your workfile. If you select this view, the display will change to provide a description of the current workfile structure, along with a list of the types and numbers of the various objects in each of the pages of the workfile.

To select the summary view, click on **View/Statistics** in the main workfile menu or toolbar. Here we see the display for a first page of a two page workfile.

To return to the directory display view, select **View/Workfile Directory**.

Saving a Workfile

You should name and save your workfile for future use. Push the **Save** button on the work-file toolbar to save a copy of the workfile on disk. You can also save a file using the **File/Save As…** or **File/Save…** choices from the main menu. EViews will display the Windows common file dialog.
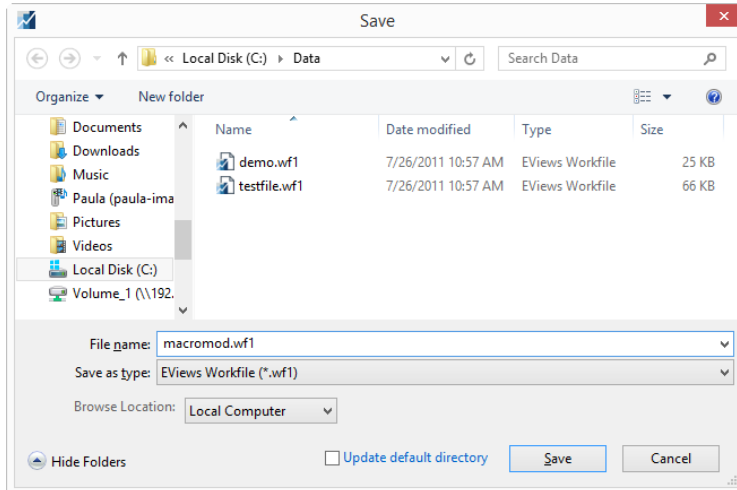
You can specify the target directory in the upper file menu labeled **Save in**. You can navigate between directories in the standard Windows fashion—click once on the down arrow to access a directory tree; double clicking on a directory name in the display area gives you a list of all the files and subdirectories in that directory. Once you have worked your way to the right directory, type the name you want to give the workfile in the **File name** field and push the **Save** button.

Alternatively, you could just type the full Windows path information and name in the **File name** edit field.

In most cases, you will save your data as an EViews workfile. By default, EViews will save your data in this format, using the specified name and the extension ".WF1". You may, of course, choose to save the data in your workfile in a foreign data format by selecting a different format in the dropdown menu. We explore the subject of saving foreign formats below in "Exporting from a Workfile" on page 315.

Saving Updated Workfiles

You may save modified or updated versions of your named workfile using the **Save** button on the workfile toolbar, or by selecting **File/Save…** from the main menu. Selecting **Save** will update the existing workfile stored on disk. You may also use **File/Save As…** to save the workfile with a new name. If the file you save to already exists, EViews will ask you whether you want to update the version on disk.

When you overwrite a workfile on disk, EViews will usually keep a backup copy of the over-written file. The backup copy will have the same name as the file, but with the first character in the extension changed to ~ . For example, if you have a workfile named "Mydata.WF1", the backup file will be named "Mydata.~ f1". The existence of these backup

files will prove useful if you accidentally overwrite or delete the current version of the workfile file, or if the current version becomes damaged.

If you wish to turn on or off the creation of these backup copies you should set the desired global options by selecting **Options/Workfile Storage Defaults...**, and selecting the desired settings.

Workfile Save Options

By default, when you click on the **Save** button, EViews will display a dialog showing the current global default options for storing the data in your workfile.

Your first choice is whether to save your series data in either **Single precision** or **Double precision**. Single precision will create smaller files on disk, but saves the data with fewer digits of accuracy (7 versus 16).
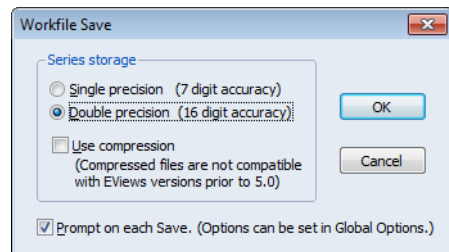
You may also choose to save your data in compressed or non-compressed form. If you select **Use compression**, EViews will analyze the contents of your series, choose an optimal (lossless) storage precision for each series, and will apply compression algorithms, all to reduce the size of the workfile on disk. The storage savings may be considerable, especially for large datasets containing lots of integer and 0, 1 variables. We caution however, that a compressed workfile is not backward compatible, and will not be readable by versions prior to EViews 5.

There is also a checkbox for showing the options dialog on each save operation. By default, the dialog will be displayed every time you save a workfile. Unchecking the **Prompt on each Save** option instructs EViews to hide this dialog on subsequent saves. If you later wish to change the save settings or wish to display the dialog on saves, you must update your global settings by selecting **Options/General Options...** from the main EViews menu and going to the **Data storage/Workfile save** portion of the dialog.

Note that, with the exception of compressed workfiles, workfiles saved in EViews 9 may be read by previous versions of EViews. Objects with features that are not supported by early versions will, however, be dropped when read by those versions of EViews. You should take great caution when saving workfiles using older versions of EViews as you will lose any deleted objects).

Note also that only the first page of a multi-page workfile will be read by versions prior to EViews 5; all other pages will be dropped. You may save individual pages of a multi-page workfile to separate workfiles so that they may be read by versions that do not support multiple pages; see "Saving a Workfile Page" on page 91.

Snapshot Backups

EViews 10 supports the snapshots backup feature. While working with a workfile, or editing an EViews program, the snapshot system will automatically take periodic backups of your changes and save them to a separate snapshot file. To review your changes, you can then bring up the list of available snapshots and compare your current data or program to each snapshot. This allows you to have a history of your changes and potentially restore any previous versions of your data or program before you save your final changes to the original file.

You can also take manual snapshots, such as before revising or editing a large series. A user-defined label and description is provided to help keep track of these changes.

Snapshots for Workfiles

When you have a workfile open in EViews, click the **Snapshot** button to instruct EViews to save a copy of the workfile at that point in time.

Note: Snapshots will only work for files that have already been saved locally. Unsaved files or files that have been saved to a cloud folder location are excluded from the snapshot system.
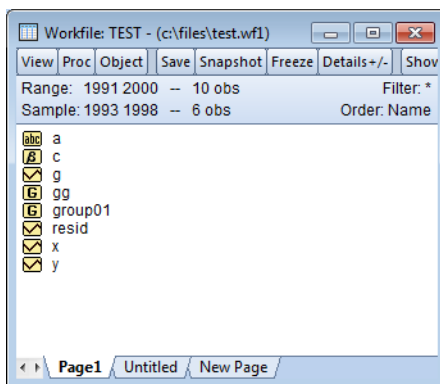
In the **Create Snapshot** dialog, you can specify your own label and description for the snapshot. For workfiles, EViews will also let you include any captured commands that were generated since the last snapshot. All of this information will be saved along with the snapshot.
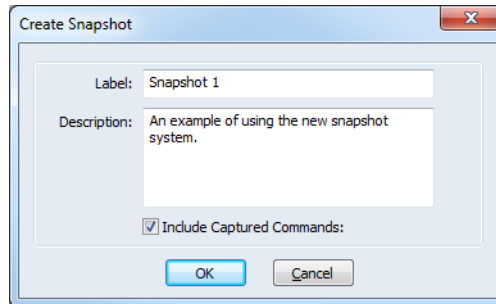
Viewing Snapshots for Workfiles

To view available snapshots for a workfile, click the View button and switch to the **Snapshots** view.
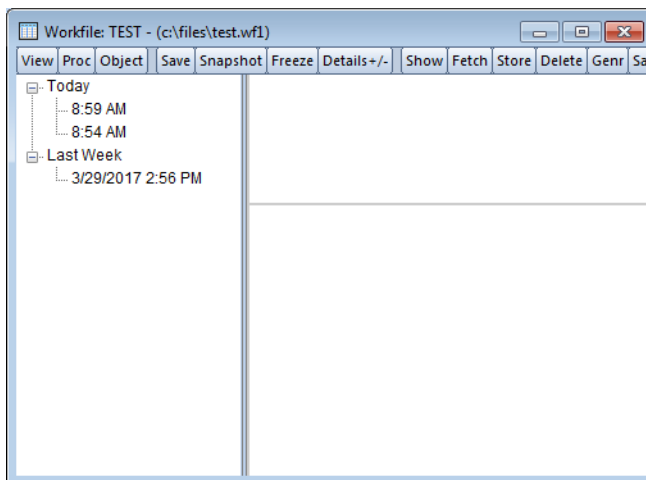
The tree on the left will display all available snapshots with the most recent snapshot at the top. If you click on a snapshot, you can see the metadata and a comparison of objects that have changed between that snapshot's timestamp and the current time.
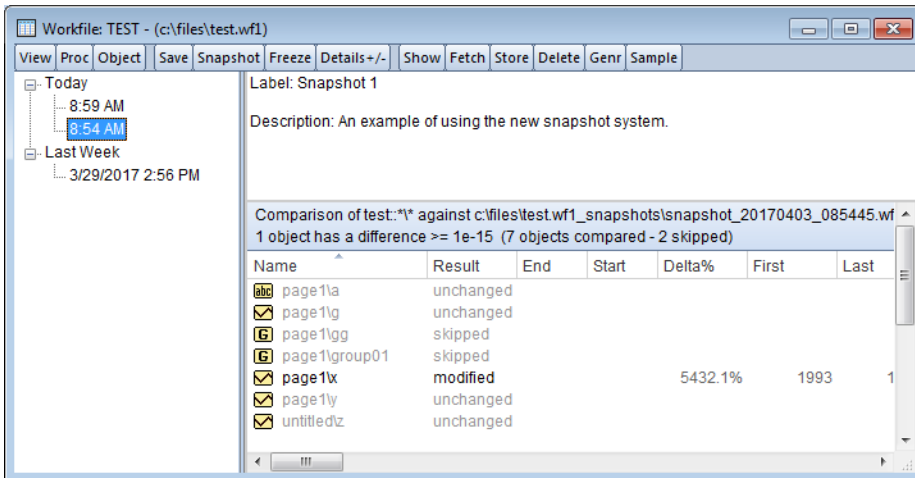
In this example, the comparison view shows that the current X object in our workfile is different from the snapshot that was taken today at 8:54am.

Double-clicking the "page1\x" row will display a new group window that shows the exact differences.

Alternatively, you can also right-click the "8:54 AM" node and select **Open…** (or simply double-click the node) to open the snapshot in its own workfile window.
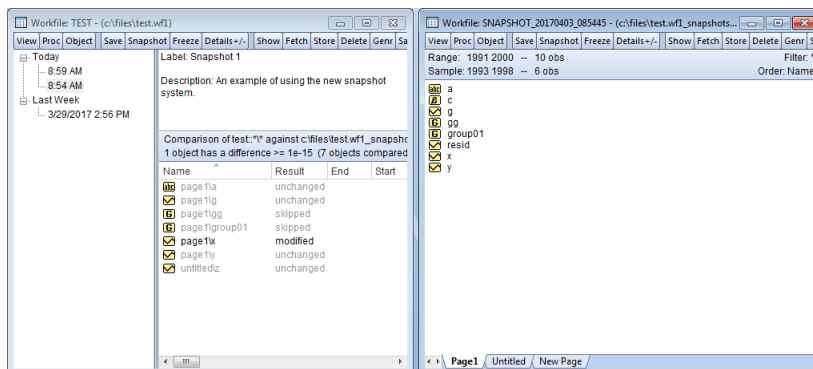
Now with both workfiles open, you can easily inspect differences and copy over individual observations by hand.

If you would simply like to revert the entire workfile to be the same as the selected snapshot, right-click the snapshot node and select **Revert To**. This will load the snapshot as the current workfile in memory. You must still manually save the workfile to overwrite the previous version of the file.

Note that automatic snapshots are suspended when an EViews program is actively running.

See the `wfsnapshot` (p. 546) command for additional information.

Loading a Workfile

You can use **File/Open/EViews Workfile…** to load into memory a previously saved workfile. You may also drag-and-drop the workfile onto your EViews workspace. You will typically save a workfile containing all of your data and results at the end of the day, and later load the workfile to pick up where you left off.

When you select **File/Open/EViews Workfile…** you will see a standard Windows file dialog. Simply navigate to the appropriate directory and double click on the name of the workfile to load it into RAM. The workfile window will open and all of the objects in the workfile will immediately be available.

For convenience, EViews keeps a record of the most recently used files at the bottom of the **File** menu. Select an entry and it will be opened in EViews.

The current version of EViews can read workfiles from all previous versions of EViews. Due to changes in the program, however, some objects may be modified when they are read into the current version.

Multi-page Workfiles

While a great many of your workfiles will probably contain a single page, you may find it useful to organize your data into multiple workfile pages. Multi-page workfiles are primarily designed for situations in which you must work with multiple datasets.

For example, you may have both quarterly and monthly data that you wish to analyze. The multi-page workfile allows you to hold both sets of data in their native frequency, and to perform automatic frequency conversion as necessary. Organizing your data in this fashion allows you to switch instantly between performing your analysis at the monthly and the quarterly level.

Likewise, you may have a panel dataset on individuals that you wish to use along with a cross-sectional dataset on state level variables. By creating a workfile with a separate page for the individual level data, and a separate page for the state level data, you can move back and forth between the individual and the state level analyses, or you can link data between the two to perform dynamic match merging.

Creating a Workfile Page

There are several ways to create a new workfile page.

Creating a Page by Describing its Structure

First, you may describe the structure of the workfile page. This method follows the approach outlined in "Creating a Workfile by Describing its Structure" on page 43. Simply call up the new page menu by clicking on the tab labeled **New Page** and selecting **Specify by Frequency/Range...**, and EViews will display the familiar **Workfile Create** dialog. Simply describe the structure of your workfile page as you would for a new workfile, and enter **OK**.
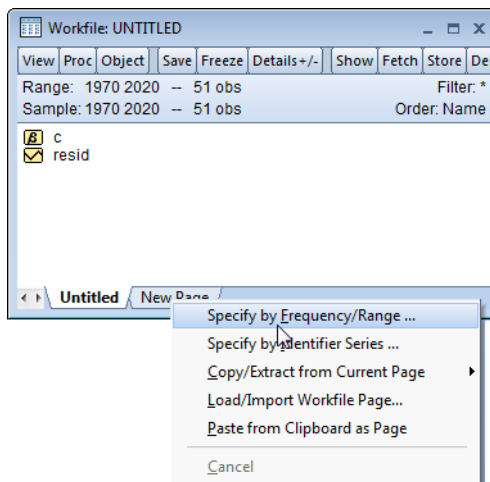
EViews will create a new workfile page with the specified structure and the new page will be given a default name and designated as the active workfile page. The default name will be constructed from the next available name for the given workfile structure. For example, if you create a regular frequency annual page, EViews will attempt to name the page ANNUAL, ANNUAL1, and so forth. The active page is noted visually by the tab selection at the bottom of the workfile window. With the exception of a few page-specific operations, you may generally treat the active page as if it were a standard workfile.

Creating a Workfile Page Using Identifiers

The second approach creates a new page using the unique values of one or more identifier series. Click on the **New Page** tab and select **Specify by Identifier Series...** EViews will open a dialog for creating a new page using one or more identifier series.

At the top of the dialog is a dropdown menu labeled **Method** that you may use to select between the various ways of using identifiers to specify a new page. You may choose between creating the page using:

```
Unique values of ID series from one page
Union of common ID series from multiple pages
Intersection of common ID series from multiple pages
Cross of two non-date ID series
Cross of one date and one non-date ID series
Cross of ID series with a date range
```

(1) the unique ID values from the current workfile page, (2) the union of unique ID values from multiple pages, (3) the intersection of unique ID values from multiple pages, (4) and (5) the cross of the unique values of two ID series, (6) the cross of a single ID series with a date range.

As you change the selected method, the dialog will change to provide you with different options for specifying identifiers.

Unique values of ID series from one page

The easiest way to create a new page from identifiers is to use the unique values in one or more series in the current workfile page.

If you select **Unique values of ID series from one page** in the **Method** dropdown, EViews will prompt you for one or more identifier series which you should enter in the **Cross-section ID series** and **Date series** edit fields.

EViews will take the set of series and will identify the unique values in the specified **Sample**. Note that when multiple identifiers are specified, the unique values are defined over the values in the set of ID series, not over each individual series.

The new page will contain identifier series containing the unique values, and EViews will structure the workfile using this information. If **Date ID** series were provided in the original dialog, EViews will restructure the result as a dated workfile page.

Suppose, for example, that we begin with a workfile page named UNDATED that contains 471 observations on 157 firms observed for 3 years. There is a series FCODE identifying the firm, and a series YEAR representing the year.

We first wish to create a new workfile page containing 157 observations representing the unique values of FCODE. Simply enter FCODE in the **Cross-section ID series,** set the sample to "@ALL", name the new page "UNDATED1", and click on **OK.**

EViews will create a new structured (undated - with identifier series) workfile page UNDATED1 containing 157 observations. The new page will contain a series FCODE with the 157 unique values found in the original series FCODE, and the workfile will be structured using this series.

Similarly, we may choose to create a new page using the series YEAR, which identifies the year that the firm was observed. There are three distinct values for YEAR in the original workfile page ("1987," "1988," "1989"). Click on the Click on the **New Page** tab and select **Specify by Identifier Series...** from the menu, and **Unique values of ID series from one page** in the **Method** dropdown. Enter "YEAR" in the **Date ID** series field, and click on **OK** to create a new annual page with range 1987–1989. Note that EViews will structures the result as a dated workfile page.

Union of common ID series from multiple pages

In some cases, you may wish to create your new page using unique ID values taken from more than one workfile page.

If you select **Union of common ID series from multiple pages**, EViews will find, for each source page, a set of unique ID values, and will create the new workfile page using the union of these values. Simply enter the list of identifiers in the **Cross-section ID series** and **Date series** and edit fields, and a list of pages in which the common identifiers may be found. When you click on **OK**, EViews will first make certain that each of the identifier series is found in each page, then will create the new workfile page using the union of the observed ID values.

We may extend our earlier example where there are three distinct values for YEAR in the original page ("1987," "1988," "1989"). To make things more interesting, suppose there is a another page in the workfile, ANNUAL5, containing annual data for the years 1985–1988 and that this page contains also contains a series YEAR with those values ("1985," "1986," "1987," "1988").

Since we want to exploit the fact that YEAR contains date information, we create a page using the union of IDs by selecting **Union of common ID series from multiple pages**, entering YEAR in the **Date series** field, and then entering "UNDATED" and "ANNUAL5" in the page field. When you click on **OK**, EViews will create a 5 observation, regular annual frequency workfile page for 1987–1989, formed by taking the union of the unique values in the YEAR series in the UNDATED panel page, and the YEAR series in the ANNUAL5 page.

Intersection of common ID series from multiple pages

In other cases, you may wish to create your new page using common unique ID values taken from more than one workfile page. If you select **Intersection of common ID series from multiple pages**, EViews will take the specified set of series and will identify the unique values in the specified **Sample**. The intersection of these sets of unique values across the pages will then be used to create a new workfile page.

In our extended YEAR example, we have two pages: UNDATED, with 471 observations and 3 distinct YEAR values ("1987," "1988," and "1989"); and the ANNUAL4 workfile page containing annual data for four years from 1985–1988, with corresponding values for the series YEAR.

Suppose that we enter YEAR in the **Date ID** field, and tell EViews to examine the intersection of values in the **Multiple pages** UNDATED and ANNUAL4. EViews will create a new workfile page containing the intersection of the unique values of the YEAR series across pages ("1987," "1988"). Since YEAR was specified as a date ID, the page will be structured as a dated annual page.

Cross of two ID series

There are two choices if you wish to create a page by taking the cross of the unique values from two ID series: **Cross of two non-date ID series** creates an undated panel page using the unique values of the two identifiers, while **Cross of one date and one non-date ID series** uses the additional specification of a date ID to allow for the structuring of a dated panel page.

Suppose for example, that you wish to create a page by crossing the 187 unique FCODE values in the UNDATED page with the 4 unique YEAR values in the ANNUAL4 page ("1985," "1986," "1987," "1988"). Since the YEAR values may be used to create a dated panel, we select **Cross of one date and one non-date ID** from our **Method** dropdown.

Since we wish to use YEAR to date structure our result, we enter "FCODE" and "UNDATED" in the **Cross ID series** and **Cross page** fields, and we enter "YEAR" and "ANNUAL4" in the **Date ID series** and **Date page** fields.

When you click on **OK**, EViews will create a new page by crossing the unique values of the two ID series. The resulting workfile will be an annual dated panel for 1985–1988, with FCODE as the cross-section identifier.

It is worth noting that had we had entered the same information in the **Cross of two non-date ID** dialog, the result would be an undated panel with two identifier series.

Cross of ID Series with a date range

In our example of crossing a date ID series with a non-date ID, we were fortunate to have an annual page to use in constructing the date ID. In some cases, the dated page may not be immediately available, and will have to be created prior to performing the crossing operation.

In cases where the page is not available, but where we wish to cross our non-date ID series with a regular frequency range, we may skip the intermediate page creation by selecting the **Cross of ID series with a date range** method.

Here, instead of specifying a date ID series and page, we need only specify a page frequency, start, and end dates. In this example, the resulting annual panel page is identical to the page specified by crossing FCODE with the YEAR series from the ANNUAL page.

While specifying a frequency and range is more convenient than specifying a date ID and page, this method is obviously more restrictive since it does not allow for irregular dated data. In these latter cases, you must explicitly specify your date ID series and page.

Creating a Page by Copying from the Current Page

You may also create a new workfile page by copying data from the current page. Click on **New Page** or click on **Proc** in the main workfile menu, and select **Copy/Extract from Current Page** and either **By Link to New Page...** or **By Value to New Page or Workfile...**. EViews will open a dialog prompting you to specify the objects and data that you wish to copy to a new page. See for a complete discussion.

Alternately, you may simply drag-and-drop a group of selected series from the current workfile page onto the **New Page** tab on the bottom of the workfile window.
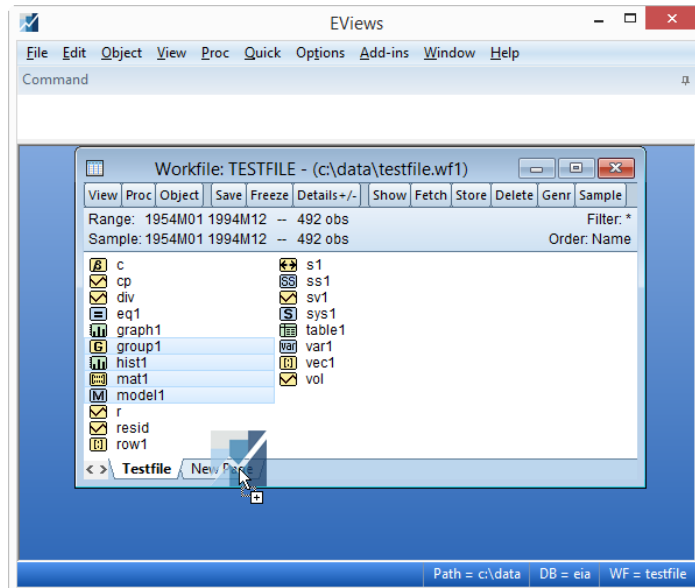
Creating a Page by Loading a Workfile or Data Source

The next method for creating a new page is to load an existing workfile or data source. Call up the new page menu by clicking on **New Page** and selecting **Load Workfile Page...** or by selecting **Proc/Load Workfile Page...** from the main workfile menu. EViews will present you with the **File Open** dialog, prompting you to select your file.

More directly, you may drag an existing workfile, workfile page, or foreign data file directly onto the **New Page** tab on the bottom of your workfile window. You can even copy a page from one workfile to another by dragging the page tab from the source workfile and dropping onto the **New Page** in the other workfile.
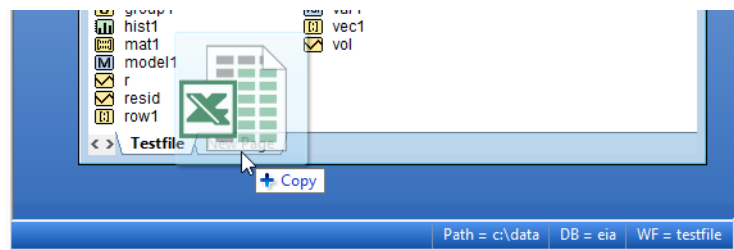
If you select an existing EViews workfile, EViews will add a page corresponding to each page in the source workfile. If you load a workfile with a single page named QUARTERLY, EViews will attempt to load the entire workfile in the new page. If your workfile contains multiple pages, each page of the workfile will be loaded into a new and separate page. The active page will be the newest page.

If you select a foreign data source as described in "Creating a Workfile by Reading from a Foreign Data Source" on page 47, EViews will load the data into a single newly created page in the workfile. This method is exactly the same as that used when creating a new workfile except that the results are placed in a new workfile page.

Creating a Page by Pasting from the Clipboard

You may create a new workfile page by pasting the contents of the Windows Clipboard. This method is particularly useful for copying and pasting data from another application such as Microsoft Word, Excel, or your favorite web browser.

Simply copy the data you wish to use in creating your page, then click on **New Page** and select **Paste from Clipboard as Page**. EViews will first analyze the contents of the clipboard. EViews then creates a page to hold the data and then will read the data into series in the page.

Note that while EViews can correctly analyze a wide range of data representations, the results may not be as expected in more complex settings.

Working With Workfile Pages

While workfile pages may generally be thought of simply as workfiles, there are certain operations that are page-specific or fundamental to multi-page workfiles.

Setting the Active Workfile Page

To select the active workfile page, simply click on the visible tab for the desired page in the workfile window. The active page is noted visually by the tab selection at the bottom of the workfile window.
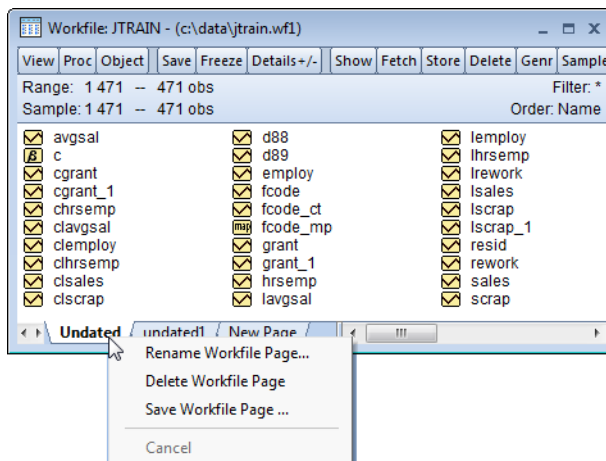
If the desired page is not visible, you may click on the small right and left arrows in the bottom left-hand corner of the workfile window to scroll the page tab display until the desired page is visible, then click on the tab.

You should note that it is possible to hide existing page tabs. If a page appears to be missing, for example if **New Page** is the only visible tab, the remaining tabs are probably hidden. You should click on the left arrow located in the bottom right of the workfile window until your page tabs are visible.

Renaming a Workfile Page

EViews will give your workfile pages a default name corresponding to the workfile structure. You may wish to rename these pages to something more informative. Simply click on the tab for the page you wish to rename and right-mouse-button click to open the workfile page menu. Select **Rename Workfile Page...** from the menu and enter the page name. Alternatively, you may select **Proc/Rename Current Page...** from the main workfile menu to call up the dialog.

Workfile page names must satisfy the same naming restrictions as EViews objects. Notably, the page names must not contain spaces or other delimiters.

Deleting a Workfile Page

To delete a workfile page, right mouse click on the page tab and select **Delete Workfile Page**, or with the page active, click on the **Proc** menu and select **Delete Current Page**.

Saving a Workfile Page

If you wish to save the active workfile page as an individual workfile, click on the page tab, right mouse click to open the workfile page menu and select **Save Workfile Page...** to open the **SaveAs** dialog. Alternatively, you may select **Proc/Save Current Page...** from the main workfile menu to access the dialog.

Saving a page as an individual workfile is quite useful when you wish to load a single page into several workfiles, or if you wish to use the page in a previous version of EViews. Once saved on disk, it is the same as any other single-page EViews workfile.

Reordering Workfile Pages

You may change the order of your workfile pages by dragging the tab for a page at the bottom of the workfile window, and dropping it on top of the tab for the page it should follow. Note the difference in behavior between dragging a tab within a workfile and dragging across workfiles. The first reorders the pages, while the second copies the contents of one page into another.

Working with Series Across Pages

When working with multi-page workfiles you may wish to access the data in one page when performing calculations or analysis on another page. EViews offers three methods for access data across pages; simple copying ("Copying and Pasting" on page 162), creating link objects (Chapter 8. "Series Links," on page 233), and auto-series across pages ("Auto-series Across Pages," on page 196).

Comparing Workfiles

When working with different workfiles, or across different workfile pages, it can be useful to quickly see any differences in the objects contained in the different files/pages. The **View/ Compare...** menu item allows you to compare your current workfile (or more optionally the current page) with a different workfile stored on disk, or a separate page.

Once you click on **View/Compare**, a dialog will appear asking you to select which files to compare, and which objects within those files to compare.
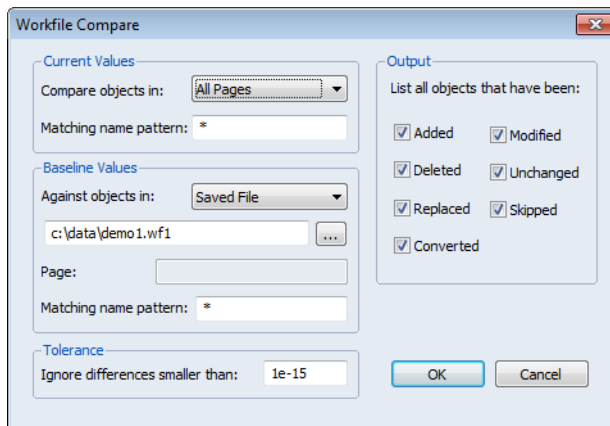
The **Current Values** area lets you select whether you wish to compare objects in all of the pages in the current workfile, or only objects in the current page by specifying the **Compare object in:** dropdown box. You may also choose to compare only a subset of objects in the current workfile or page by using the **Matching name pattern:** field to enter a naming pattern. For example, if you only wish to compare objects whose names begins with "US_", you would enter "US_*" in the box.

You may specify the comparison objects in the **Baseline Values** area. The **Against objects in:** dropdown box lets you specify whether you wish to compare against objects in a separate workfile saved on disk (by selecting **Saved File**), or against objects in the current workfile. If you choose to compare against a file on disk, you should type the path and filename of the file in the edit field below the dropdown. Alternatively, you can click the "**...**" button to select the file using a standard file dialog.

If you chose to compare only objects in the current workfile page in the **Current Values** area, you may select a comparison page to compare against by entering its name in the **Page:** field.

The **Matching name pattern:** field again allows you to choose a subset of objects to compare. Note that you do not need to use the same naming pattern as in the **Current Values**

area. This means you may compare all objects starting with "US_" with all objects starting with "UK_" by entering "UK_*" in this field.

You should note that if you leave the **Page:** field blank and have selected **Current Workfile** in the **Against objects in:** dropdown, EViews will use the current workfile page as the comparison page. Since you have chosen to compare the current page in the current workfile against the current page in the current workfile (i.e. the same page twice), EViews will report no differences between objects, unless you have used different name patterns for **Current Values** and **Baseline Values**.

Note that by default the settings in the **Current Values** and the **Baseline Values** areas are set such that the current workfile is compared against the version of the same workfile currently saved on disk. This is useful when trying to track what changes you have made in your current session (since you last saved to disk), or what has updated when you update links in you workfile.

The **Tolerance** box lets you specify the degree of tolerance to use when deciding if objects have differences. By default, it is set to a very small number (1e-15), which should capture any real differences between objects.

Finally, the **Output** area lets you specify which types of changes in objects to report. Including **Added** items tells EViews to report any objects that exist in the current workfile (or page), but not in the baseline workfile (or page). **Deleted** items are objects that do not exist in the current workfile but do in the baseline. Items that have been **Replaced** are objects that exist in both current and baseline workfiles, but have different object types (for example, it may be an Equation in the current workfile, but a System in the baseline). **Converted** objects are series objects which have different frequencies.

Modified objects are objects that exist in both current and baseline workfiles, are the same type and frequency, yet have different values for at least one observation. **Unchanged** objects are objects that are identical in both. Finally, **Skipped** objects are objects that exist in both workfiles and have the same type, but cannot have their values compared. Equations, for example, cannot be compared between workfiles, and so they are skipped.

As an example, we compare two workfiles, Demo1.wf1 and Demo2.wf1. The results are show below:

The top of the output gives a summary of the comparison. Here demo2 (the current work-file) was compared against demo1.wf1. Eleven objects were compared, with four being skipped and four containing differences.

The main table of the output shows the specifics of the comparison. The first object containing differences is EQ02. This object shows "deleted", meaning that it was present in Demo1 (the baseline workfile), but not in Demo2. Conversely, the graph object, G1, is shown as "added", meaning it is present in Demo2 but not Demo1.

The series GDP is showing as "modified". That means that at least one observation contains different values. In fact, looking at the **Revised** column of the table we can see that only a single observation contains differences. The **First** and **Last** columns show the first and last dates that contain differences. Since we only have a single observation that is different, in this case they show the same date, 1953Q1. The **Delta%** column indicates that the difference between the value in Demo1 and Demo2 is only 0.6%.

Finally, the series M1 shows "modified". It does not have an entry in the **Revised** column, but has an entry in the **End** column instead. This indicates that the series M1 has more observations of data in Demo2 than it did in Demo1. Indeed, the **End** column indicates that there are four extra periods of observations and the end of the workfile. Note that an entry in the **Start** column would have indicated that there were more observations with data at the start of the workfile for that particular series.

Addendum

File Dialog Features

There are additional features in the file open and save dialogs which you may find useful.

Set Default Directory

All EViews file dialogs begin with a display of the contents of the *default directory*. You can always identify the default directory from the listing on the EViews status line. The default directory is set initially to be the directory containing the EViews program, but it can be changed at any time.

You can change the default directory by using the **File/Open…** or the **File/Save As…** menu items, navigating to the new directory, and checking the **Update Default Directory** box in the dialog. If you then open or save a workfile, the default directory will change to the one you have selected. The default directory may also be set from the **Options/File locations…** dialog. See "File Locations" on page 877 of *User's Guide I*.

An alternative method for changing the default EViews directory is to use the `cd` command. Simply enter "CD" followed by the directory name in the command window (see `cd` (p. 310) for details).

File Operations

Since EViews uses a variant of the Windows common file dialog for all open and save operations, you may use the dialog to perform routine file operations such as renaming, copying, moving, and deleting files.
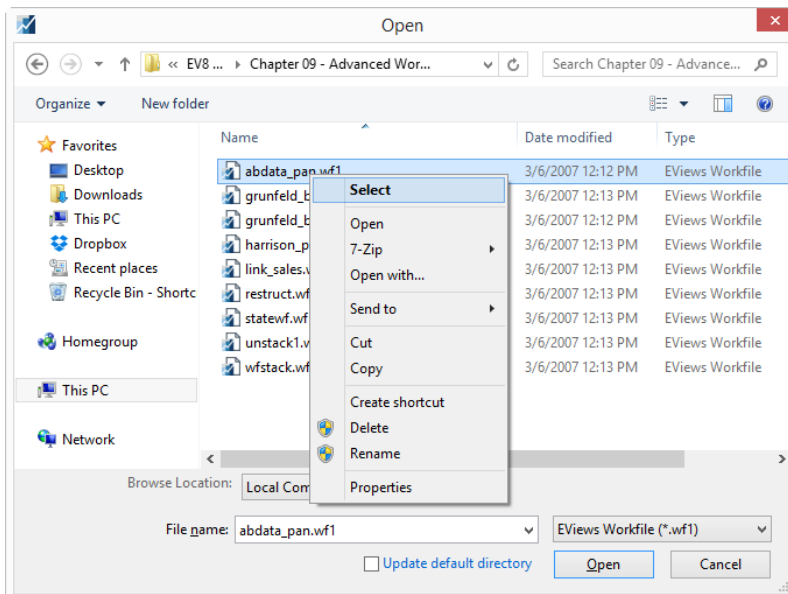
For example, to delete a file, click once of the file name to select the file, then right click once to call up the menu, and select **Delete**. Likewise, you may select a file, right-mouse click, and perform various file operations such as **Copy** or **Rename**.

Cloud Drive Support

You may access files on your cloud drive from within EViews. There is a combo on the **File Open…** and **File/Save As…** dialogs that allows you to configure access to your Box, Dropbox, Google Drive, or OneDrive and to read and write files directly from your cloud drives.

Additionally, many of our commands that take a path to a file can now be used with a cloud location. For example:

```
cd "<myonedrive>:\temp"
wfopen "<mydropboxdrive>:\folder\main.wf1"
wfsave "<mygoogledrive>:\EViews\test.wf1"
```

To access files in your account you must first configure a cloud location within EViews. Click on **File Open…** and then select **New Location** from the **Browse Location** dropdown menu.
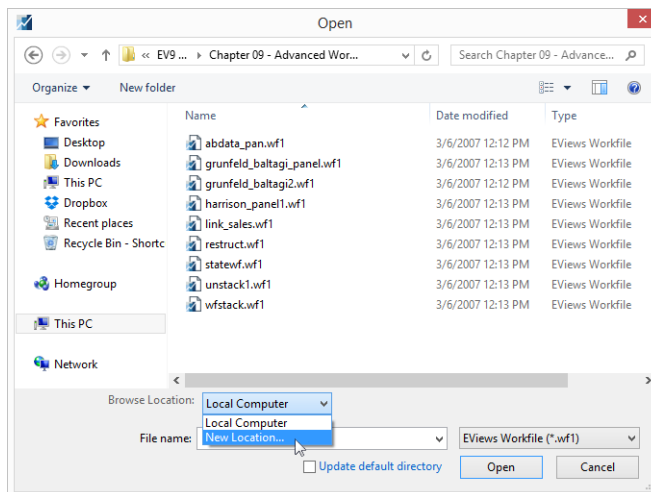
EViews will display a New Location dialog prompting you to select a service provider and to provide a name for use when referring to the location:
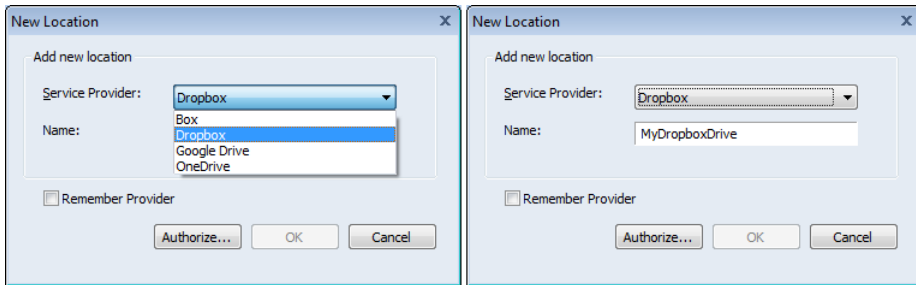
Click the **Authorize** button to start the sign-in and authorization process. The authorization procedure will differ for each provider, but will generally involve launching a browser window and opening the appropriate sign-in page (if are already signed in to your cloud account, your provider may not require this additional authorization).
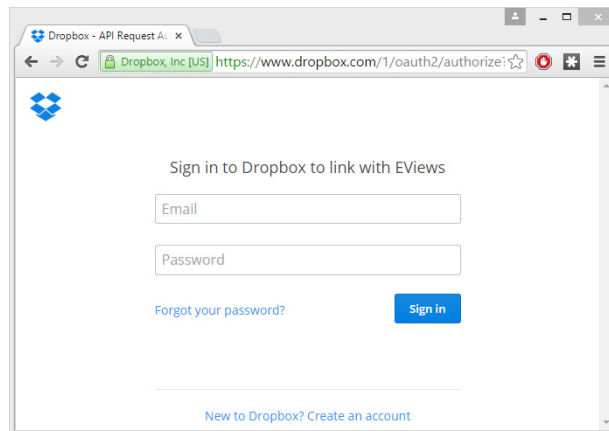
Once signed in, you will be prompted to allow EViews access to your files and folders.
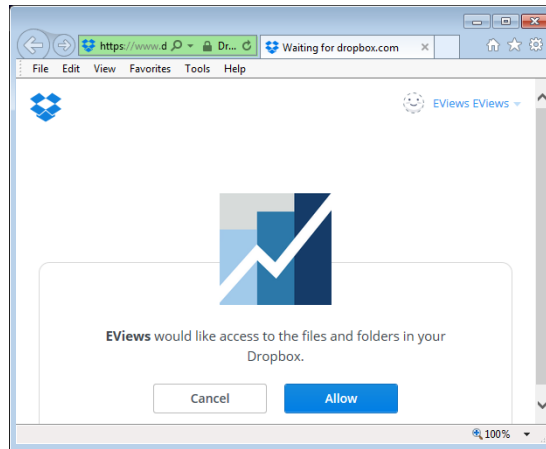
Click the **Allow** button to authorize EViews access. Once authorization has been granted, you will see a message saying you can now close the browser window and return to the EViews **New Location** dialog.
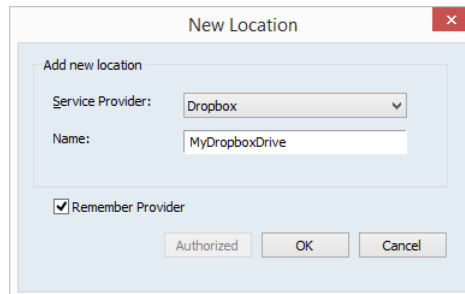
Notice that the **OK** button is now enabled, indicating that you may save the location. The **Remember Provider** checkbox allows you to control the lifetime of this cloud location. If you wish to use the new location only once (such as when you are using a public/lab computer) then leave the **Remember Provider** checkbox unchecked. This will ensure the location is only available to the current instance of EViews. Once EViews is shutdown, the cloud location will no longer be available. However, if you are using a private computer and would like to access this cloud location in the future, you may select this option so that EViews will remember this cloud location for as long as the cloud provider allows (or until you instruct EViews to forget).

Security Note: Cloud location information is stored on the local machine in encrypted format and never contains any user passwords.Stored locations are only available to the current Windows user. Some cloud providers will always require re-authorization after a certain period of time, regardless of activity.

Once the location has been created, the **File Open…** dialog will be displayed, with the **Browse Location** dropdown set to the newly created location, so that you may examine the contents of your cloud drive.
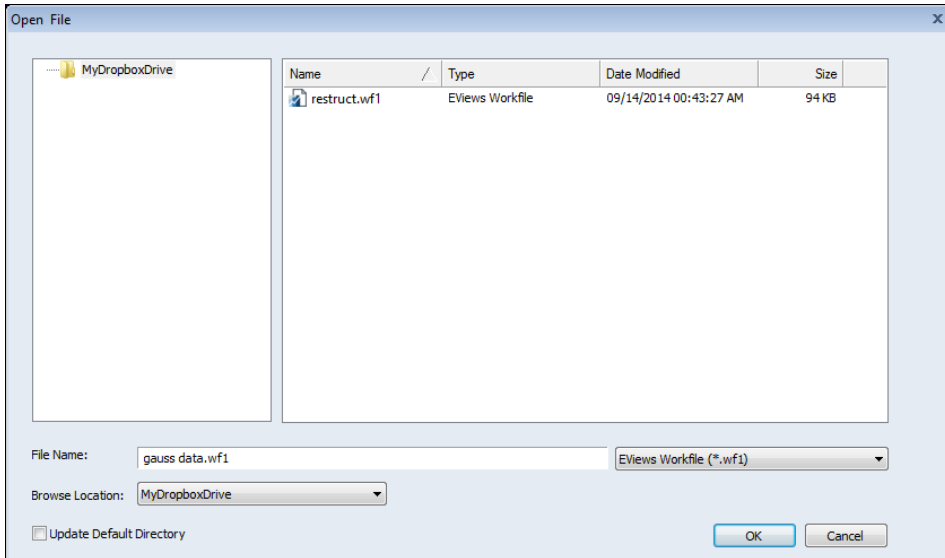
You can create, edit, and delete cloud locations using the **Cloud Management** dialog. To display the dialog, select **Options/Cloud Management** from the main EViews menu:
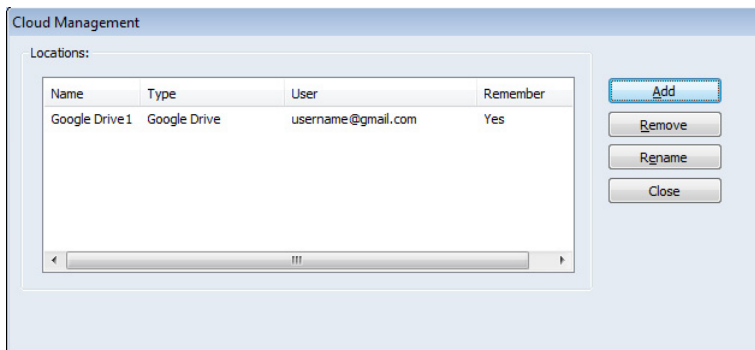
From here you can see relevant information such as the local name of the location, the provider, the user account associated with the location, and whether or not the location has been saved.

Cloud FAQ

- Which EViews files can be opened/saved to a cloud location?

Most file formats are supported, including EViews Workfiles, Programs, Text Files, Excel Files, *etc.* The one notable exception to that are EViews Database files (*.EDB).

- I'm concerned about my cloud account security. Is my account password saved locally?

  No. EViews does not have access to your account password. Authorization is negotiated between you and your provider via your browser. The cloud provider sends EViews a special authorization key and EViews keeps this special key in encrypted format. All authorizations can be revoked by the provider whenever you request, forcing cloud clients such as EViews to obtain new authorization for continued access to your files.

- Are cloud files cached locally? If so, where?

  When a file is opened or saved to a cloud location, all files are cached locally to the following directory:

  < *EViews Temp File Path* > \ < *location name* >. Cached files will be automatically removed after the end of the session if your cloud location was not defined using the **Remember Provider** setting.

Chapter 4. Object Basics

At the heart of the EViews design is the concept of an object. In brief, objects are collections of related information and operations that are bundled together into an easy-to-use unit. Virtually all of your work in EViews will involve using and manipulating various objects.

EViews holds all of its objects in object containers. You can think of object containers as filing cabinets or organizers for the various objects with which you are working. The most important object container in EViews is the workfile, which is described in Chapter 3. "Workfile Basics," beginning on page 41.

The remainder of this chapter describes basic techniques for working with objects in a workfile. While you may at first find the idea of objects to be a bit foreign, the basic concepts are easy to master and will form the foundation for your work in EViews. But don't feel that you have to understand all of the concepts the first time through. If you wish, you can begin working with EViews immediately, developing an intuitive understanding of objects and workfiles as you go.

Subsequent chapters will provide a more detailed description of working with the various types of objects and other types of object containers.

Note that the current discussion focuses on interactive methods for working with objects. If you feel more comfortable using commands, Chapter 1. "Object and Command Basics," beginning on page 3 of the *Command and Programming Reference*, offers command equivalents for the operations described in this chapter.

What is an Object?

Information in EViews is stored in *objects*. Each object consists of a collection of information related to a particular area of analysis. For example, a *series object* is a collection of information related to a set of observations on a particular variable. An *equation object* is a collection of information related to the relationship between a collection of variables.

Note that an object need not contain only one type of information. For example, an estimated equation object contains not only the coefficients obtained from estimation of the equation, but also a description of the specification, the variance-covariance matrix of the coefficient estimates, and a variety of statistics associated with the estimates.

Associated with each type of object is a set of views and procedures which can be used with the information contained in the object. This association of views and procedures with the type of data contained in the object is what we term the *object oriented* design of EViews.

The object oriented design simplifies your work in EViews by organizing information as you work. For example, since an equation object contains all of the information relevant to an

estimated relationship, you can move freely between a variety of equation specifications simply by working with different equation objects. You can examine results, perform hypothesis and specification tests, or generate forecasts at any time. Managing your work is simplified since only a single object is used to work with an entire collection of data and results.

This brief discussion provides only the barest introduction to the use of objects. The remainder of this section will provide a more general description of EViews objects. Subsequent chapters will discuss series, equations, and other object types in considerable detail.

Object Data

Each object contains various types of information. For example, series, matrix, vector, and scalar objects, all contain mostly numeric information. In contrast, equations and systems contain complete information about the specification of the equation or system, and the estimation results, as well as references to the underlying data used to construct the estimates. Graphs and tables contain numeric, text, and formatting information.

Since objects contain various kinds of data, you will want to work with different objects in different ways. For example, you might wish to compute summary statistics for the observations in a series, or you may want to perform forecasts based upon the results of an equation. EViews understands these differences and provides you with custom tools, called views and procedures, for working with an object's data.

Object Views

There is more than one way to examine the data in an object. Views are tabular and graphical windows that provide various ways of looking at the data in an object.

For example, a series object has a spreadsheet view, which shows the raw data, a line graph view, a bar graph view, a histogram-and-statistics view, and a correlogram view. Other views of a series include distributional plots, QQ-plots, and kernel density plots. Series views also allow you to compute simple hypothesis tests and statistics for various subgroups of your sample.

An equation object has a representation view showing the equation specification, an output view containing estimation results, an actual-fitted-residual view containing plots of fitted values and residuals, a covariance view containing the estimated coefficient covariance matrix, and various views for specification and parameter tests.

Views of an object are displayed in the object's window. Only one window can be opened for each object and each window displays only a single view of the object at a time. You can change views of an object using the **View** menu located in the object window's toolbar or the EViews main menu.

Perhaps the most important thing to remember about views is that views normally do not change data outside the object. Indeed, in most cases, changing views only changes the display format for the data, and not the data in the object itself.

Object Procedures

Most EViews objects also have *procedures*, or *procs*. Like views, procedures often display tables or graphs in the object's window. Unlike views, however, procedures alter data, either in the object itself or in another object.

Many procedures create new objects. For example, a series object contains procedures for smoothing or seasonally adjusting time series data and creating a new series containing the smoothed or adjusted data. Equation objects contain procedures for generating new series containing the residuals, fitted values, or forecasts from the estimated equation.

You select procedures from the **Proc** menu on the object's toolbar or from the EViews main menu.

Object Types

The most common objects in EViews are series and equation objects. There are, however, a number of different types of objects, each of which serves a unique function. Most objects are represented by a unique icon which is displayed in the object container (workfile or database) window. The basic object icons are given by:

Alpha	Matrix	Spool	Text
Coef	Model	Sspace	Userobj
Equation	Pool	String	Valmap
Factor	Rowvector	Svector	Var
Graph	Sample	Sym	Vector
Group	Scalar	System	
Logl	Series	Table	

Despite the fact that they are also objects, object containers do not have icons since they cannot be placed in other object containers—thus, workfiles and databases do not have icons since they cannot be placed in other workfiles or databases.

Note also that there are special icons that correspond to special versions of the objects:

 Auto-updating Series

 Group data and definitions (in databases)

 Undefined Link

If you set a series object to be auto-updating (see "Auto-Updating Series" on page 203), EViews will use the special icon to indicate that the series depends upon a formula. In contrast, an auto-updating alpha series (which we imagine to be less common) uses the original alpha icon, with an orange color to indicate the presence of a formula.

When group data are stored in databases, you will be given the option of storing the group definition (list of series names) alone, or both the group definition and the series contained in the group (see "Store, Fetch, and Copy of Group Objects" on page 328). If the latter are stored, the standard group icon will be modified, with the " + " indicating the additional presence of the series data.

Lastly, a link object (see "Series Links" on page 233), is always in one of three states, depending upon the definition contained in the link. If the link is to a numeric source series, the link object is displayed using a series icon, since it may be used as though it were an ordinary series, with a distinctive pink color used to indicate that the object depends on linked data. If the link is to an alpha source series, the link will show up as an alpha series icon, again in pink. If, however, the link object is unable to locate the source series, EViews will display the "?" icon indicating that the series type is unknown.

Basic Object Operations

Creating Objects

To create an object, you must first make certain that you have an open workfile container and that its window is active. Next, select **Object/New Object…** from the main menu. Until you have created or loaded a workfile, this selection is unavailable. After you click on the **Object/New Object…** menu entry, you will see the **New Object** dialog box.

You can click on the type of object you want, optionally provide a name for the object, and then click on **OK**. For some object types, a second dialog box will open prompting you to describe your object in more detail. For most objects, however, the object window will open immediately.

For example, if you select **Equation**, you will see a dialog box prompting you for additional informa-

tion. Alternatively, if you click on **Series** and then select **OK**, you will see an object window (series window) displaying the spreadsheet view of an UNTITLED series.

We will discuss object windows in greater detail in "The Object Window" on page 112.

Objects can also be created by applying procedures to other objects or by freezing an object view (see "Freezing Objects" on page 118).

Selecting Objects

Creating a new object will not always be necessary. Instead, you may want to work with an existing object. One of the fundamental operations in EViews is *selecting* one or more objects from the workfile directory.

The easiest way to select objects is to point-and-click, using the standard Windows conventions for selecting contiguous or multiple items if necessary. Keep in mind that if you are

selecting a large number of items, you may find it useful to use the display filter before beginning to select items.

In addition, the **View** button in the workfile toolbar provides convenient selection shortcuts:

- **Select All** selects all of the objects in the workfile with the exception of the C coefficient vector and the RESID series.

- **Deselect All** eliminates any existing selections.

Note that all of the selected objects will be highlighted.

Opening Objects

Once you have selected your object or objects, you will want to open your selection, or create a new object containing the selected objects. You can do so by double clicking anywhere in the highlighted area.

If you double click on a single selected object, you will open an object window.

If you select multiple graphs or series and double click, a pop-up menu appears, giving you the option of creating and opening new objects (group, equation, VAR, graph) or displaying each of the selected objects in its own window.

(You may also select **View/Open Selected** or right-mouse click and select **Open** to access menus for opening the multiple objects.)

Note that if you select multiple graphs and double click or select **View/Open Selected/One Window**, all of the graphs will be merged into a single graph and displayed in a single window.

Other multiple item selections are not valid, and will either issue an error or will simply not respond when you double click.

When you open an object, EViews will display the current view. In general, the current view of an object is the view that was displayed the last time the object was opened (if an object has never been opened, EViews will use a default view). The exception to this general rule is for those views that require significant computational time. In this latter case, the current view will revert to the default.

Showing Objects

An alternative method of selecting and opening objects is to "show" the item. Click on the **Show** button on the toolbar, or select **Quick/Show…** from the menu and type in the object name or names.

Showing an object works exactly as if you first selected the object or objects, and then opened your selection. If you enter a single object name in the dialog box, EViews will open

the object as if you double clicked on the object name. If you enter multiple names, EViews will always open a single window to display results, creating a new object if necessary.

The **Show** button can also be used to display functions of series, also known as auto-series. All of the rules for auto-series that are outlined in "Database Auto-Series" on page 329 will apply.

Previewing Objects

EViews offers a preview tool for those times when you wish to quickly look through a number of objects. Instead of opening and closing each object, you may use the preview to quickly scroll through a preview of each object, viewing metadata (name, type, description, frequency, last update, source, units, *etc.*) and object type-specific information (for example, series will show observation data and a small graph while equations will show estimation output).

To preview an object you simply select the object in the workfile (or database), and, if necessary, open the preview window. The preview window will display a preview of the contents of the first selected object.

If the preview window is not open you may:

- Press the F9 key.

- Right-click on an object and select **Preview** or **Preview members** (the latter is only be available for previewing group objects in a workfile).

- Select **Window/Preview** from the main menu.
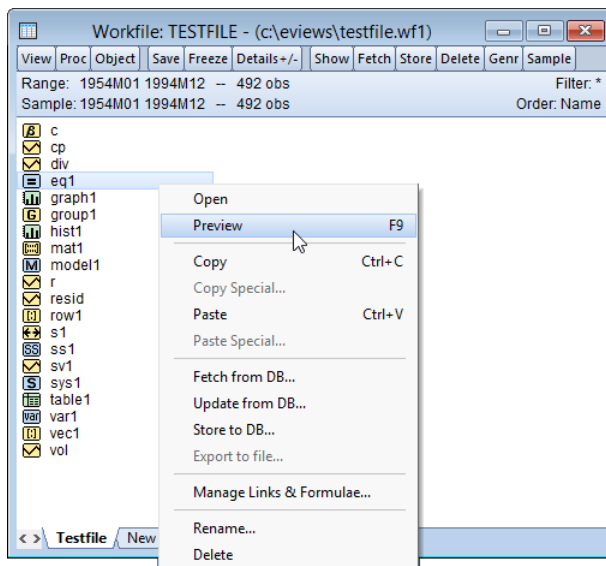
Alternately, to preview an object in a database you can simply double-click on the object in the database window to display the contents of the object in a preview window. (Note that previewing objects from hosted databases may involve a slight delay as the data are retrieved.)

Once opened, you may move the preview window anywhere on your screen, even outside of the EViews application window. A subsequent press of the F9 key will activate and bring to the front the preview window (if it is active) or close the preview window (if the window is active).

The title bar of the preview window shows the type, name and location of the object, and the number of objects in the current preview list (either the workfile or database, or the currently selected subset of objects).

For example, here, we see a series preview window for the series CP which is located in the workfile TESTFILE. CP is the second (alphabetically) of the 20 objects in this workfile.
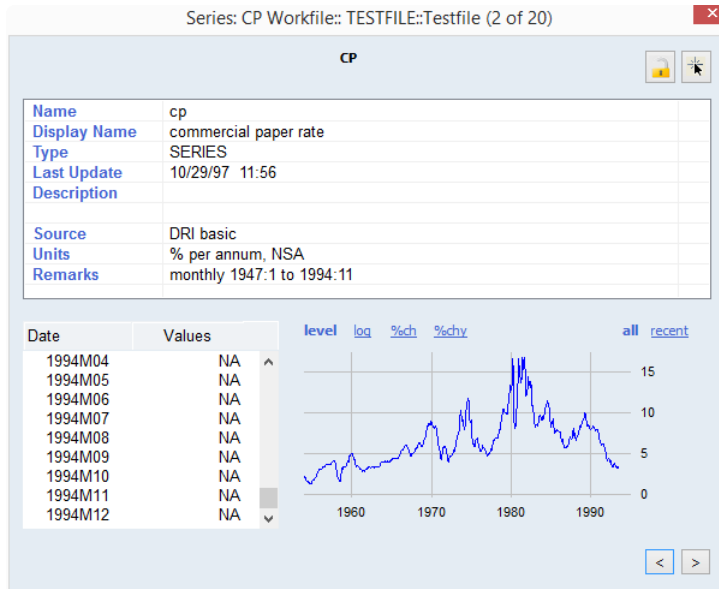
The top part of the display area shows the name of the object and possibly a description, along with metafile information taken from the label view of the object.

The remaining contents of the window will differ depending on the selected object.

For the series object depicted here, the bottom area shows a spreadsheet of the values in the series and a graph of the values in the series. Notice that you can interact with the graph: change the display to show the levels of the series, or simple log, percentage change, and year percentage change transforms of the data; click on **recent** to change the graph to only show recent values of the series or all to show **all** of the values in the series.
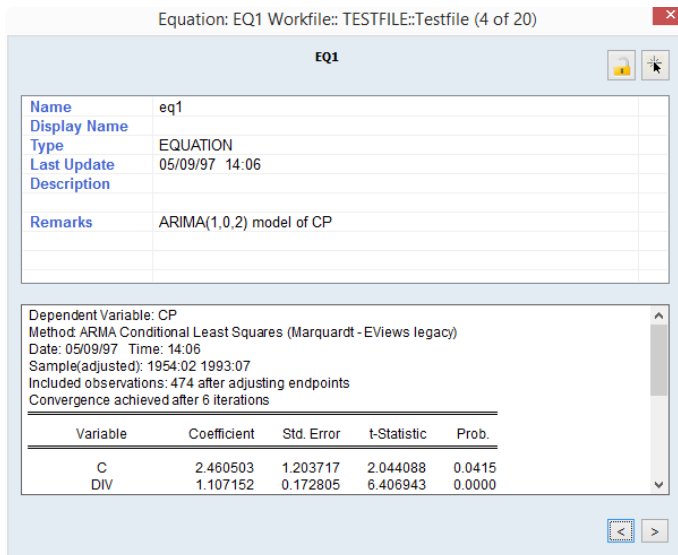
Alternately, the preview window for an equation as seen here displays the estimation output in the bottom portion of the window.

Working with the Preview Window

To the right of the name or description at the top of the preview area are two icons:

- Lock: prevent the contents of the preview window from changing, even if you select a different object. The preview window will remain locked until the lock button is pressed again, therefore unlocking it.

- Find: bring the workfile or database window containing the object to forefront, and if possible, select the object.
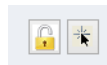
On the bottom right of the window, you will find two navigation buttons that facilitate easy browsing through objects:

- \> : navigate (forward) to the next object in the preview list, cycling around to the first object if necessary.

- \< : navigate (backward) to the previous object in the preview list, cycling around to the last object if necessary.

If you select a single object, the list of objects to browse contains all of the objects in the active database or workfile. Alternately, you can browse through a subset of objects simply by selecting the objects of interest.

For group objects there is an additional navigation button in the lower right corner of the preview window alongside the > and < buttons:

- Next 5 > : navigate forward to the next five members of the group, cycling around to the first object if necessary.

Note that the group preview lists all the members of a group under the Group Members subwindow, but displays the line graphs of only five group members at a time. The members displayed have colored line symbols before their names. If you click on a group member in the list, the selection and graph will change. This member will be the first of the five objects previewed in the graph.
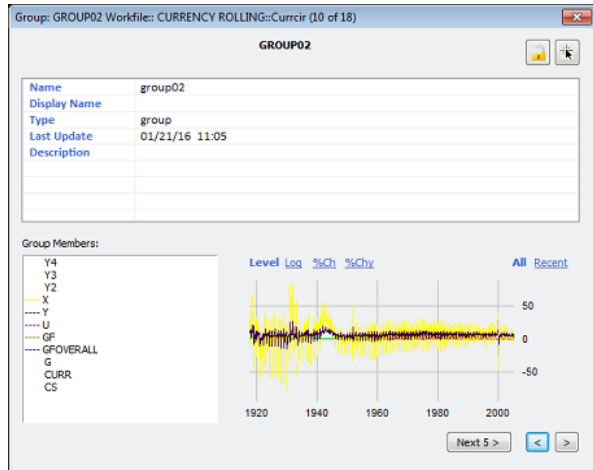
For database objects, in the lower left corner of the preview window will contain an **Export** button.
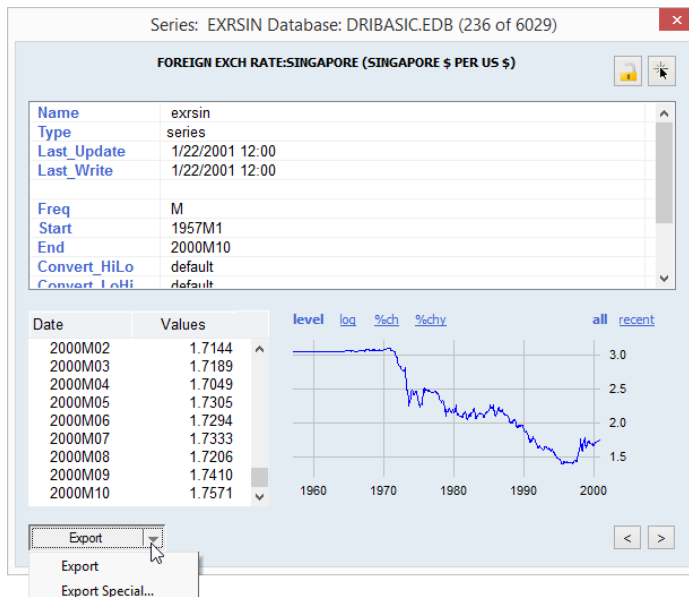
Pressing the **Export** button fetches the object shown in the preview window from the database into the active workfile. If there is no open workfile, the object will be exported to a new untitled workfile.

Clicking on the down arrow on the right side of the **Export** button shows two options: **Export** and **Export Special...**

- **Export** will simply fetch the object into the current workfile as described above.

- **Export Special...** will display a dialog offering various options for the fetch operation.
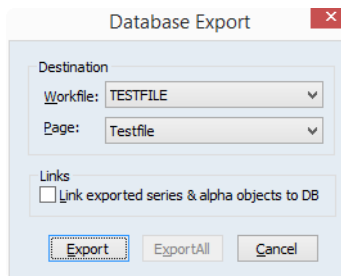
You can control the workfile and page destination, whether to export series and alpha objects as links or by (the default) copying of current values. Clicking on **Export** exports only the currently displayed object. Clicking on **Export All** exports all selected objects.

The Object Window

We have been using the term *object window* somewhat loosely in the previous discussion of the process of creating and opening objects. Object windows are the windows that are displayed when you open an object or object container. An object's window will contain either a view of the object, or the results of an object procedure.

One of the more important features of EViews is that you can display object windows for a number of items at the same time. Managing these object windows is similar to the task of managing pieces of paper on your desk.

Components of the Object Window

Let's look again at a typical object window:
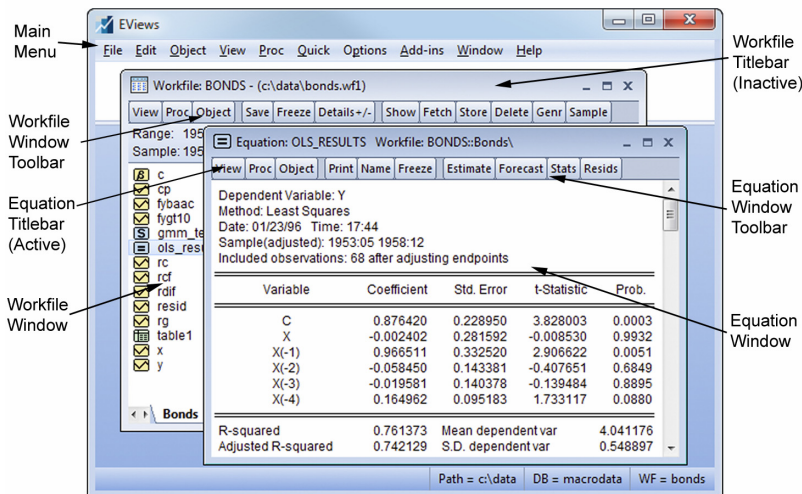
Here, we see the equation window for OLS_RESULTS. First, notice that this is a standard window which can be closed, resized, minimized, maximized, and scrolled both vertically and horizontally. As in other Windows applications, you can make an object window active by clicking once on the titlebar, or anywhere in its window. Making an object window active is equivalent to saying that you want to work with that object. Active windows may be identified by the darkened titlebar.

Second, note that the titlebar of the object window identifies the object type, name, and object container (in this case, the BONDS workfile or the OLS_RESULTS equation). If the object is itself an object container, the container information is replaced by directory information.

Lastly, at the top of the window there is a toolbar containing a number of buttons that provide easy access to frequently used menu items. These toolbars will vary across objects—the series object will have a different toolbar from an equation or a group or a VAR object.

There are several buttons that are found on all object toolbars:

- The **View** button lets you change the view that is displayed in the object window. The available choices will differ, depending upon the object type.

- The **Proc** button provides access to a menu of procedures that are available for the object.

- The **Object** button lets you manage your objects. You can store the object on disk, name, delete, copy, or print the object.

- The **Print** button lets you print the current view of the object (the window contents).

- The **Name** button allows you to name or rename the object.

- The **Freeze** button creates a new object graph, table, or text object out of the current view.

Menus and the Object Toolbar

As we have seen, the toolbar provides a shortcut to frequently accessed menu commands. There are a couple of subtle, but important, points associated with this relationship that deserve special emphasis:

- Since the toolbar simply provides a shortcut to menu items, you can always find the toolbar commands in the menus.

- This fact turns out to be quite useful if your window is not large enough to display all of the buttons on the toolbar. You can either enlarge the window so that all of the buttons are displayed, or you can access the command directly from the menu.

- The toolbar and menu *both* change with the object type. In particular, the contents of the **View** menu and the **Proc** menu will always change to reflect the type of object (series, equation, group, etc.) that is active.

The toolbars and menus differ across objects. For example, the **View** and **Proc** drop-down menus differ for every object type. When the active window is displaying a series window, the menus provide access to series views and series procedures. Alternatively, when the active window is a group window, clicking on **View** or **Proc** in the main menu provides access to the different set of items associated with group objects.
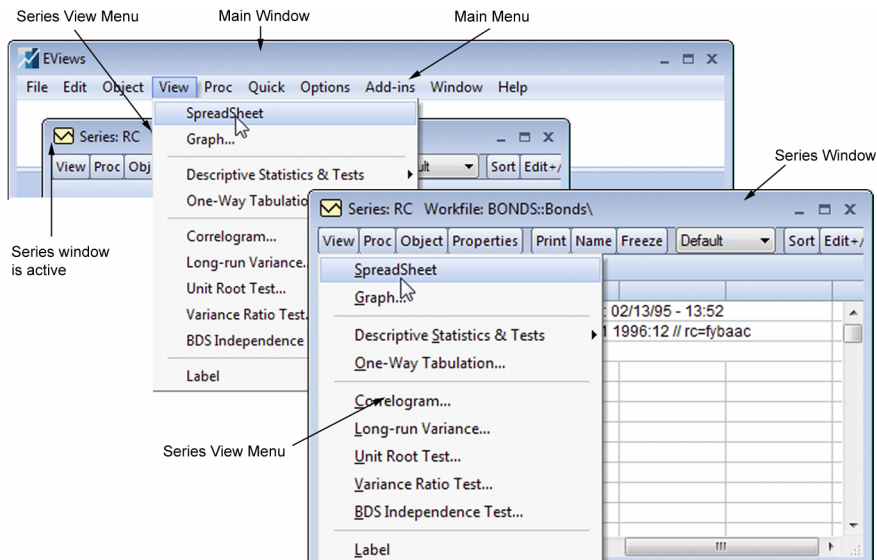
The figure above illustrates the relationship between the **View** toolbar button and the **View** menu when the series window is the active window. In the left side of the illustration, we see a portion of the EViews *main window,* as it appears, after you click on View in the main menu (note that the RC series window is the active window). On the right, we see a depiction of the *series window* as it appears after you click on the **View** button in the series toolbar. Since the two operations are identical, the two drop-down menus are identical.

In contrast to the **View** and **Proc** menus, the **Object** menu does not, in general, vary across objects. An exception occurs, however, when an object container window (*i.e.*, a workfile or database window) is active. In this case, clicking on **Object** in the toolbar, or selecting **Object** from the menu provides access to menu items for manipulating the objects in the container.

Working with Objects

Naming Objects

Objects may be named or unnamed. When you give an object a name, the name will appear in the directory of the workfile, and the object will be saved as part of the workfile when the workfile is saved.

You must name an object if you wish to keep its results. If you do not name an object, it will be called "UNTITLED". Unnamed objects are not saved with the workfile, so they are deleted when the workfile is closed and removed from memory.

To rename an object, first open the object window by double clicking on its icon, or by clicking on **Show** on the workfile toolbar, and entering the object name. Next, click on the **Name** button on the object window, and enter the name (up to 300 characters), and optionally, a display name to be used when labeling the object in tables and graphs. If no display name is provided, EViews will use the object name. *Note: before* EViews 10, *the maximum object name length was 24 characters. You may not be able to correctly open a workfile with object names longer than 24 characters in previous versions of EViews.*
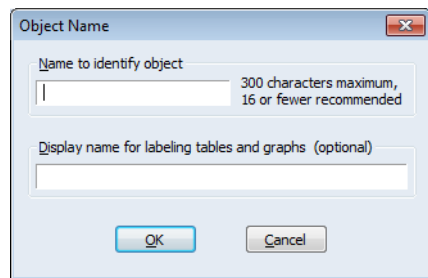
You can also rename an object from the workfile window by selecting **Object/Rename Selected…** and then specifying the new object name. This method saves you from first having to open the object.

The following names are reserved and cannot be used as object names: "ABS", "ACOS", "AND", "AR", "ASIN", "C", "CON", "CNORM", "COEF", "COS", "D", "DLOG", "DNORM",

"ELSE", "ENDIF", "EXP", "LOG", "LOGIT", "LPT1", "LPT2", "MA", "NA", "NOT", "NRND", "OR", "PDL", "RESID", "RND", "SAR", "SIN", "SMA", "SQR", and "THEN".

EViews accepts both capital and lower case letters in the names you give to your series and other objects, but does not distinguish between names based on case. Its messages to you will follow normal capitalization rules. For example, "SALES", "sales", and "sAles" are all the same object in EViews. For the sake of uniformity, we have written all examples of input using names in lower case, but you should feel free to use capital letters instead.

Despite the fact that names are not case sensitive, when you enter text information in an object, such as a plot legend or label information, your capitalization will be preserved.

By default, EViews allows only one untitled object of a given type (one series, one equation, etc.). If you create a new untitled object of an existing type, you will be prompted to name the original object, and if you do not provide one, EViews will replace the original untitled object with the new object. The original object will not be saved. If you prefer, you can instruct EViews to retain all untitled objects during a session but you must still name the ones you want to save with the workfile. See "Window Behavior" on page 861.

Labeling Objects

In addition to the display name described above, EViews objects have label fields where you can provide extended annotation and commentary. To view these fields, select **View/Label** from the object window:

This is the label view of an unmodified object. By default, every time you modify the object, EViews automatically appends the modification in the **Last Update** field.
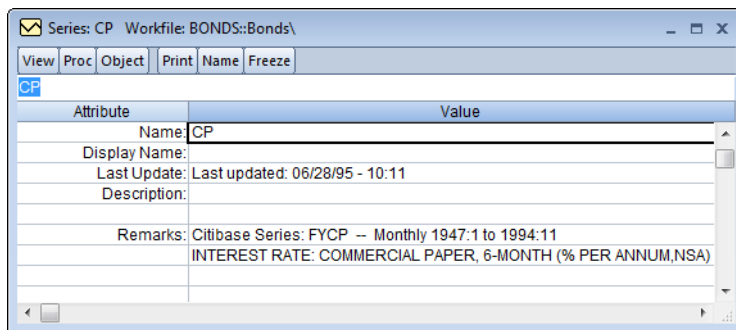
You can edit any of the fields, except **Last Update**. Simply click in the cell that you want to edit. All attributes contain only one line, except the **Remarks** field, which can contain multiple lines. Press ENTER after entering data into the current field to add a new line to this attribute.

These annotated fields are useful when you want to search for an object stored in an EViews database. Any text that is in the fields is searchable in an EViews database; see "Querying the Database" on page 333 for further discussion.

Adding Custom Attributes in the Label View

EViews allows you to add any custom attribute that you'd like to your objects. Click in the empty cell below the last existing attribute, and before the **Remarks** label. You will see the text "Enter attribute name" in the edit field at the top of the window. Enter the name of your new custom attribute, and then move to the cell to the right to enter a value.

In our example, click below **Units**. Enter the name of the new attribute, and type the appropriate value in the field to the right. Here, we have created a custom attribute called **Default Rate**, and entered the value "6.23%". Close the object window to accept the changes.
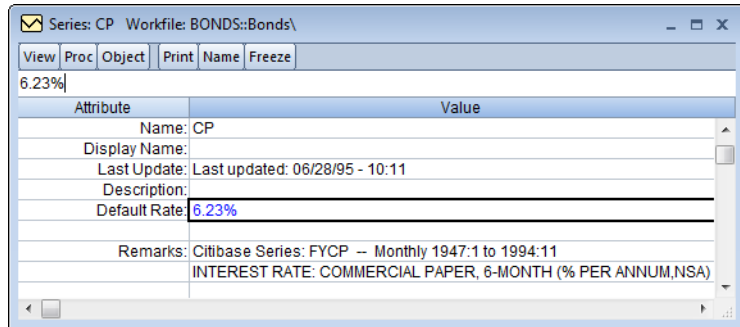
You must type something into the associated field on the right, or the custom attribute will not be saved when the object window is closed. To add an attribute whose data will be filled out in the future, you may want to enter "none" as a placeholder. However, adding custom attributes is easy, either via the label view as just described, or in the workfile window (see "Adding Your Own Label Attributes," beginning on page 65), so this step should not be necessary.

Similarly, you can delete an attribute by removing its value. If we delete "6.23%" in our example and close the window, the **Default Rate** attribute will be removed from the object.

If you are adding custom attributes to many objects, EViews provides tools for doing so in the workfile window. See "Adding Your Own Label Attributes," beginning on page 65. From the workfile **Details** view, you can add attributes across multiple objects, and populate them with values using workfile filtering techniques.

Copying Objects

There are two distinct methods of duplicating the information in an object: copying and freezing.

If you select **Object/Copy** from the menu, EViews will create a new untitled object containing an exact copy of the original object. By exact copy, we mean that the new object duplicates all the features of the original (except for the name). It contains all of the views and procedures of the original object and can be used in future analyses just like the original object.

You may also copy an object from the workfile window. Simply highlight the object and click on **Object/Copy Selected…** or right mouse click and select **Object/Copy…**, then specify the destination name for the object.

We mention here that **Copy** is a very general and powerful operation with many additional features and uses. For example, you can copy objects across both workfiles and databases using wildcards and patterns. See "Copying Objects" on page 325 for details on these additional features.

Copy-and-Pasting Objects

The standard EViews copy command makes a copy of the object in the same workfile. When two workfiles are in memory at the same time, you may copy objects between them using copy-and-paste.

Highlight the objects you wish to copy in the source workfile. Then select **Edit/Copy** from the main menu.

Select the destination workfile by clicking on its titlebar. Then select either **Edit/Paste** or **Edit/Paste Special…** from the main menu or simply **Paste** or **Paste Special…** following a right mouse click.

Edit/Paste will perform the default paste operation. For most objects, this involves simply copying over the entire object and its contents. In other cases, the default paste operation is more involved. For example, when copy-and-pasting series between source and destination workfiles that are of different frequency, frequency conversion will be performed, if possible, using the default series settings (see "Frequency Conversion" on page 170 for additional details). EViews will place named copies of all of the highlighted objects in the destination workfile, prompting you to replace existing objects with the same name.

If you elect to **Paste Special…**, EViews will open a dialog prompting you for any relevant paste options. For example, when pasting series, you may use the dialog to override the default series settings for frequency conversion, to perform special match merging by creating links ("Series Links" on page 233). In other settings, **Paste Special…** will simply prompt you to rename the objects in the destination workfile.

Freezing Objects

The second method of copying information from an object is to *freeze* a view of the object. If you click **Object/Freeze Output** or press the **Freeze** button on the object's toolbar, a table or graph object is created that duplicates the current *view* of the original object.

Before you press **Freeze**, you are looking at a view of an object in the object window. Freezing the view makes a copy of the view and turns it into an independent object that will remain even if you delete the original object. A frozen view does not necessarily show what is currently in the original object, but rather shows a snapshot of the object at the moment

you pushed the button. For example, if you freeze a spreadsheet view of a series, you will see a view of a new *table object*; if you freeze a graphical view of a series, you will see a view of a new *graph object*.

Frozen graphs are a bit more sophisticated than other frozen objects. While frozen tables do not change when the workfile sample or data change, you may select if and when a frozen graph should update as the sample or its underlying data change. You may treat a frozen graph as a snapshot of the data at the time it was frozen, or update it at will. For a complete discussion of graph updating, see "Freezing a Graph View" on page 752.

The primary feature of freezing an object is that the tables and graphs created by freezing may be edited for presentations or reports. Many options for customization are only available for frozen objects or may be deleted when the object is refreshed, so you should be sure to freeze an object before customizing it.

Deleting Objects

To delete an object or objects from your workfile, select the object or objects in the workfile directory. When you have selected everything you want to delete, click **Delete** or **Object/ Delete Selected** on the workfile toolbar. EViews will prompt you to make certain that you wish to delete the objects.

Printing Objects

To print the currently displayed view of an object, push the **Print** button on the object window toolbar. You can also choose **File/Print** or **Object/Print** on the main EViews menu bar.

EViews will open a **Print** dialog containing the default print settings for the type of output you are printing. Here, we see the dialog for printing text information; the dialog for printing from a graph will differ slightly.

The default settings for printer type, output redirection, orientation, and text size may be set in the **Print Setup...** dialog (see "Print Setup" on page 880) or they may be overridden in the current print dialog.

For example, the print commands normally send a view or procedure output to the current Windows printer. You may specify instead that the output should be saved in the workfile as a table or graph, spooled to an RTF or
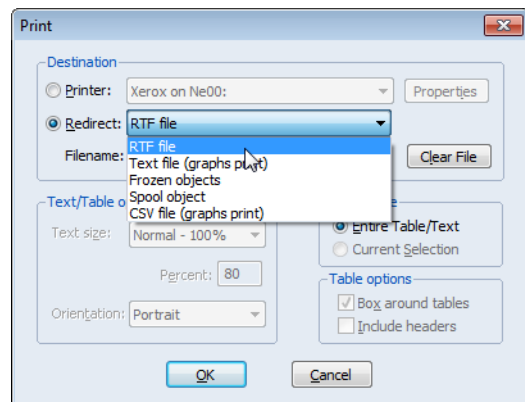
ASCII text file on disk, sent to a spool object. or CSV file. Simply click on **Redirect**, then select the output type from the list.

The "(graphs print)" in the entries indicates that graphs will still be sent to the printer since the specified redirect type does not support graph output.

Storing Objects

EViews provides three ways to save your data on disk. You have already seen how to save entire workfiles, where all of the objects in the workfile are saved together in a single file with the ".WF1" extension. You may also store individual objects in their own *data bank* files. They may then be fetched into other workfiles.

We will defer a full discussion of storing objects to data banks and databases until Chapter 10. "EViews Databases," on page 317. For now, note that when you are working with an object, you can place it in a data bank or database file by clicking on the **Object/Store to DB…** button on the object's toolbar or menu. EViews will prompt you for additional information.

You can store several objects, by selecting them in the workfile window and then pressing the **Object/Store selected to DB…** button on the workfile toolbar or menu.

Fetching Objects

You can fetch previously stored items from a data bank or database. One of the common methods of working with data is to create a workfile and then fetch previously stored data into the workfile as needed.

To fetch objects into a workfile, select **Object/Fetch from DB…** from the workfile menu or toolbar. You will see a dialog box prompting you for additional information for the fetch: objects to be fetched, directory and database location, as applicable.

See "Fetching Objects from the Database" on page 323 for details on the advanced features of the fetch procedure.

Updating Objects

Updating works like fetching objects, but requires that the objects be present in the workfile. To update objects in the workfile, select them from the workfile window, and click on **Object/Update from DB…** from the workfile menu or toolbar. The **Fetch** dialog will open, but with the objects to be fetched already filled in. Simply specify the directory and database location and click **OK**.

The selected objects will be replaced by their counterparts in the data bank or database.

See "Fetching Objects from the Database" on page 323 and "Database Auto-Series" on page 329 for additional details.

Copy-and-Paste of Object Information

You can copy the list of object information displayed in a workfile or database window to the Windows clipboard and paste the list to other program files such as word processing files or spreadsheet files. Simply highlight the objects in the workfile directory window, select **Edit/Copy** (or click anywhere in the highlighted area, with the right mouse button, and select **Copy**). Then move to the application (word processor or spreadsheet) where you want to paste the list, and select **Edit/Paste**.

If only object names and icons are displayed in the window, EViews will copy a single line containing the highlighted names to the clipboard, with each name separated by a space. If the window contains additional information, either because **View/Display Comments (Label + /−)** has been chosen in a workfile window or a query has been carried out in a database window, each name will be placed in a separate line along with the additional information.

Note that if you copy-and-paste the list of objects into another EViews workfile, the objects themselves will be copied.

Chapter 5. Basic Data Handling

The process of entering, reading, editing, manipulating, and generating data forms the foundation of most data analyses. Accordingly, much of your time in EViews will probably be spent working with data. EViews provides you with a sophisticated set of data manipulation tools that make these tasks as simple and straightforward as possible.

This chapter describes the fundamentals of working with data in EViews. There are three cornerstones of data handling in EViews: the two most common data objects, *series* and *groups*, and the use of *samples* which define the set of observations in the workfile that we wish to use in analysis.

We begin our discussion of data handling with a brief description of series, groups, and samples, and then discuss the use of these objects in basic input, output, and editing of data. Lastly, we describe the basics of frequency conversion.

In Chapter 6. "Working with Data," on page 179, we discuss the basics of EViews' powerful language for generating and manipulating the data held in series and groups. Subsequent chapters describe additional techniques and objects for working with data.

Data Objects

The actual numeric values that make up your data will generally be held in one or more of EViews' *data objects* (series, groups, matrices, vectors, and scalars). For most users, series and groups will by far be the most important objects, so they will be the primary focus of our discussion. Matrices, vectors, and scalars are discussed at greater length in Chapter 11. "Matrix Language," on page 261 in the *Command and Programming Reference*.

The following discussion is intended to provide only a brief introduction to the basics of series and groups. Our goal is to describe the fundamentals of data handling in EViews. An in-depth discussion of series and group objects follows in subsequent chapters.

Series

An EViews series contains a set of observations on a numeric variable. Associated with each observation in the series is a date or observation label. For series in regularly dated workfiles, the observations are presumed to be observed regularly over time. For undated data, the observations are not assumed to follow any particular frequency.

Note that the series object may only be used to hold numeric data. If you wish to work with alphanumeric data, you should employ alpha series. See "Alpha Series" on page 208 for discussion.

Creating a series

One method of creating a numeric series is to select **Object/New Object…** from the menu, and then to select **Series**. You may, at this time, provide a name for the series, or you can let the new series be untitled. Click on **OK**.

EViews will open a spreadsheet view of the new series object. All of the observations in the series will be assigned the missing value code "NA". You may then edit or use expressions to assign values for the series.

You may also use the **New Object** dialog to create alpha series. Alpha series are discussed in greater detail in "Alpha Series" on page 208.

A second method of creating a series is to generate the series using mathematical expressions. Click on **Quick/Generate Series…** in the main EViews menu, and enter an expression defining the series. We will discuss this method in depth in the next chapter.

Lastly, you may create the numeric or alpha series by entering a `series` or `alpha` command in the command window. Entering an expression of the form:

    series series_name = series_expr

creates a series with the name *series_name* and assigns the expression to each observaton. Alternately:

    alpha alpha_name = alpha_expr

creates an alpha series object and assigns the *alpha_expr* to each observation.

You may leave out the right-hand side assignment portion of the commands; in this case, the series or alpha will be initialized to missing values (NA and blank strings, respectively).

Changing the Spreadsheet Display

EViews provides you with extensive ability to customize your series spreadsheet display.

Column Widths

To resize the width of a column, simply move your mouse over the column separator until the icon changes, then drag the column to its desired width. The new width will be remembered the next time you open the series and will be used when the series is displayed in a group spreadsheet.
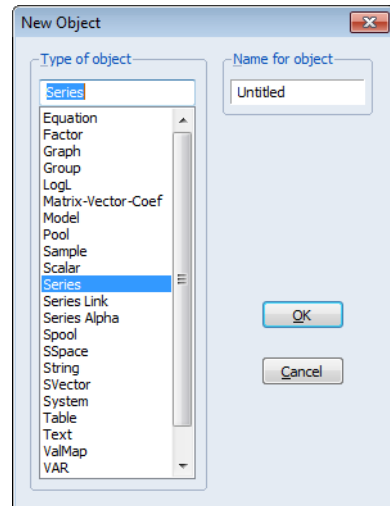
Display Type

The series display type, which is listed in the dropdown menu in the series toolbar, determines how the series spreadsheet window shows your data.

The **Default** method shows data in either raw (underlying data) form or, if a value map is attached to the series, shows the mapped values. Alternatively, you may use the **Raw Data** to show only the underlying data. See "Value Maps" on page 219 for a description of the use of value maps.
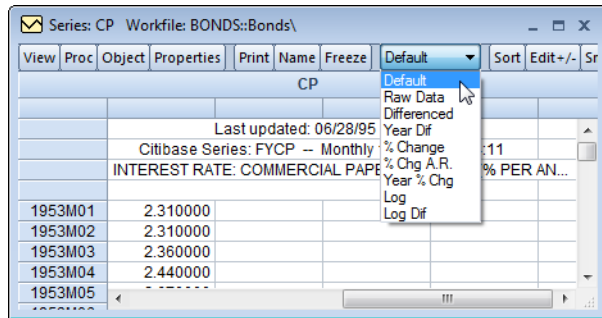
You may also use the display type setting to show transformations of the data. You may, for example, set the display method to **Differenced**, in order to have EViews display the first-differences of your data.
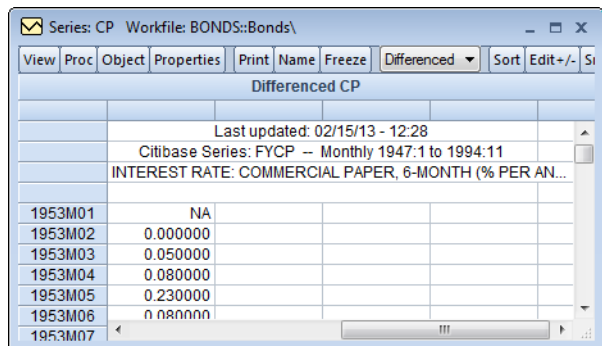
Changing the display of your series values does not alter the underlying values in the series, it only modifies the values shown in the spreadsheet (the series header, located above the labels, will also change to indicate the transformation). Note, however, that if you edit the values of your series while displayed in transformed mode, EViews will change the underlying values of the series accordingly. Changing the display and editing data in transformed mode is a convenient method of inputting data that arrive as changes or other transformed values.

Display Formats

You may customize the way that numbers or characters in your series are displayed in the spreadsheet by setting the series display properties. To display the dialog, click on **Properties** in the series toolbar, or right mouse click and select the **Display Format...** entry in the menu to display the first tab of the dialog.

EViews will open the **Properties** dialog
with the **Display** tab selected. You may use
this dialog to change the default column
width and justification for the series, and
to choose from a large list of numeric dis-
play formats for data displayed using the
different display types.

The **Data displayed as** dropdown menu is
used to select the display type to which the
current **Numeric display** settings will
apply. There are a different set of numeric
settings for each type.
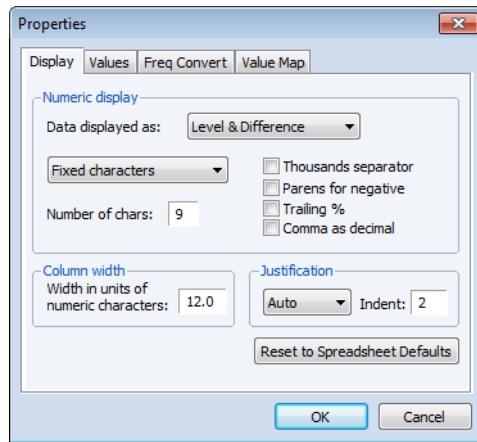
You may, for example, elect to change the
display of numbers to show additional digits, to separate thousands with a comma, or to dis-
play numbers as fractions. The last four items in the **Numeric display** dropdown menu pro-
vide options for the formatting of date number.

Similarly, you may elect to change the series justification by selecting **Auto**, **Left**, **Center**, or
Right. Note that **Auto** justification will set justification to right for numeric series, and left
for alpha series.

You may also use this dialog to change the column width (note that column widths in
spreadsheets may also be changed interactively by dragging the column headers).

Once you click on **OK**, EViews will accept the current settings and change the spreadsheet
display to reflect your choices. In addition, these display settings will be used whenever the
series spreadsheet is displayed or as the default settings when the series is used in a group
spreadsheet display.

Note that when you apply a display format, you may find that a portion of the contents of a
cell are not visible, when, for example, the column widths are too small to show the entire
cell. Alternately, you may have a numeric cell for which the current display format only
shows a portion of the full precision value.
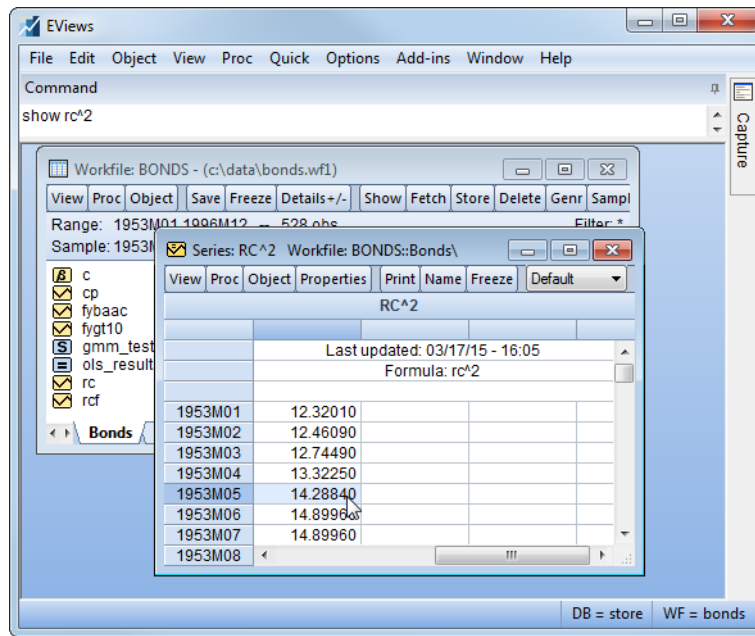
In these cases, it may be useful to examine the actual contents of a table cell. To do so, simply select the table cell. The unformatted contents of the cell will appear in the status line at the bottom of the EViews window.

Sorting a Series

The data in a series may be displayed sorted by observation or by the values in the series. You may change the order in which observations are *displayed* in the spreadsheet by clicking on the sort button and filling out the dialog. Note that this display sorting does not change the underlying order of the observations in the workfile.

You may press the **Sort** button on the button bar or by click on the right-mouse button and select **Sort** from the menu. To sort by series value, the entire series must be selected. To select the series, simply press the column header directly above the series values. Similarly, to sort by observation, the observation column must be selected.

If only a subset of the entire data series or observation series is selected, the **Sort** menu item will not be available.

Narrow versus Wide

The *narrow display* displays the observations for the series in a single column, with date labels in the margin. The typical series spreadsheet display will use this display format.

The *wide display* arranges the observations from left to right and top to bottom, with the label for the first observation in the row displayed in the margin. For dated workfiles, EViews will, if possible, arrange the data in a form which matches the frequency of the data. Thus, semi-annual data will be displayed with two observations per row, quarterly data will contain four observations per row, and 5-day daily data will contain five observations in each row.

You can change the display to show the observations in your series in multiple columns by clicking on the **Wide +/-** button on the spreadsheet view toolbar (you may need to resize the series window to make this button visible). For example, toggling the **Wide +/-** button switches the display between the wide display (as depicted), and the narrow (single column) display.
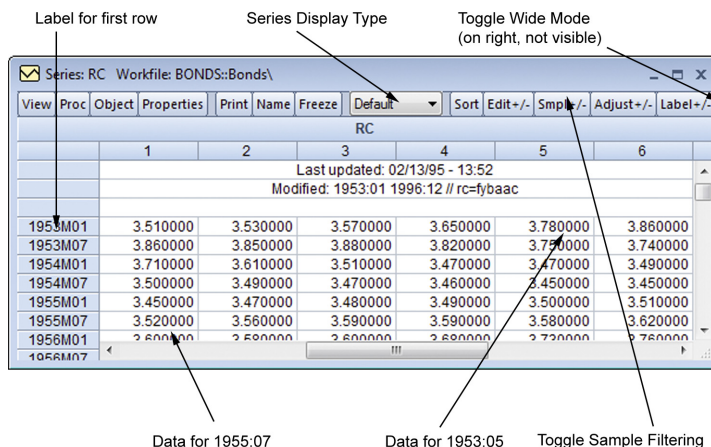
This wide display format is useful when you wish to arrange observations for a particular season in each of the columns.

Observation Numbers

If you wish to modify the observation labels to include an observation number, right-click and select **ObsID +/-**. Selecting **ObsID +/-** switches the display between showing a label that includes the observation label and one which does not. This feature is useful if you wish to identify the observation number of a particular observation.

Sample Subset Display

By default, all observations in the workfile are displayed, even those observations not in the current sample. By pressing **Smpl +/–** you can toggle between showing all observations in the workfile, and showing only those observations included in the current sample.

There are two features that you should keep in mind as you toggle between the various display settings:

- If you choose to display only the observations in the current sample, EViews will switch to single column display.

- If you switch to wide display, EViews automatically turns off the display filter so that all observations in the workfile are displayed.

One consequence of this behavior is that if you begin with a narrow display of observations in the current sample, click on **Wide +/-** to switch to wide display, and then press the **Wide +/-** button again, EViews will provide a narrow display of all of the observations in the workfile. To return to the original narrow display of the current sample, you will need to press the **Smpl +/-** button again.

Live statistics

The series and group spreadsheet views now display interactive statistics. When a spreadsheet is first opened the statistics are computed using all of the data in the spreadsheet.

Double-clicking on the bottom bar brings up a menu where the statistics shown can be changed. A maximum of six statistics may be displayed at once. The window may need to be resized to display all the statistics.
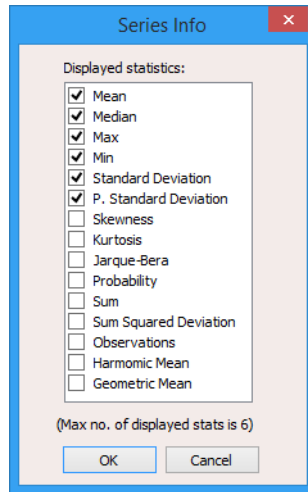
The statistics are recalculated every time a group of cells is highlighted.

The statistics displayed can also be set by going to **Options/General Options...** from the main menu and then the **Spreadsheets/Live statistics** page.
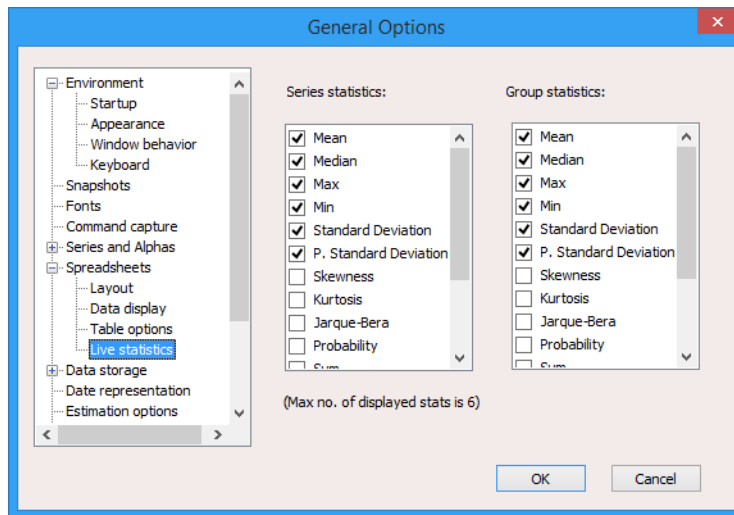
Editing a series

You can edit individual values of the data in a series.

First, open the spreadsheet view of the series. If the series window display does not show the spreadsheet view, click on the **Sheet** button, or select **View/Spreadsheet**, to change the default view.

Next, make certain that the spreadsheet window is in *edit mode*. EViews provides you with the option of protecting the data in your series by turning off the ability to edit from the spreadsheet window. You can use the **Edit +/–** button on the toolbar to toggle between *edit mode* and *protected mode*.

Here we see a series spreadsheet window in edit mode. Notice the presence of the edit window just beneath the series toolbar containing the value of RC in 1953M01, and the box around the selected cell in the spreadsheet; neither are present in protected mode.
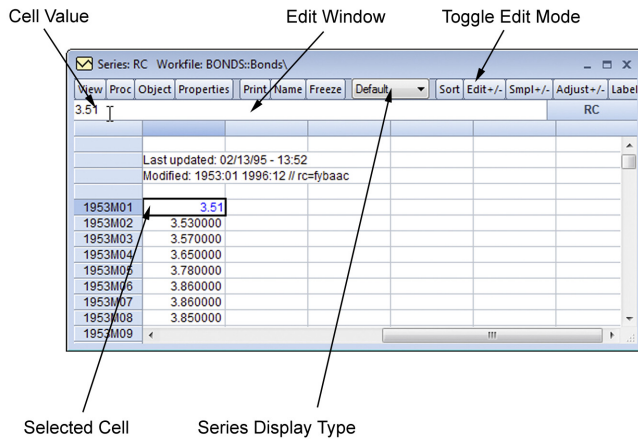
To change the value for an observation, select the cell, type in the value, and press ENTER. For example, to change the value of RC in 1953M01, simply click on the cell containing the value, type the new value in the edit window, and press ENTER.

When editing series values, you should pay particular attention to the series display format, which tells you the units in which your series are displayed. Here, we see that the series values are displayed in **Default** mode so that you are editing the underlying series values (or their value mapped equivalents). Alternately, if the series were displayed in **Differenced** mode, then the edited values correspond to the first differences of the series.

Note that some cells in the spreadsheet are protected. For example, you may not edit the observation labels, or the "Last update" series label. If you select one of the protected cells, EViews will display a message in the edit window telling you that the cell cannot be edited.

When you have finished editing, you should protect yourself from inadvertently changing values of your data by clicking on **Edit +/–** to turn off edit mode.

Advanced tools for editing series values are described in "Series Adjust" on page 433.

Inserting and Deleting Observations in a Series

You can also insert and delete observations in the series. First, click on the cell where you want the new observation to appear. Next, right click and select **Insert obs...** or **Delete obs...** from the menu. You will see a dialog asking how many observations you wish to insert or delete at the current position.

If you choose to insert a single observation, EViews will insert a missing value at the appropriate position and push all of the observations down so that the last observation will be lost from the workfile. If you wish to preserve this observation, you will have to expand the workfile before inserting observations. If you choose to delete an observation, all of the remaining observations will move up, so that you will have a missing value at the end of the workfile range.
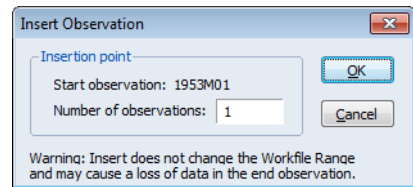
Groups

When working with multiple series, you will often want to create a group object to help you manage your data. A group is a list of series names (and potentially, mathematical expressions) that provides simultaneous access to all of the elements in the list.

With a group, you can refer to sets of variables using a single name. Thus, a set of variables may be analyzed, graphed, or printed using the group object, rather than each one of the individual series. Therefore, groups are often used in place of entering a lengthy list of names. Once a group is defined, you can use the group name in many places to refer to all of the series contained in the group.

You will also create groups of series when you wish to analyze or examine multiple series at the same time. For example, groups are used in computing correlation matrices, testing for cointegration and estimating a VAR or VEC, and graphing series against one another.

Creating Groups

There are several ways to create a group. Perhaps the easiest method is to select **Object/New Object…** from the main menu or workfile toolbar, click on **Group**, and if desired, name the object.

You should enter the names of the series to be included in the group, separated by spaces, and then click **OK**.

You may use the wildcard characters "*" and "?" to match more than one series in the workfile, and you may use the keywords "and" and "not" to specify that certain items should be excluded from the group.
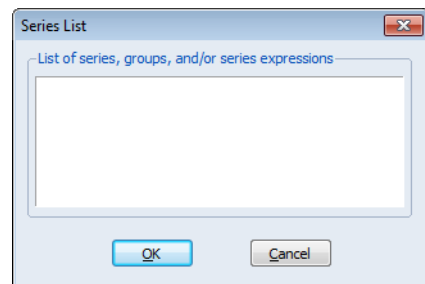
A group window will open showing a spreadsheet view of the group.

You may have noticed that the dialog allows you to use group names and series expressions. If you include a group name, all of the series in the named group will be included in the new group. For example, suppose that the group GR1 contains the series X, Y, and Z, and you create a new group GR2, which contains GR1 and the series A and B. Then GR2 will contain X, Y, Z, A and B. Bear in mind that only the series contained in GR1, not GR1 itself, are included in GR2; if you later add series to GR1, they will not be added to GR2.

Series expressions will be discussed in greater depth later. For now, it suffices to note that series expressions are mathematical expressions that may involve one or more series (*e.g.* "7/2" or "3*X*Y/Z"). EViews will automatically evaluate the expressions for each observation and display the results as if they were an ordinary series. Users of spreadsheet programs will be familiar with this type of automatic recalculation.

Here, for example, is a spreadsheet view of an untitled group containing the series RC, a series expression for the lag of RG, RG(–1), and a series expression involving RC and RG.

Notice here the **Default** setting for the group spreadsheet display indicates that the series RC and RG(-1) are displayed using the original values,

	RG	RG(-1)	2*RC/RG
1953M01	0.000000	NA	NA
1953M02	0.000000	0.000000	NA
1953M03	0.000000	0.000000	NA
1953M04	2.830000	0.000000	2.579505
1953M05	3.050000	2.830000	2.478689
1953M06	3.110000	3.050000	2.482315
1953M07	2.930000	3.110000	2.634812
1953M08	2.950000	2.930000	2.610169
1953M09	2.870000	2.950000	2.703833
1953M10	2.660000	2.870000	2.872180
1953M11	2.680000	2.660000	2.798507
1953M12	2.590000	2.680000	2.888031
1954M01	2.480000	2.590000	2.991935
1954M02			

spreadsheet types, and formats set in the original series (see "Display Formats" on page 125). A newly created group always uses the **Default** display setting, regardless of the settings in the original series, but the group does adopt the original series cell formatting. You may temporarily override the display setting by selecting a group display format. For example, to use the display settings of the original series, you should select **Series Spec**; to display differences of all of the series in the group, select **Differenced**.

An equivalent method of creating a group is to select **Quick/Show…**, or to click on the **Show** button on the workfile toolbar, and then to enter the list of series, groups and series expressions to be included in the group. This method differs from using **Object/New Object…** only in that it does not allow you to name the object at the time it is created.

You can also create an empty group that may be used for entering new data from the keyboard or pasting data copied from another Windows program. These methods are described in detail in "Entering Data" on page 146 and "Copying-and-Pasting" on page 148.

Editing in a Group

Editing data in a group is similar to editing data in a series. Open the group window, and click on **Sheet**, if necessary, to display the spreadsheet view. If the group spreadsheet is in protected mode, click on **Edit +/–** to enable edit mode, then select a cell to edit, enter the new value, and press ENTER. The new number should appear in the spreadsheet.

Since groups are simply references to series, editing the series within a group changes the values in the original series.

As with series spreadsheet views, you may click on **Smpl +/–** to toggle between showing all of the observations in the workfile and showing only those observations in the current sample. Unlike the series window, the group window always shows series in a single column.

Note that while groups inherit many of the series display formats when they are created, to reduce confusion, groups do not initially show transformed values of the series. If you wish to edit a series in a group in transformed form, you must explicitly set a transformation type for the group display.

Adding to or Modifying a Group

You can add to or modify an existing group by clicking on **View/Group Members** and then editing the list of names. Clicking on the **UpdateGroup** accepts your edits (see "Group Members" on page 547 for details).

Alternately, you can drag-and-drop series into the **Group Members** or **Spreadsheet** views of the group EViews will automatically add the series to the group.
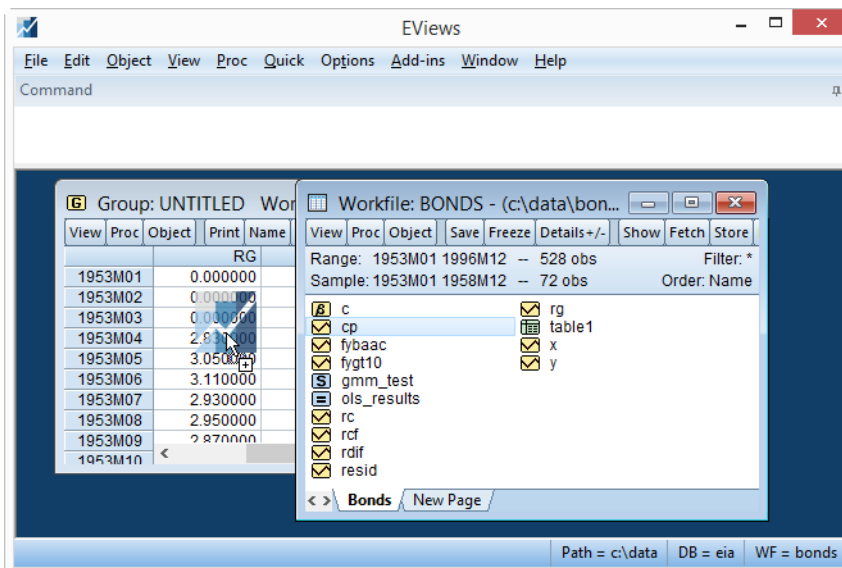
Samples

One of the most important concepts in EViews is the *sample* of observations. The sample is the set (often a subset) of observations in the workfile to be included in data display and in performing statistical procedures. Samples may be specified using ranges of observations and "if conditions" that observations must satisfy to be included.

For example, you can tell EViews that you want to work with observations from 1953M1 to 1970M12 and 1995M1 to 1996M12. Or you may want to work with data from 1953M1 to 1958M12 where observations in the RC series exceed 3.6.

The remainder of this discussion describes the basics of using samples in non-panel work-files. For a discussion of panel samples, see "Panel Samples," beginning on page 902 of *User's Guide II*.

The Workfile Sample

When you create a workfile, the *workfile sample* or *global sample* is set initially to be the entire range of the workfile. The workfile sample tells EViews what set of observations you wish to use for subsequent operations. Unless you want to work with a different set of observations, you will not need to reset the workfile sample.

You can always determine the current workfile sample of observations by looking at the top of your workfile window.

Workfile sample (included observations)   Workfile range (all observations)   Click to change sample

Workfile: BONDS - (c:\data\bonds.wf1)   _ □ ✕

View | Proc | Object | Save | Freeze | Details+/- | Show | Fetch | Store | Delete | Genr | Sample

Range: 1953M01 1996M12 -- 528 obs   Filter: *
Sample: 1953M01 1958M12 if rc>3.6 -- 45 obs   Order: Name

Double-click to change sample
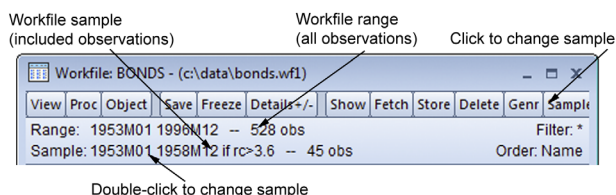
Here the BONDS workfile consists of 528 observations from January 1953 to December 1996. The current workfile sample uses a subset of those observations consisting of the 45 observations between 1953M01 and 1958M12 for which the value of the RC series exceeds 3.6.

Changing the Sample

There are four ways to set the workfile sample: you may click on the **Sample** button in the workfile toolbar, you may double click on the sample string display in the workfile window, you can select **Proc/Set Sample…** from the main workfile menu, or you may enter a `smpl` command in the command window. If you use one of the interactive methods, EViews will open the **Sample** dialog prompting you for input.

Date Pairs

In the upper edit field you will enter one or more pairs of dates (or observation numbers). Each pair identifies a starting and ending observation for a range to be included in the sample.

Sample

Sample range pairs (or sample object to copy)

1953 1958

OK

IF condition (optional)

rc>3.6

Cancel
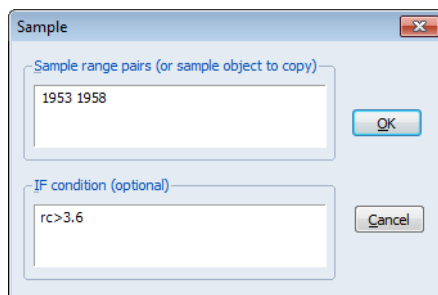
For example, if, in an annual workfile, you entered the string "1950 1980 1990 1995", EViews will use observations for 1950 through 1980 and observations for 1990 through 1995 in subsequent operations; observations from 1981 through 1989 will be excluded. For undated data, the date pairs correspond to observation identifiers such as "1 50" for the first 50 observations.

You may enter your date pairs in a frequency other than that of the workfile. Dates used for the starts of date pairs are rounded down to the first instance of the corresponding date in the workfile frequency, while dates used for the ends of date pairs are rounded up to the last instance of the corresponding date in the workfile frequency. For example, the date pair "1990m1 2002q3" in an annual workfile will be rounded to "1990 2002", while the date pair "1/30/2003 7/20/2004" in a quarterly workfile will be rounded to "2003q1 2004q3".

For intraday data, the sample may be further specified by including a time after the start and end dates. For example, given an hourly workfile, entering "1/3/2000 10AM 12/30/2000 2PM" will limit the start time to 10AM on 1/3/2000 and to 2PM on 12/30/2000.

EViews provides special keywords that may make entering sample date pairs easier. First, you can use the keyword "@ALL", to refer to the entire workfile range. In the workfile above, entering "@ALL" in the dialog is equivalent to entering "1953M1 1996M12". Furthermore, you may use "@FIRST" and "@LAST" to refer to the first and last observation in the workfile. Thus, the three sample specifications for the above workfile:

```
@all
@first 1996m12
1953m1 @last
```

are identical.

Note that when interpreting sample specifications involving days, EViews will, if necessary, use the global defaults ("Date Representation" on page 871) to determine the correct ordering of days, months, and years. For example, the order of the months and days is ambiguous in the date pair:

```
1/3/91 7/5/95
```

so EViews will use the default date settings to determine the desired ordering. We caution you, however, that using the default settings to disambiguate dates in samples is not generally a good idea since a given pair may be interpreted in different ways at different times if your settings change.

Alternately, you may use the IEEE standard format, "YYYY-MM-DD", which uses a four-digit year, followed by a dash, a two-digit month, a second dash, and a two-digit day. The presence of a dash in the format means that you must enclose the date in quotes for EViews to accept this format. For example:

```
"1991-01-03" "1995-07-05"
```

will always be interpreted as January 3, 1991 and July 5, 1995. See "Free-format Conversion Details" on page 114 in the *Command and Programming Reference* for related discussion.

Sample IF conditions

The lower part of the sample dialog allows you to add conditions to the sample specification. The sample is the intersection of the set of observations defined by the range pairs in the upper window and the set of observations defined by the "if" conditions in the lower window. For example, if you enter:

Upper window: `1980 1993`
Lower window: `incm > 5000`

the sample includes observations for 1980 through 1993 where the series INCM is greater than 5000.

Similarly, if you enter:

Upper window: `1958q1 1998q4`

Lower window: `gdp > gdp(-1)`

all observations from the first quarter of 1958 to the last quarter of 1998, where GDP has risen from the previous quarter, will be included.

The "or" and "and" operators allow for the construction of more complex expressions. For example, suppose you now wanted to include in your analysis only those individuals whose income exceeds 5000 dollars per year and who have at least 13 years of education. Then you can enter:

Upper window: `@all`

Lower window: `income > 5000 and educ >= 13`

Multiple range pairs and "if" conditions may also be specified:

Upper window: `50 100 200 250`

Lower window: `income >= 4000 and educ > 12`

includes undated workfile observations 50 through 100 and 200 through 250, where the series INCOME is greater than or equal to 4000 and the series EDUC is greater than 12.

You can create even more elaborate selection rules by including EViews built-in functions:

Upper window: `1958m1 1998m1`

Lower window: `(ed>=6 and ed<=13) or earn<@mean(earn)`

includes all observations where the value of the variable ED falls between 6 and 13, or where the value of the variable EARN is lower than its mean. Note that you may use parentheses to group the conditions and operators when there is potential ambiguity in the order of evaluation.

It is possible that one of the comparisons used in the conditioning statement will generate a missing value. For example, if an observation on INCM is missing, then the comparison INCM > 5000 is not defined for that observation. EViews will treat such missing values as though the condition were false, and the observation will not be included in the sample.

Sample Commands

You may find it easier to set your workfile sample from the command window—instead of using the dialog, you may set the active sample using the `smpl` command. Simply click on the command window to make it active, and type the keyword "SMPL", followed by the sample string:

```
smpl 1955m1 1958m12 if rc>3.6
```

and then press ENTER (notice, in the example above, the use of the keyword "IF" to separate the two parts of the sample specification). You should see the sample change in the workfile window.

Samples for Intraday data

When your data is specified in a frequency higher than a day, you may use IF conditions to set the sample to include specific hours over multiple days. Four functions in EViews facilitate this: `@hour`, `@minute`, `@second`, and `@hourf`.

Suppose you have minute data over all 24 hours and would like to set the sample to only include 9:30AM to 2:30PM. This can be accomplished using the `@hourf` command, which returns the observation time as a floating point hour:

```
smpl if @hourf>=9.5 and @hourf<=14.5
```

To select only observations that appear on the half hour, use the `@minute` command:

```
smpl if @minute=0 or @minute=30
```

To select only observations that are on Mondays at 10AM:

```
smpl if @weekday=1 and @hourf=10
```

Sample Offsets

Sample range elements may contain mathematical expressions to create date offsets. This feature can be particularly useful in setting up a fixed width window of observations. For example, in the regular frequency monthly workfile above, the sample string:

```
1953m1 1953m1+11
```

defines a sample that includes the 12 observations in the calendar year beginning in 1953M1.

While EViews expects date offsets that are integer values, there is nothing to stop you from adding or subtracting non-integer values—EViews will automatically convert the number to an integer. You should be warned, however, that the conversion behavior is not guaranteed to be well-defined. If you must use non-integer values, you are strongly encouraged to use the `@round`, `@floor` or `@ceil` functions to enforce the desired behavior.

The offsets are perhaps most useful when combined with the special keywords to trim observations from the beginning or end of the sample. For example, to drop the first observation in your sample, you may use the sample statement:

```
smpl @first+1 @last
```

Accordingly, the following commands generate a series containing cumulative sums of the series X in XSUM:

```
smpl @first @first
```

```
series xsum = x
smpl @first+1 @last
xsum = xsum(-1) + x
```

(see "Basic Assignment" on page 190). The first two commands initialize the cumulative sum for the first observation in each cross-section. The last two commands accumulate the sum of values of X over the remaining observations.

Similarly, if you wish to estimate your equation on a subsample of data and then perform cross-validation on the last 20 observations, you may use the sample defined by,

```
smpl @first @last-20
```

to perform your estimation, and the sample,

```
smpl @last-19 @last
```

to perform your forecast evaluation.

While the use of sample offsets is generally straightforward, there are a number of important subtleties to note when working with irregular dated data and other advanced workfile structures ("Advanced Workfiles" on page 263). To understand the nuances involved, note that there are three basic steps in the handling of date offsets.

First, dates used for the starts of date pairs are rounded down to the first instance of the corresponding date in the workfile regular frequency, while dates used for the ends of date pairs are rounded up to the last instance of the corresponding date in the regular frequency. If date pairs are specified in the workfile frequency (*e.g.*, the pair "1990 2000" is used in an annual workfile), this step has no effect.

Next, EViews examines the workfile frequency date pair to determine whether the sample dates fall within the range of the observed dates in the workfile, or whether they fall outside the observed date range. The behavior of sample offsets differs in the two cases.

For simplicity of discussion, assume first that both dates fall within the range of observed dates in the workfile. In this case:

- EViews identifies base observations consisting of the earliest and latest workfile observations falling within the date pair range.

- Offsets to the date pair are then applied to the base observations by moving through the workfile observations. If, for example, the offset for the first element of a date pair is " + 1", then the sample is adjusted so that it begins with the observation following the base start observation. Similarly, if the offset for the last element of a date pair is "-2", then the sample is adjusted to end two observations prior to the base end observation.

Next, we assume that both dates fall outside the range of observed workfile dates. In this setting:

- EViews applies offsets to the date pair outside of the workfile range using the regular frequency until the earliest and latest workfile dates are reached. The base observations are then set to the earliest and latest workfile observations.

- Any remaining offsets are applied to the base observations by moving through the workfile observations, as in the earlier case.

The remaining two cases, where one element of the pair falls within, and the other element falls outside the workfile date range, follow immediately.

It is worth pointing out that the difference in behavior is not arbitrary. It follows from the fact that within the date range of the data, EViews is able to use the workfile structure to identify an irregular calendar, but since there is no corresponding information for the dates beyond the range of the workfile, EViews is forced to use the regular frequency calendar.

A few examples will help to illustrate the basic concepts. Suppose for example, that we have an irregular dated annual workfile with observations for the years "1991," "1994," "1995," "1997," "2000," and "2002":

The sample statement:

```
smpl 1993m8+1 2002q2-2
```

is processed in several steps. First, the date "1993m8" is rounded to the previous regular frequency date, "1993," and the date "2002q2" is rounded up to the last instance of the regular frequency date "2002"; thus, we have the equivalent sample statement:

```
smpl 1993+1 2002-2
```

Next, we find the base observations in the workfile corresponding to the base sample pair ("1993 2002"). The "1994" and the "2002" observations are the earliest and latest, respectively, that fall in the range.

Lastly, we apply the offsets to the remaining observations.

The offsets for the start and end will drop one observation ("1994") from the beginning and two observations ("2002" and "2000") from the end of the sample, leaving two observations ("1995," "1997") in the sample.
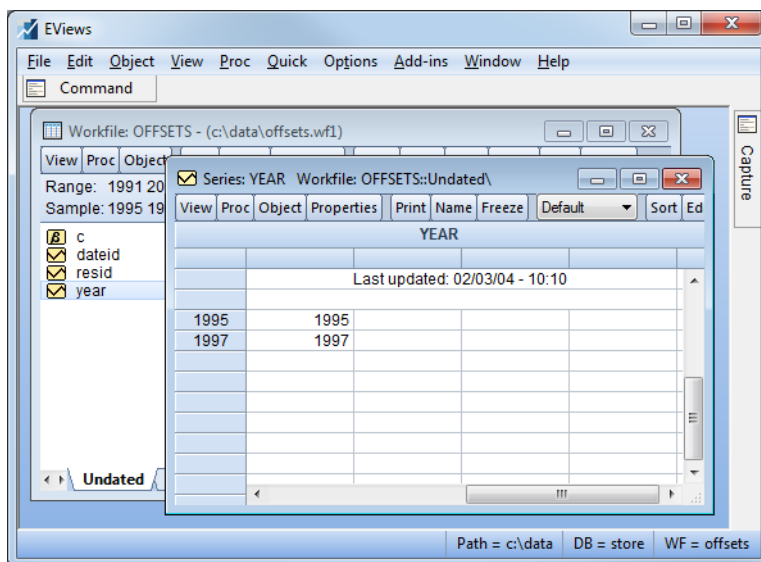
Consider instead the sample statement:

```
smpl 1995-1 2004-4
```

In this case, no rounding is necessary since the dates are specified in the workfile frequency.

For the start of the date pair, we note that the observation for "1995" corresponds to the start date. Computing the offset "-1" simply adds the "1994" observation.

For the end of the date pair, we note that "2004" is beyond the last observation in the workfile,
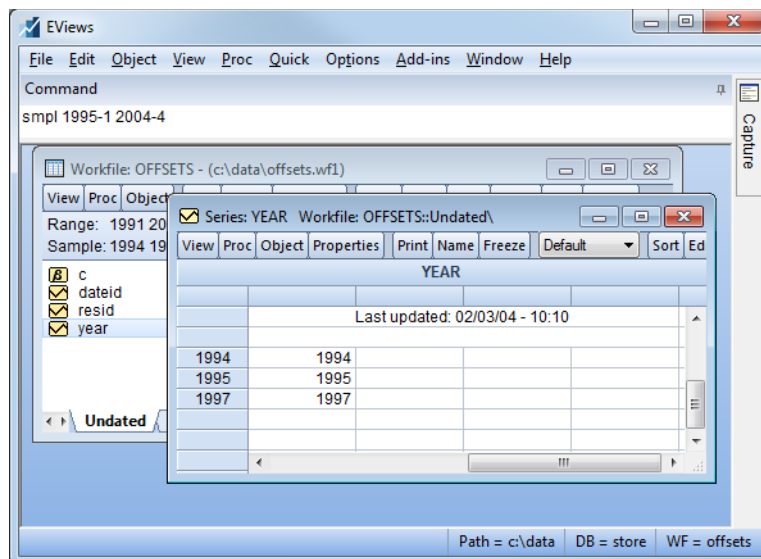
"2002". We begin by computing offsets to "2004" using the regular frequency calendar, until we reach the highest date in the workfile, so that we "drop" the two observations "2004" and "2003". The remaining two offsets, which use the observed dates, drop the observations for "2002" and "2000". The resulting sample includes the observations "1994," "1995," and "1997".

Sample Objects

As you have seen, it is possible to develop quite elaborate selection rules for the workfile sample. However, it can become quite cumbersome and time-consuming to re-enter these rules if you change samples frequently. Fortunately, EViews provides you with a method of saving sample information in an object which can then be referred to by name. If you work with many well-defined subsets of your data, you will soon find sample objects to be indispensable.

Creating a Sample Object

To create a sample object, select **Object/New Object…** from the main menu or the workfile toolbar. When the **New Object** dialog appears, select **Sample** and, optionally provide a name. If you do not provide a name, EViews will automatically assign one for you (sample objects may not be untitled). Click on **OK** and EViews will open the sample object specification dialog:

Here is a partially filled-in sample object dialog for SMPL1. Notice that while this dialog looks very similar to the one we described above for setting the sample, there are minor cosmetic differences: the name of the sample object appears in the title bar, and there is a check box for setting the workfile sample equal to this sample object.
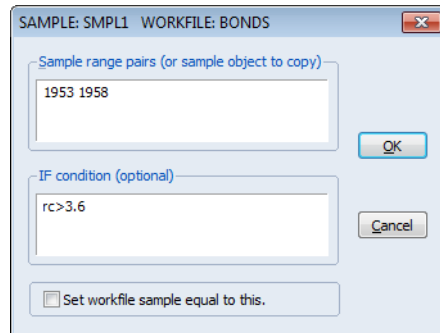
These cosmetic differences reflect the two distinct purposes of the dialog: (1) to define the sample object, and (2) to set the workfile sample. Since EViews separates the act of defining the sample object from the act of setting the workfile sample, you can define the object without changing the workfile sample, and vice versa.

To define the sample object, you should fill out this dialog as described before and click on **OK**. The sample object now appears in the workfile directory with a double-arrow icon.

To declare a sample object using a command, simply issue the `sample` declaration, followed by the name to be given to the sample object, and then the sample string:

```
sample mysample 1955m1 1958m12 if rc>3.6
```

EViews will create the sample object MYSAMPLE which will use observations between 1955:01 and 1958:12, where the value of the RC series is greater than 3.6.

Using a Sample Object

You may use a previously defined sample object directly to set the workfile sample. Simply open a sample object by double clicking on the name or icon. This will reopen the sample dialog. If you wish to change the sample object, you may edit the sample specification; otherwise, simply click the **Set workfile sample** check box and click on **OK**.

Or, you may set the workfile sample using the sample object, by entering the smpl command, followed by the sample object name. For example, the command:

```
smpl mysample
```

will set the workfile sample according to the rules contained in the sample object MYSAMPLE.

For many purposes, you may also use a named sample object as though it were an ordinary EViews series containing the values 1 and 0, for observations that are and are not included, respectively. Thus, if SMP2 is a named sample object, you may use it as though it were a series in any EViews expressions (see "Series Expressions" on page 181). For example:

```
y1*(smp2=0) + 3*y2*(smp2=1)
```

is a valid EViews expression, evaluating to the value of 3*Y2 if an observation is in SMP2, and Y1, otherwise.

You may also, for example, create a new series that is equal to a sample object, and then examine the values of the series to see which observations do and do not satisfy the sample criterion.

Additionally, one important consequence of this treatment of sample objects is that you may use sample objects in the construction of other sample objects. For example, if you create a sample object FEMALE containing observations for individuals who are females,

```
sample female @all if gender="female"
```

and a second sample object HIGHINC if INCOME is greater than 25000:

```
sample highinc @all if income>25000
```

You may set the sample to observations where individuals are low income females using:

```
smpl @all if female and not highinc
```

where we use the NOT keyword to take the complement of the observations in HIGHINC. To create a sample object HIGHFEMALE using this sample, use the command:

```
sample highfemale @all if female and not highinc
```

Alternatively, we could have used the equivalent expression

```
sample highfemale @all if female and highinc=0
```

More generally, we may use any expression involving sample objects and the keywords "AND", "OR", and "NOT", as in

```
smpl 1950 1980 if female or not highinc
```

which sets the sample to those observations from 1950 to 1980 that are also in the sample FEMALE, but not in the sample HIGHINC.

Importing Data

Earlier, we described workfile creation tools that allow you to open data from foreign sources into a *new* workfile ("Creating a Workfile by Reading from a Foreign Data Source" on page 47). This is most likely the easiest way to move data from foreign files and database sources such as ODBC into EViews and we recommend it as your first approach, but you should note that these tools are expressly designed for creating new workfiles.

Alternatively, you may wish to import data into an *existing* workfile, perhaps into existing series in the workfile—you may, for example, wish to read a portion of an Excel file into a subset of observations in a series or group of series. We term the reading of data into existing workfiles and/or series *importing series data* to distinguish it from the creation of entirely new workfiles and series.

There are several methods for importing series data into EViews. In the remainder of this discussion, we outline the basics of data import from spreadsheet, text file, or printed formats, into series and group objects. Note that we omit, for the moment, discussion of importing data into EViews matrix, vector and pool objects, and discussion of EViews and foreign databases:

- Matrix and vector import tools are touched on briefly in "Matrix and Table Object Import" on page 161.

- Pool import is described in "Importing Pooled Data" on page 850 of *User's Guide II*.

- EViews databases are the subject of Chapter 10. "EViews Databases," beginning on page 317.

Entering Data

For small datasets in printed form, you may wish to enter the data by typing at the keyboard.

- Your first step is to open a temporary spreadsheet window in which you will enter the data. Choose **Quick/Empty Group (Edit Series)** from the main menu to open an untitled group window:

1. Click up arrow once
2. Click in box next to second obs label
3. Enter name
4. Select series type (dialog shown below)
5. Enter data in cells
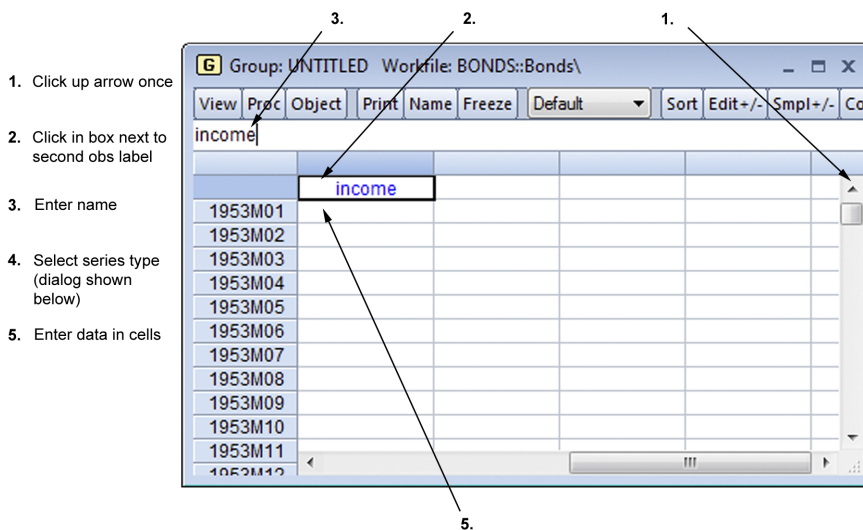
- The next step is to create and name the series. First click once on the up arrow in the scroll bar to display the second **obs** label on the left-hand column. The row of cells next to the second **obs** label is where you will enter and edit series names.

  Click once in the cell next to the second **obs** label, and enter your first series name. Here we have typed "income" in the edit window (the name in the cell changes as we type in the edit window). Press RETURN. If you enter the name of an existing series, the series data will be brought into the group.

- EViews will prompt you to specify a series type for the column. You may select a numeric series, numeric series containing date value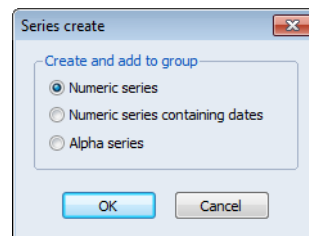s, or an alpha series. When you click on **OK**, EViews will create a numeric or alpha series and will apply formatting information that will aid you in viewing your data.

- You should repeat this procedure in subsequent columns for each additional series. If you decide you want to rename one of your series, simply select the cell containing the series name, edit the name in the edit window, and then press RETURN. EViews will prompt you to confirm the series rename.

- To enter the data, click on the appropriate cell and type the number or text. Pressing RETURN after entering the cell value will move you to the next cell. If you prefer, you can use the cursor keys to navigate the spreadsheet.

- When you are finished entering data, close the group window. If you wish, you can first name the untitled group by clicking on the **Name** button. Otherwise, if you do

not wish to keep the group, answer **Yes** when EViews asks you to confirm the deletion.

Copying-and-Pasting
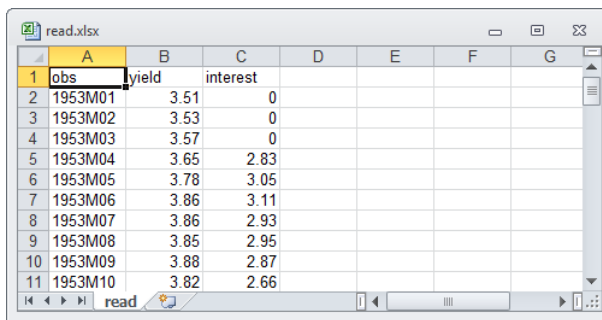
The Windows clipboard is a handy way to move small amounts of data within EViews and between EViews and other software applications. It is a natural tool for importing these types of data from Excel and other Windows applications that support Windows copy-and-paste.

Copying from Windows Applications

The following discussion involves an example using an Excel spreadsheet, but the basic principles apply for other Windows applications.

Suppose you have bond yield and interest rate data in an Excel spreadsheet that you would like to bring into EViews.

Open the spreadsheet in Excel. Your first step is to highlight the cells to be imported into EViews. Since the column headings YIELD and INTEREST will be used as EViews variable names, you should highlight them as well. Since EViews understands dated data, and we are going to create a monthly workfile, you do not

need to copy the date column. Instead, click on the column label B and drag to the column label C. The two columns of the spreadsheet will be highlighted.
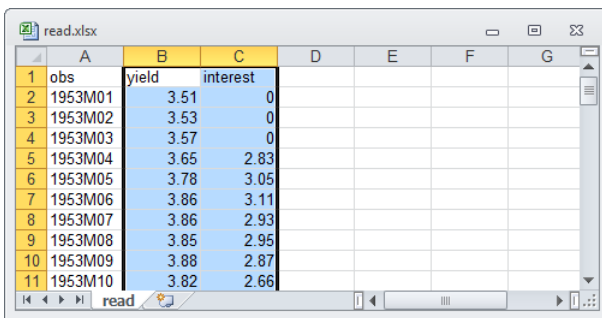
Select **Edit/Copy** to copy the highlighted data to the clipboard.

Pasting into New Series

Start EViews and create a new, or load an existing, monthly workfile containing the dates in the Excel spreadsheet (in our example, 1953m01 through

1994m11). Make certain that the sample is set to include the same observations that you have copied onto the clipboard.

Select **Quick/Empty Group (Edit Series)**. Note that the spreadsheet opens in edit mode so there is no need to click the **Edit +/−** button.

Here, we have created a monthly workfile with a range from 1953:1 to 1999:12. The first row of the EViews spreadsheet is labeled 1953:01. Since we are pasting in the series names as well, you should click on the up arrow in the scroll bar to make room for the series names.

Place the cursor in the upper-left cell, just to the right of the second **obs** label. Then select **Edit/Paste** from the main menu (*not* **Edit +/−** in the toolbar). The group spreadsheet will now contain the data from the clipboard.

EViews automatically analyzes the data on the clipboard to determine the most likely series type. If, for example, your series contains text that can always be interpreted as a number, EViews will create a numeric series. Here, the numeric series YIELD and INTEREST have been created in the workfile.

If the numbers in the series may all be interpreted as date values, or if the data are all string representations of dates, EViews will create a numeric series formatted to display dates.

If you paste a name corresponding to an object that already exists in the workfile, EViews will find the next available name by appending an integer to the series name. For example, if SER already exists in the workfile, pasting the name "SER" will create a series SER01.

You may now close the group window and delete the untitled group without losing the two series.

Pasting into Existing Series

You can import data from the clipboard into an existing EViews series or group spreadsheet by using **Edit/Paste** in the same fashion. There are only a few additional issues to consider:

- To paste several series, you will first open a group window containing the existing series. The easiest way to do this is to click on **Show**, and then type the series names in the order they appear on the clipboard. Alternatively, you can create an untitled group by selecting the first series, holding down the **Ctrl**-key and click select each subsequent series (in order), and then double clicking to open.

- Make certain that the group window is showing the sample range that corresponds to the data on the clipboard.

- Next, make certain that the group window is in edit mode. If not in edit mode, press the **Edit +/−** button to toggle to edit mode. Place the cursor in the target cell, and select **Edit/Paste** from the main menu.

- Finally, click on **Edit +/−** to return to protected mode.

- If you are pasting into a single series you will need to make certain that the series window is in edit mode, and that the series is viewed in a single column. If the series is in multiple columns, push on the **Wide +/−** button. Then **Edit/Paste** the data as usual, and click on **Edit +/−** to protect the data.

Linking pasted data

When pasting data into EViews, either as a new workfile, or into an existing workfile, EViews will ask if the new series objects should be linked back to the source. Click **Yes** to mark the new series objects as externally linked. Linked objects can be refreshed with updated data when the workfile is next opened, or on demand by right-clicking on the workfile window and selecting **Update from DB…** or by clicking on **Objects/Manage Links & Formulae…** in the main menu to display the link management dialog.

When you next open your existing workfile containing linked data, EViews will prompt you to refresh the data, and optionally, to display a comparison of the data before and after the refresh:
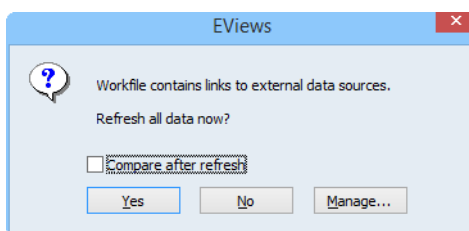
Importing Data

EViews provides easy-to-use tools for importing data into an existing workfile, matching observations between the source and destination as required. The source data may be available in any of a number of data formats, from one of the EViews supported foreign file formats (*i.e.*, Microsoft Excel, ASCII text, SPSS, SAS portable, Stata, *etc.*), to an EViews format workfile.

Opening the Source File

To use the import tools, first, make certain that you have an open workfile to receive the contents of the data import and that the workfile window is active, then click on **File-**

Import/Import from File... or click on the **Proc** menu of the workfile and select **Import from File...** In either case, EViews will display a standard file **Open** dialog.

Next, type the name of the file you wish to read (with full path information, if appropriate) or select a file type, navigate to the directory containing the file, and double click on the name. Click on the **Open** button to continue.

(Alternately, you may simply drag-and-drop the source file onto the open workfile window.)

EViews will open the file (automatically determining the file type, if possible, or interpreting the file as ASCII text, if not). Next, EViews may display a set of dialogs prompting you for information about the structure of the data you are reading. These dialogs are identical to those encountered when you open a file as a new workfile (see "Creating a Workfile by Reading from a Foreign Data Source" on page 47). Fill out the dialogs appropriately, clicking on **Next** at each step.

Once you have described the file and data, EViews will display a dialog prompting you for details on how you wish to perform the import of data:

There are two parts to the import specification: the choice of import method and the setting of various import options.

In the top-left portion of the dialog is the **Import method** dropdown, which controls how the source data will be read into the existing workfile. The area to the right of the dropdown will change to show options associated with the currently selected method.

Directly below the **Import method** dropdown is the **Import options** section, which offers access to settings for renaming series and specifying frequency conversion methods. Additionally, depending on the precise form of your import, you may be presented with additional dialogs for resizing the destination workfile or choosing how to handle import name conflict.

The bottom portion of the dialog consists of an import preview display, which shows a sample of destination observations along with corresponding data from the source file. The preview will change along with your selected import method specification.

We discuss all of these settings and choices below.

Import Method

The central question in importing data is how to relate observations in the source file to observations in the destination workfile. The **Import method** dropdown at the top left of the dialog should be used to specify the desired method. Depending upon the structure of the destination workfile, you will be asked to select between four or five import methods: **Dated read** (only if destination workfile is dated), **Matched Read**, **Sequential Read**, **Append to end**, and **Create new page**.

Each method has a variety of associated settings, so we consider each in turn.

Dated Read

You may perform a dated read if the destination workfile is a dated (or panel dated) workfile. In a dated read, observations in the source file are matched by date to observations in the current workfile page, with frequency conversion performed if necessary. The concepts underlying dated reads are identical to those in creating links between two dated workfile pages, as outlined in

To perform a dated read you must first specify the date structure of the source workfile. Accordingly, when you select **Dated read** in the **Import method** dropdown, the upper portion of the dialog will change to show you date settings for the data to be imported.

When the source file is first opened for dated read, EViews will attempt to identify a date series in the file (*i.e.* a column of data containing dates), and to determine the date frequency of those dates. If a source date series is located, the **Basic structure** dropdown will be set to **Dated - specified by date series** and the source name will be entered in the **Date series** edit field. If no date series is identified, the import file data structure will be set to match the destination.

Altering the specified date structure changes the nature of the import since it affects the matching of observations between the source and destination. To aid you in visualizing the effect of this setting, the left-hand column of the data preview display updates with changes in the structure to show you the current date matching settings:

	OBS	GDP	PR	M1	RS
unmatched	1958Q3	118.0167	0.227799	143.171	1.710667
unmatched	1958Q4	121.2750	0.228756	144.112	2.787667
1959M01	1959Q1	124.0750	0.229238	145.860	2.800333
1959M04	1959Q2	127.3250	0.229074	146.140	3.019333
1959M07	1959Q3	127.4000	0.229405	147.396	3.533000
1959M10	1959Q4	128.4250	0.230256	145.483	4.299333
1960M01	1960Q1	131.8250	0.231353	145.699	3.943000
1960M04	1960Q2	131.5250	0.232223	145.599	3.092333
1960M07					
1960M10					

Here, we see that the source observation with OBS value 1959Q1 is matched with the workfile destination observation 1959M1 and that the source observation with OBS = 1959Q2 is matched with the destination observation 1959M4. Note that there are no matches for the source observation OBS = 1958Q4 since the destination workfile begins in 1959.

If we were instead importing into an annual file, the bottom portion of the display would change to

	OBS	GDP	PR	M1	RS
unmatched	1958Q1	113.4750	0.225529	139.633	1.838000
	1958Q2	114.6000	0.226404	139.655	1.017667
	1958Q3	118.0167	0.227799	143.171	1.710667
	1958Q4	121.2750	0.228756	144.112	2.787667
1959	1959Q1	124.0750	0.229238	145.860	2.800333
	1959Q2	127.3250	0.229074	146.140	3.019333
	1959Q3	127.4000	0.229405	147.396	3.533000
	1959Q4	128.4250	0.230256	145.483	4.299333
1960					

depicting the matching of quarterly observations in the source file with annual observations in the destination workfile.

In both of these examples, importing of the data into the page requires frequency conversion. To control the frequency conversion method, you may click on the **Frequency Conversion button** on the left-hand side of the page to bring up the conversion settings.

As usual, you may control both the high-to-low and the low-to-high frequency conversion method. By default, EViews will use the method specified in the destination series (or the global defaults, if the destination series does not yet exist). When converting from high-to-low using a specific method such as Average observations, you will be
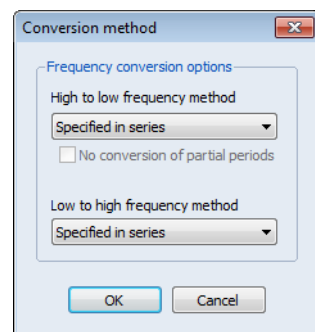
given the option of selecting the **No conversion of partial periods** checkbox so that no conversion is allowed if there are missing data for a given low frequency period.

Click on **OK** to close the dialog and accept the conversion method.

In most cases, EViews will have correctly identified the date series so that the default settings will require no change. Click on the **Finish** button complete the import specification and instruct EViews to begin the import procedure.
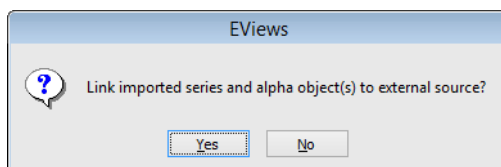
EViews will now ask if the new series objects should be linked back to the source. Click **Yes** to mark the new series objects as externally linked. Linked objects can be refreshed with updated data when the workfile is next opened, or on demand by right-clicking on the workfile window and selecting **Update from DB...** or by clicking on **Objects/Manage Links & Formulae...** in the main menu to display the link management dialog.

You may, if desired, override the default EViews date specification settings by changing the **Basic structure** dropdown menu setting to either **Dated - regular frequency** or **Dated panel**, and then using the dialog to specify the data frequency explicitly.

For example, if you select **Dated - regular frequency** in the **Basic structure** dropdown, the dialog changes to reflect the new settings:

Notice that EViews no longer shows the date series under **Basic structure**, instead prompting you to provide a **Frequency/date specification** using the **Frequency** dropdown menu and the **Start date** edit field. EViews will assume that the source file has sequential observations of the given frequency beginning in the specified date.

Since we have not yet specified a **Start date**, the "?"s in the observation preview at the bottom of the dialog indicate that the observation matching is indeterminate. Once a start date is specified, the display will be updated to depict matching observations.

If the destination workfile is panel structured, you may instead choose **Dated Panel** in the **Basic structure** dropdown.

When **Dated Panel** is selected, the import structure section of the dialog changes, prompting you to specify a **Cross section ID series** and a **Date series**. If you continue with the import, EViews will perform the date and ID matching as specified.
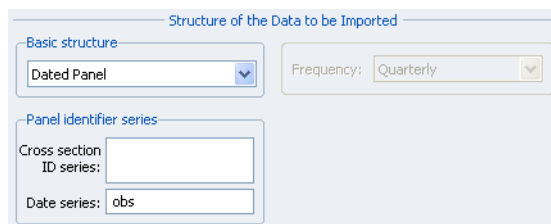
Matched Read

The **Matched Read** method performs a general match-merge between the data in the source file and the current workfile page. The concept is outlined in detail in "Linking by general match merging" on page 234.
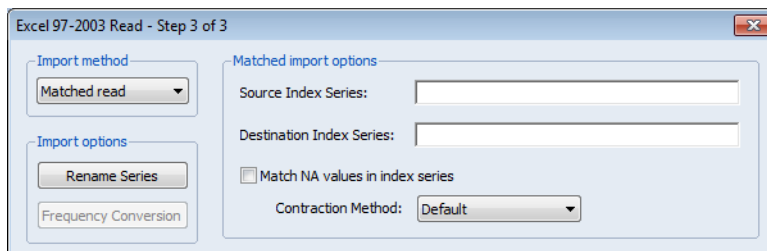
To perform a matched read import, change the **Import method** dropdown to **Matched read**, then specify the **Source Index Series** and corresponding **Destination Index Series** that will be used as define observation matches. In many cases, EViews will suggest likely values for the match indices determined by examining the contents of the source file and destination workfile.

Suppose for example, that your destination workfile has data on unemployment rates in each of the EU countries, you might have an identifying series called "COUNTRY" containing the country names. If you have a corresponding series in the source file, called "NATION," that also contains country names, then you would specify "NATION" as the source index, and "COUNTRY" as the destination index. Note that the observations in the source index need not be in the same order as they are in the destination.

In the simplest case, your index series will contain identical observation identifiers that can be used to match up the observations in the two files. If the matches are one-to-one or if there are multiple destination matches for every source observation (one-to-many), EViews will simply perform the merge in the obvious fashion, repeating source observations for every destination observation if necessary.

If, however, the matching is many-to-one so that there are multiple source observations for a given destination observation, EViews will contract the source data before performing the match merge. You may control the contraction method using the **Contraction Method** drop-down.

Suppose, for example, that your destination workfile contains a series COUNTRY containing the identifiers for "Mexico," "Canada," and "US", your source file contains U.S. state level data. You should enter COUNTRY as your **Destination Index Series** and indicate that your source should be matched with the "US" destination observation. (In the likely event that the source file does not have a COUNTRY series, you can create an auto-series by typing "US", with the quotes, in the **Source Index Series** box.) Note that in this case, the state level data must first be contracted (summarized) before it is matched to the "US" observation in the destination. To contract the source data by taking the sum of the states as a value for "US", you would set the **Contraction Method** to **Sum**.

The last dialog setting, **Match NA values in index series**, should be checked if you want to treat the NA value as a distinct category for purposes of defining matches.

The observation preview in the bottom of the dialog may be used to assess the validity of your source and destination index specifications.

Sequential Read

A sequential import uses no information on the structure of the source data, it simply places each observation in the source file into the corresponding observation of the destination workfile. Thus the first row of the source file will be placed in the first observation of the destination, the second row will be placed in the second observation, and so on.
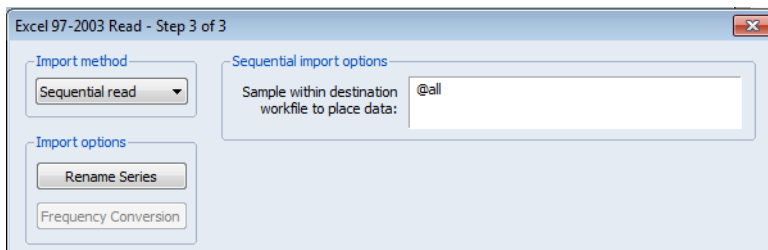
For a sequential read, simply select **Sequential read** in the **Import method** dropdown. There is only one setting; you may specify a destination sample in which to place the data, so that the first observation of the source file will be placed into the first observation of the specified sample, the second source observation will be placed in the second sample observation, and so on.

As before, the observation preview in the bottom of the dialog may be used to preview the destination for some of the source observations.

Append to End

The **Append to end** import method appends each row of the source file to the end of the destination workfile. The existing workfile structure will be removed.
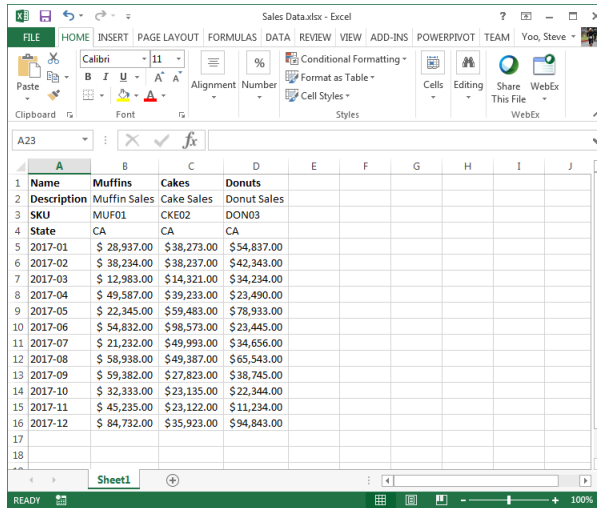
For example if your destination workfile is an annual file running from 1990 - 2000, and the source file contains 5 rows of data, the import will place the 5 rows of data at the end of the workfile, leaving you with an unstructured 16 observation workfile.

Create New Page

The **Create new page** method creates a new page in your workfile and reads the source file into that new page. This form of data import is identical selecting **Proc/Load workfile page...** in the workfile window or clicking on the **New Page** tab and selecting **Load workfile page...** For additional detail, see in *User's Guide I*.
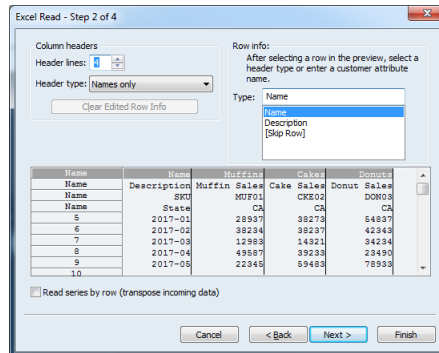
Importing Attributes

Support for reading in custom series attributes (along with the series data) specified as header rows during an import from Excel and text files was introduced in EViews 10. For example, the following Excel spreadsheet has custom attributes for Description, SKU, and State specified in header rows:

When we drag and drop this file onto EViews, the new Import Wizard dialog appears. On the second page (Step 2), you may change how each header row is used:

By default, EViews will treat every header row as a Name field, but you can change the row type for each header by selecting (or specifying a custom value for) the attribute name:
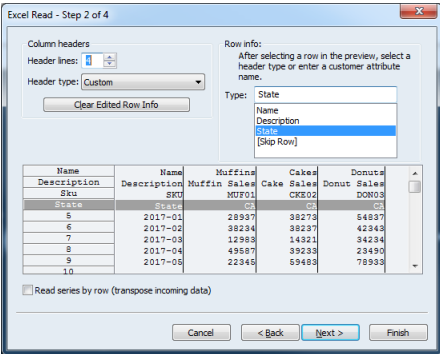
You can now complete the import and verify that all custom attributes have been read in correctly:
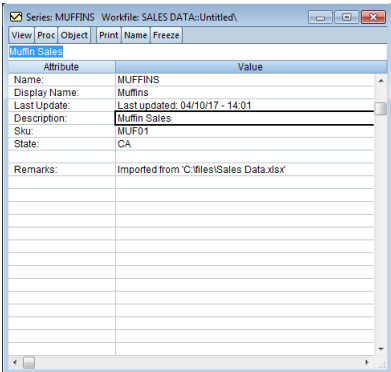
Both the `wfopen` and `import` commands have been modified to support new values for the "namepos = " argument and a new "colheadnames = " argument to specify individual attribute names for each column. See `wfopen` (p. 526) for more details.

Import Options

The **Import options** sections of the dialog consists of two buttons: **Rename series** and **Frequency conversion**.

Pressing the **Rename series** button brings up a dialog that allows you to rename some of the imported series to different names. Renaming the series on import allows you to handle illegal input series names, and to avoid name conflicts with existing series.
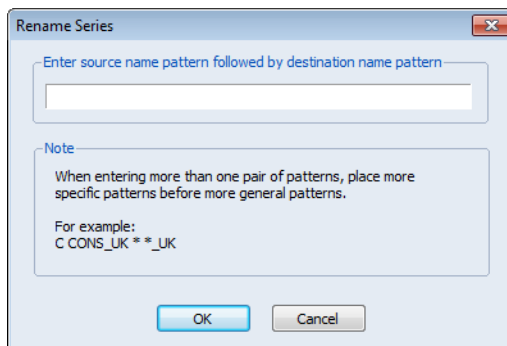
The **Frequency conversion** button, which brings up a dialog that controls the high-to-low and low-to-high frequency conversion methods, is enabled whenever you are performing a dated import. See "Dated Read," on page 152 for discussion.

Import Workfile Resizing

When you import data from a source file that contains observations for which there are no destination workfile matches, EViews will prompt you to accommodate the data by resizing the destination.

If, for example, your original workfile has observations from 1990M01 to 1994M12 and you import quarterly data for 1990Q1 to 1996Q4, EViews will determine that there are no destination observations corresponding to the source observations for 1995Q1 to 1996Q4. When you proceed with the import, EViews will display a dialog informing you of this fact.

There are two ways to proceed. First, you may click on **Yes** to resize the destination workfile, adding 24 monthly observations for 1995 to 1996. The merge will then proceed using all of the data in the source.
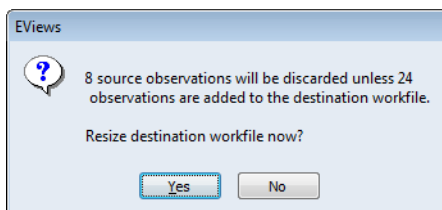
Alternately, you may click on **No** to retain the original destination workfile range. In this case, the 8 observations in the source file for 1995Q1 to 1996Q4 will be discarded on import, and will not appear in the final destination workfile.

Import Name Conflict

If you select an import method that brings source data into the current workfile page, and the source file contains series with the same name as an existing series, EViews will display a dialog asking how you wish to handle the conflict.

You may choose to overwrite the existing series object (which will delete the existing series and replace it with the source data), to merge the two series, to rename the incoming series, or to cancel the import of the series. Clicking on the **All** button performs the corresponding operation for all cases where there is conflict.
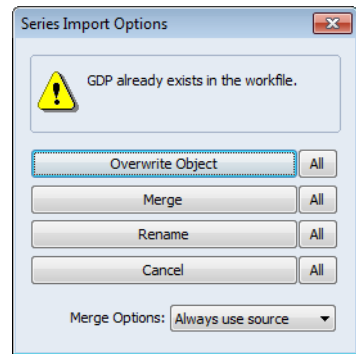
If you choose to **Merge** the source and destination series, EViews will use the **Merge Options** dropdown settings to determine how to join the data:

- **Always use source** will overwrite the destination series values for all observations in the source file, *including observations where the source contains an NA*.

- **Prefer source** will overwrite the destination series views with those from the source file, *excluding observations where the source contains an NA*.

- **Prefer destination** will only overwrite an observation in the destination file if its existing value is an NA.

Note that whatever the merge settings, observations in the destination workfile that are not also in the source file will be unchanged.

Matrix and Table Object Import

The preceding discussion focused on importing data into series or group objects. Similar tools are available for importing data directly into a matrix or table object from external data.

Exporting Data

EViews provides you with a number of methods for getting data from EViews into other applications.

Automatic Export using Save As

The easiest way to save series data is to use the save your workfile page as a foreign file. Simply select **File/Save As** from the main EViews menu, choose your destination file type and provide a name, then fill out the dialogs as prompted.

EViews will prompt you to specify the variables and object you wish to save. You will notice that the identifier series are listed first in the top portion of the dialog, with `@date` selected and `@enddate` unselected by default. You may also specify the sample of observations to be written, and for some output formats, the valmaps to be written.
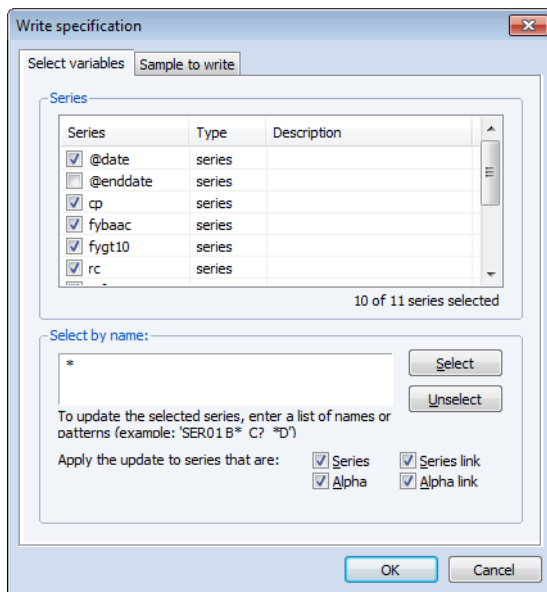
(See "Creating a Workfile by Reading from a Foreign Data Source" on page 47 for a discussion of the dialog controls.)

Once you have specified the variables and sample of observations to export, click on OK. EViews will write the specified data in the foreign data format.

Other Export Methods

For most applications, this method of saving data to a foreign format will be more than sufficient. The remainder of this section discusses alternate methods that support other object types. If, for example, you wish to export data held in a matrix object, you will need to use one of the methods described below.

Copying and Pasting

You can click and drag in a spreadsheet view or table of statistical results to highlight the cells you want to copy. Then click **Edit/Copy…** in the main menu to put the data into the clipboard. You will see a dialog box asking whether to copy the numbers with the precision showing on your screen (formatted copy) or to copy the numbers at full precision (unformatted copy).

As a shortcut, you can highlight entire rows or columns of cells by clicking on the gray border that surrounds the spreadsheet. Dragging across the border selects multiple rows or columns. To copy several adjacent series from the spreadsheet, drag across their names in the top border. All of their data will be highlighted. Then click **Edit/Copy…** to put the data into the clipboard.

Once the data are on the clipboard, switch to the target application, highlight the cells to which the data is to be copied and select **Edit/Paste**.

When pasting to a spreadsheet view or a table in EViews, if the paste cell range is larger than the copy range, the data will be repeated to fill the entire paste range. However, this will only occur if the paste range is proportional to copy range. Ranges are considered proportional when the paste range is a multiple of the copy range. For example, if a 3 by 1 area (3 rows by 1 column) is copied, the paste range must be at least 3 by 1. Proportional paste ranges could include 3 by 2, 6 by 1, 6 by 2, etc.

Exporting to a Spreadsheet or Text File

First, click on **Proc/Export/Write Text-Lotus-Excel…** from the workfile toolbar or main menu, then enter the name and type of the output file in the **SaveAs** dialog. As you fill out the **SaveAs** dialog, keep in mind the following behavior:

- If you enter a file name with an extension, EViews will use the file extension to identify the file type. Files with common spreadsheet extensions (".XLS", ".WK3", ".WK1", and ".WKS") will be saved to the appropriate spreadsheet type. All others will be saved as ASCII files.

- If you do not enter an extension, EViews will use the file type selected in the drop-down to determine the output type. Spreadsheet files will have the appropriate extensions appended to the name. ASCII files will be saved using the name provided in the dialog, without an extension. EViews will not append extensions to ASCII files unless you explicitly include one in the file name.

- Note that this method of export does not allow you to write into an existing file. If you wish to add to a file you must use the automatic export method outlined in "Automatic Export using Save As" on page 161.

Once you have specified the output file, click **OK** to open the export dialog.

The ability to save series object attributes when the destination is an Excel spreadsheet file or text file was introduced in EViews 10. Attributes will be saved as extra header rows. See wfsave (p. 542) for more details on the *attr* option.

Tip: if you highlight the series you wish to export before beginning the export procedure, the series names will be used to fill out the export dialog.

Reading EViews Data using Other Applications

The EViews OLEDB driver provides an easy way for OLEDB-aware clients or custom programs to read data stored in EViews workfiles (WF1) and EViews databases (EDB).

We also provide an EViews Microsoft Excel Add-in that allows users to fetch and link to EViews data located in workfiles and databases. The Add-in offers an easy-to-use interface to OLEDB for reading EViews data from within Excel.

The following discussion offers only a brief overview of the OLEDB driver and the Excel Add-in. For additional details, see the *Using the EViews OLEDB Driver* and *Using EViews Excel Add-in* whitepapers available on our website www.eviews.com/download/download.html.

The Excel Add-in

The EViews Excel Add-in offers a simple interface for fetching and linking from within Microsoft Excel (97 and later) to series and matrix objects stored in EViews workfiles and databases.

The method of installing and activating the EViews Excel Add-in our depends on which version of Excel is installed. Excel 97 thru 2003 requires the use of a single "EViews Add In.xla" file which is installed automatically by the EViews installer. Newer versions of Excel require the use of our new Excel 2007 Add-In, installation of which is offered as a separate option during the EViews installation procedure.

Once installed, you may need to activate the Add-in from within Microsoft Excel. Open the Excel Add-ins management dialog:
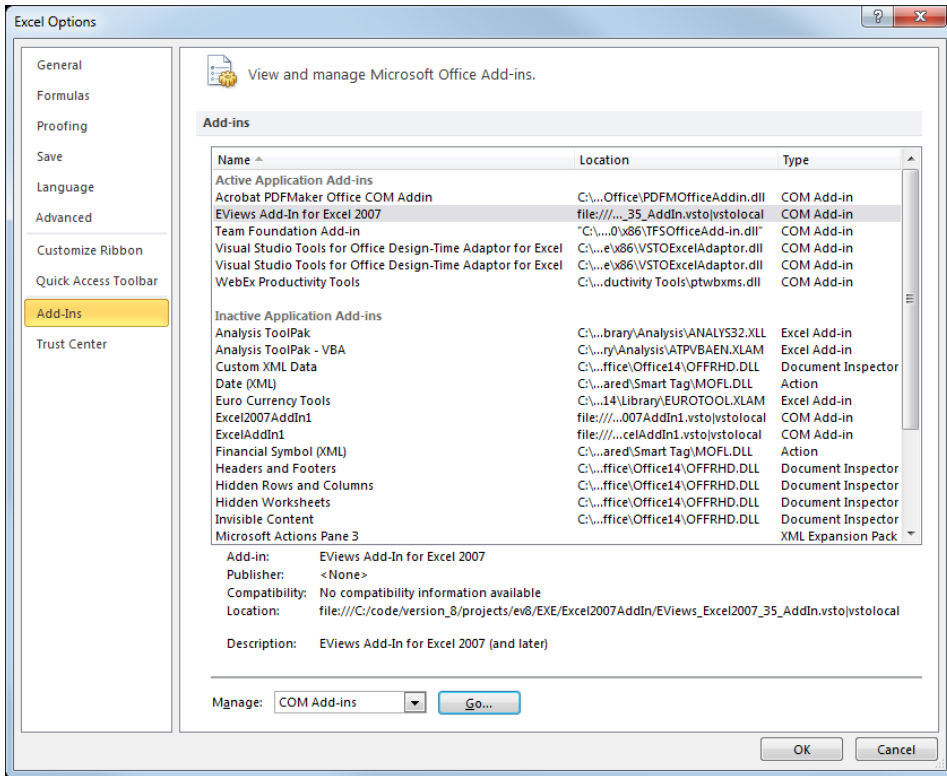
and use the **Manage** dropdown menu to select the EViews Excel Add-in. Excel 97-2003 users will find our add-in under "Excel Add-ins" in the Excel Add-ins management dialog:

while Excel 2007 and above users will find the newer add-in under "COM Add-ins",

Click the **Go...** button to display the Add-in selection dialog:
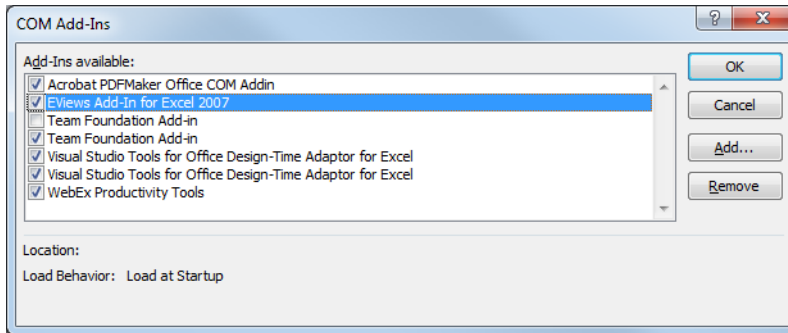
Enable the EViews Add-ins entry by clicking on the checkbox, then click on **OK**.

Once enabled, you may use the Add-in to retrieve EViews data. The following Excel 2007 instructions illustrate use of the EViews Add-in to retrieve data:

- From the **Add-ins** ribbon, select **Get Data** from the EViews section to display the **Get EViews Data** popup dialog. A **File Open** dialog will appear, prompting you to select an existing EViews workfile or database. If the specified source is a workfile, a **Page** dropdown menu will display all available pages; simply pick the page from which you wish to read. If the source is a database, a **Freq** dropdown menu will let you filter the series in the database by frequency.
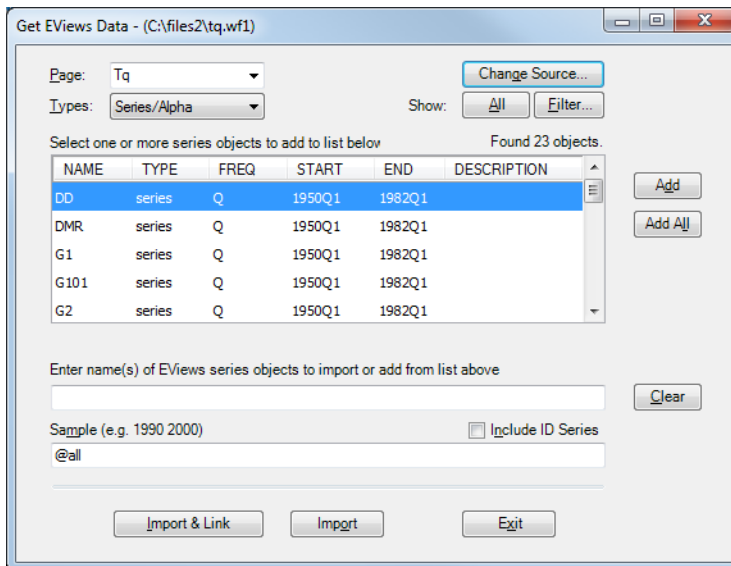
- You may use the **Types** dropdown menu to change the default series object filter so that you only consider matrix objects.

- You may further filter objects by name and description by clicking on the **Filter...** button and entering restrictions in the dialog:
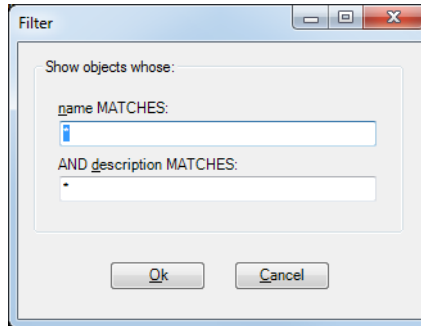
- You may select one or more series objects or a single matrix object and click on **Import** or **Import & Link** to continue. If you wish, you may instead click on **Add** to add selected items to the **Objects to import** list. Clicking on **Add All** adds all of the displayed items to the import list. You may also type in the name of the objects to import. Once your list is complete, click on **Import** or **Import & Link** to proceed. The import list approach is particularly useful if you wish to build-up a list of objects using various approaches or multiple filtered lists.

- The last step is to import or link the EViews data into Excel. On the resulting **Select Insert Point** popup, you should specify a destination cell in your Excel worksheet (you can click on a target cell to fill out the dialog). Click on **OK**.
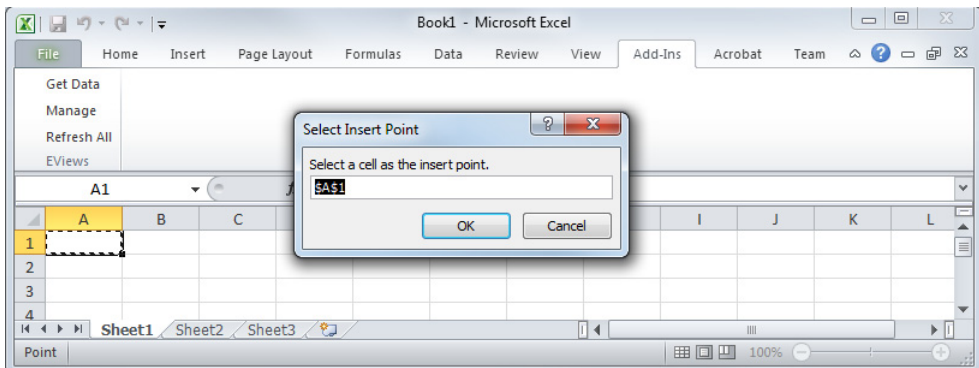
At this point, the data should now be displayed in the specified worksheet. If we select **Import & Link** instead of **Import** in the first step, this data will automatically refresh itself every time the worksheet is opened. (Note: this worksheet must be saved in a trusted location to allow the data to refresh properly. See the Microsoft Excel documentation regarding "Trusted File Locations" for details.)

The OLEDB Driver

The EViews OLEDB driver is automatically installed and registered on your computer when you install EViews. Once installed, you may use OLEDB-aware clients or custom programs to read series, vector, and matrix objects directly from EViews workfiles and databases.

For additional details, see the *Using the EViews OLEDB Driver* whitepaper.

The following instructions for Excel 2007 illustrate the use of built-in OLEDB support to read EViews workfile data. (Note that describes a streamlined interface for performing the same operation.)
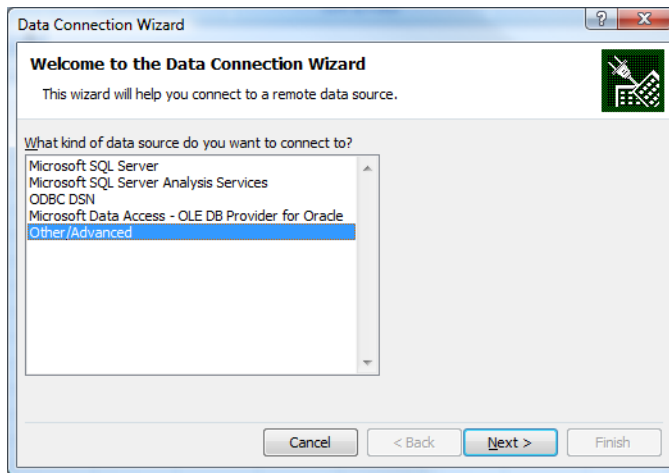
- From the Data ribbon, select **From Other Sources** in the **Get External Data** section. In the dropdown, select **From Data Connection Wizard**. In the **Data Connection Wizard**, select **Other/Advanced** in the listbox and click **Next**.
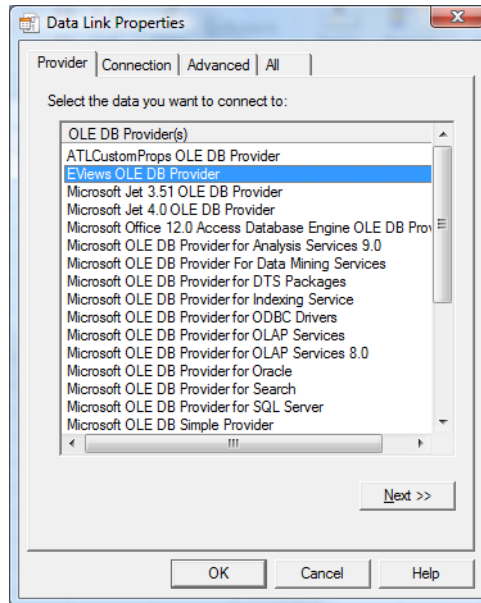
- In the **Data Link Properties** popup, select **EViews OLE DB Provider** and click **Next.**

- On the **Connection** tab, select an EViews workfile or database and click **OK** (not shown).
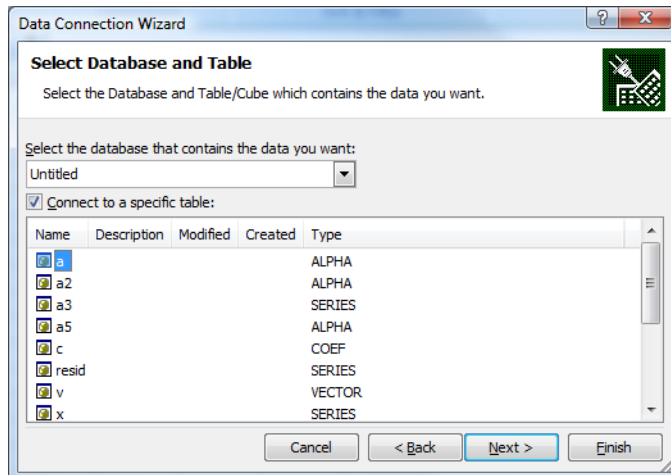
- Next, on the **Data Connection Wizard** popup, you should see a dropdown with all the available pages (if a workfile was selected) and in the grid below a list of readable series, vector, and matrix objects. Select one here and click **Finish**.
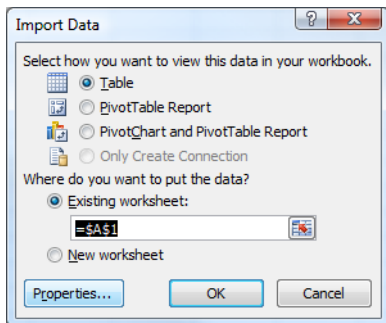
- On the **Import Data** popup, point to where you'd like to see the data in the worksheet and click **OK**.

- At this point, you should see the data (and any corresponding ID series data) appear in the specified worksheet

Frequency Conversion

Every series in EViews has an associated frequency. When a series is in a workfile, the series is stored at the frequency of the workfile. When a series is held in a database (Chapter 10. "EViews Databases"), it is stored at its own frequency. Since all series in the same workfile page must share a common frequency, moving a series from one workfile to another or from a database to a workfile page will cause the series being moved to be converted to the frequency of the workfile page into which it is being placed.

Performing Frequency Conversion

Frequency conversion is performed in EViews simply by copying or fetching a series with one frequency into a workfile of another frequency.

Copy-and-Paste

Suppose that you have two workfile pages (or a source database and a destination workfile page), where the source contains quarterly data on the series YQ, and the destination workfile contains annual data. Note that you may copy between pages in the same workfile or between separate workfiles.

To convert YQ from a quarterly to annual frequency, you may copy-and-paste the series from the source quarterly workfile to the annual workfile. Click on the YQ series in the quarterly workfile, press the right-mouse button and select **Copy**, navigate to the annual workfile, then right mouse button and select **Paste** or **Paste Special...**.

If you select **Paste**, EViews will copy YQ to the annual page, using the default frequency conversion settings present in YQ to perform the conversion.

If you select **Paste Special...**, EViews will display a dialog offering you the opportunity to override the default frequency conversion settings. Before describing this dialog ("Overriding Default Conversion Methods" on page 177), we provide a background on frequency conversion methods, and describe how default conversion methods are specified in EViews.

Drag-and-Drop

An alternative to copy-and-paste is to use drag-and-drop to copy files between workfile pages. Simply select the series to be copied, then drag-and-drop them on the destination page tab. If you use the right-mouse button to select the series, dropping them on the new tab will bring up the **Paste Special** dialog.

You can even drag-and-drop a workfile page tab into a workfile window to copy all of the objects from one workfile page into another. Depending on the objects being copied and the frequencies of the workfiles, you may receive a series of prompts to assist in completing the paste properly.

Using Commands

You may use either the `copy` or `fetch` command to move series between workfiles or between a database and a workfile. EViews will perform frequency conversion if the frequencies of the source and destination do not match.

See `copy` (p. 332) and `fetch` (p. 360) in the *Command and Programming Reference* for details.

Frequency Conversion Methods

There are three types of frequency conversion: high frequency to low frequency conversion, low frequency to high frequency conversion, and frequency conversion between a dated and undated workfile.

EViews provides you with the ability to specify methods for all types of conversion. In addition, there are settings that control the handling of missing values when performing the conversion.

High Frequency to Low Frequency

If a numeric series being imported has a higher frequency than the workfile, you may choose between a number of different conversion methods:

- Average observations
- Sum observations
- First observation
- Last observation

- Maximum observation

- Minimum observation

- No down conversions

with the latter setting permitting you to disallow high to low conversions. In this case, EViews will generate an error if you attempt to convert from high to low frequency.

In addition, you may specify how EViews handles missing data when carrying out the calculations. You may elect to propagate NAs so that whenever a missing value appears in a calculation, the result for the corresponding period will be an NA. Alternatively, you may elect not to propagate NAs so that calculations will be performed ignoring the missing values (though if all values for a period are missing, the corresponding result will still be an NA).

Low Frequency to High Frequency

EViews also provides a number of different interpolation methods for dealing with the case where the series being brought into the workfile has a lower frequency than the workfile. Since observing a series at a lower frequency provides fundamentally less information than observing the same series at a higher frequency, it is generally not possible to recover the high frequency series from the low frequency data. Consequently, the results from EViews' interpolation methods should be considered to be suggestive rather than providing the true values of the underlying series.

EViews supports the following interpolation methods:

- **Constant**: Constant with sum or average matched to the source data.

- **Quadratic**: Local quadratic with sum or average matched to the source data.

- **Linear**: Linear with first or last observation matched to the source data.

- **Cubic**: Cubic spline with first or last observation matched to the source data.

- **Point**: Copy the low frequency data into the first or last observation for the corresponding high frequency range.

- **Denton**: Statistical interpolation minimizing the proportional first difference between the interpolated and high frequency target series.

- **Chow-Lin**: Regression-based interpolation relating one or more high frequency target series to the low frequency series.

- **Litterman**: Random walk variant of Chow-Lin.

- **No up conversions**: Do not allow up conversion.

Using an interpolation method which matches the average means that the average of the interpolated points for each period is equal to the source data point for that period. Similarly if the sum is matched, the interpolated points will sum to the source data point for the

period, and if the last observation is matched, the last interpolated point will equal the source data point for the period.

For all methods, all relevant data from the low frequency series is used when forming the high frequency series, even if the destination observations are a subset of the observations available in the source.

The following describes the different methods in greater detail.

Constant

The constant method assigns the same value to all observations in the high frequency series associated with a particular low frequency period. If you match average, the value is chosen so that the average of the high frequency observation matches the low frequency observation (the value is simply repeated). If you match sum, the value is chosen so that the sum of the high frequency observations matches the low frequency observation (the value is divided by the number of observations).

Quadratic

Fits a local quadratic polynomial for each observation of the low frequency series, then use this polynomial to fill in all observations of the high frequency series associated with the period. The quadratic polynomial is formed by taking sets of three adjacent points from the source series and fitting a quadratic so that either the average or the sum of the high frequency points matches the low frequency data actually observed. For most points, one point before and one point after the period currently being interpolated are used to provide the three points. For end points, the two periods are both taken from the one side where data are available.

This method is a purely local method. The resulting interpolation curves are not constrained to be continuous at the boundaries between adjacent periods. Because of this, the method is better suited to situations where relatively few data points are being interpolated and the source data is fairly smooth.

Linear:

This method assigns each value in the low frequency series to the first or last high frequency observation associated with the low frequency period, then places all intermediate points on straight lines connecting these points.

Cubic

This method assigns each value in the low frequency series to the first or last high frequency observation associated with the low frequency period, then places all intermediate points on a natural cubic spline connecting all the points.

A natural cubic spline is defined by the following properties:

1. Each segment of the curve is represented by a cubic polynomial.

2. Adjacent segments of the curve have the same level, first derivative and second derivative at the point where they meet.

3. The second derivative of the curve at the two global end points is equal to zero (this is the "natural" spline condition).

Cubic spline interpolation is a global interpolation method so that changing any one point (or adding an additional point) to the source series will affect all points in the interpolated series.

Point

The point method simply copies the low frequency data into the first or last observation for the corresponding high frequency range.

Denton

This method finds an interpolated series x by relating a higher-frequency indicator series z to a lower-frequency benchmark series y. This is done by minimizing the proportional first difference function proposed by Denton (1971):

$$F = \sum_{t=1}^{T} \left(\frac{x(t)}{z(t)} - \frac{x(t-1)}{z(t-1)} \right)^2 \tag{5.1}$$

This can be solved as a constrained optimization problem subject to the constraints:

$$\sum_{t=b_k}^{e_k} x(t) = y_k \tag{5.2}$$

with the benchmark y in each period k, and b_k and e_k are the beginning and end of each period. For stock variables the constraints apply at specific points in time ($b_k = e_k$) where the interpolated series is equal to the benchmark y_k. For index and flow variables the average or sum of the interpolated series over each period k are equal to the benchmark y_k for that period. For flow variables this process is sometimes called *temporal distribution/disaggregation* instead of interpolation.

EViews uses Cholette's (1984) modification of Denton's original method to remove a transient movement at the beginning of the interpolated series.

The goal of Denton interpolation is movement preservation: the interpolated series x should preserve the movement in the indicator series z as much as possible. While there are many ways to specify the constraint problem, EViews uses the more popular proportional first difference in Equation (5.1) and Equation (5.2), which minimizes the proportional period-to-period change in the interpolated and indicator series. This produces an interpolated series that closely follows the growth rate of the indicator series.

Denton interpolation is a global interpolation method. Changing any point or adding points to the indicator or benchmark series will affect all points in the interpolated series. Also note that Denton interpolation can be performed without an indicator series in a procedure sometimes termed *benchmarking* (in practice, the indicator series z in Equation (5.1) is replaced by ones). In cases where the indicator series extends beyond the period covered by the endpoints of the benchmark series, x will be extrapolated with the last benchmark/indicator ratio.

Chow-Lin

The Chow-Lin method is a regression-based interpolation technique that finds values of a series x by relating one or more higher-frequency indicator series Z to a lower-frequency benchmark series through the equation

$$x(t) = Z(t)\beta + a(t) \tag{5.3}$$

where β is a vector of coefficients and $a(t)$ is a random variable with mean zero and covariance matrix V. The interpolated series is subject to the same constraints as in Equation (5.2) for stock, index, and flow variables. While the original solution by Chow and Lin (1971) used generalized least squares to estimate the covariance matrix assuming that the errors follow an AR(1) process, EViews uses a state space model with the following time series model for the states:

$$a(t) = \rho\, a(t-1) + \epsilon(t) \tag{5.4}$$

where $\epsilon(t) \sim N(0, \sigma^2)$ and $|\rho| < 1$. The parameters ρ and β are estimated via maximum likelihood and the Kalman filter, and the interpolated series is subsequently calculated with Kalman smoothing.

Chow-Lin interpolation is a global interpolation method, and changing any point or adding points to the indicator or benchmark series will affect all points in the interpolated series. Also note that Chow-Lin interpolation can be done without an indicator series (in practice, the indicator matrix Z in Equation (5.3) is replaced by a vector of ones). In cases where the indicator series extends beyond the period covered by the endpoints of the benchmark series, x will be extrapolated using the Kalman filter.

Litterman

For the Chow-Lin method the calculation of the interpolated series requires knowledge of the covariance matrix V, which is usually not known. Various techniques use various assumptions about the structure of V beyond the simplest (and unrealistic) case of homoskedastic, uncorrelated residuals. A common flavor of Chow-Lin is Litterman interpolation, in which the covariance matrix is calculated from residuals that follow:

$$a(t) = a(t-1) + \epsilon(t)$$

where $\epsilon \sim N(0, V)$ and

$$\epsilon(t) = \rho\epsilon(t-1) + e(t)$$

and the initial condition $a(0) = 0$. This is essentially an ARIMA(1,1,0) model.

Undated Conversion

If you fetch or copy a series to or from an undated or unstructured workfile into or from a dated workfile, the data will be copied sequentially, beginning at the starting observation number of the undated or unstructured series (generally the first observation).

Panel Conversion

Performing frequency conversion when either the source or destination are panel workfile pages raises a number of additional issues. See "Panel links with date matching" on page 241 and "Panel frequency conversion" on page 245 for discussion of the general issues.

Specifying Default Conversion Methods

When performing frequency conversion of one or more series, EViews uses the default settings in each series to perform the conversion. These settings may be specified in each series using the **Freq Convert** tab of the **Properties** dialog. To access the dialog, click on the **Properties** button on the series toolbar and select the **Freq Convert** tab.

If the series default setting is set to **EViews default**, the series will take its frequency conversion setting from the EViews global options (see "Frequency Conversion" on page 864 in Appendix A. "Global Options"). Here, the high to low conversion is set to **Sum observations**, overriding the global setting, while the low to high uses the **EViews default** global setting.

This two level default system allows you to set global default settings for frequency conversion that apply to all newly created series, while allowing you to override the default settings for specific series.
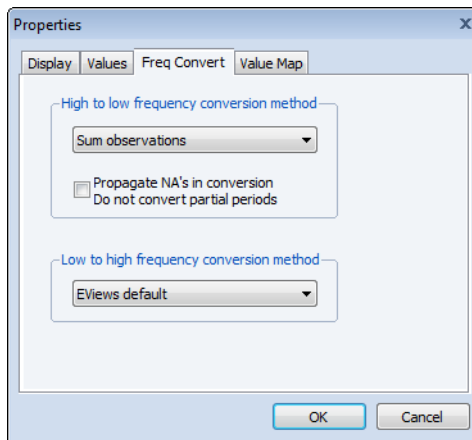
As an example of controlling frequency conversion using default settings, suppose you have daily data consisting of HIGH, LOW, and CLOSE series for a particular stock, from which you would like to construct a monthly workfile. If you use the default frequency conversion methods, the monthly workfile will contain series which use the series defaults, which is not likely to be what you want. By setting the frequency conversion method of the HIGH series to **Max observation**, of the LOW series to **Min observation**, and of the CLOSE series to **Last observation**, you may use conversion to populate a monthly workfile with converted daily data that follow the desired behavior.

Overriding Default Conversion Methods

If you use copy-and-paste to copy one or more series between two workfiles, EViews will copy the series to the destination page, using the default frequency conversion settings present in the series to perform the conversion.

If, when pasting the series into the destination, you use **Paste Special...** in place of **Paste**, EViews will display a dialog offering you the opportunity to override the default frequency conversion settings.

You need not concern yourself with most of the settings in this dialog at the moment; the dialog is discussed in greater detail in "Frequency conversion links" on page 254.

We note, however, that the dialog offers us the opportunity to change both the name of the pasted YQ series, and the frequency conversion method.

The "*" wildcard in the Pattern field is used to indicate that we will use the original name (wildcards are most useful when pasting multiple series). We may edit the field to provide a name or alternate wildcard pattern. For example, changing this setting to "*A" would copy the YQ series as YQA in the destination workfile.

Additionally, we note that the dialog allows us to use the frequency conversion method **Specified in series** or to select alternative methods.

If, instead of copy-and-paste, you are using either the `copy` or `fetch` command and you provide an option to set the conversion method, then EViews will use this method for all of the series listed in the command (see `copy` (p. 332) and `fetch` (p. 360) in the *Command and Programming Reference* for details).
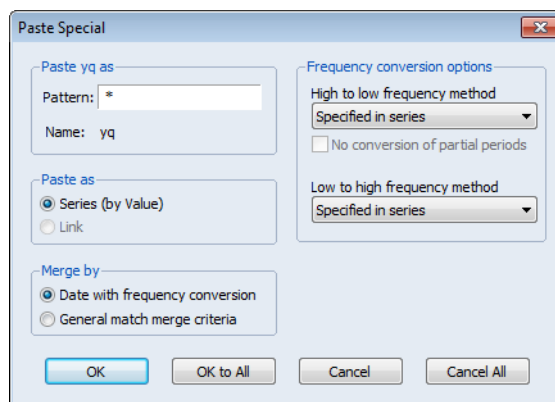
References

Denton, F. T. (1971). "Adjustment of monthly or quarterly series to annual totals: an approach based on quadratic minimization," *Journal of the American Statistical Association*, 66(333), 99-102.

Chen, B. (2007). "An empirical comparison of methods for temporal distribution and interpolation at the national accounts," *Bureau of Economic Analysis*.

Cholette, P. (1984). "Adjusting sub-annual series to yearly benchmarks," *Survey Methodology*, 10, 35-49.

Chow, G. C., & Lin, A. L. (1971). "Best linear unbiased interpolation, distribution, and extrapolation of time series by related series," *The Review of Economics and Statistics*, 53(4), 372-375.

Litterman, R. B. (1983). "A random walk, markov model for the distribution of time series," *Journal of Business and Statistics*, 1, 169-173.

Chapter 6. Working with Data

In the following discussion, we describe EViews' powerful language for using numeric expressions and generating and manipulating the data in series and groups. We first describe the fundamental rules for working with mathematical expressions in EViews, and then describe how to use these expressions in working with series and group data.

More advanced tools for working with numeric data, and objects for working with different kinds of data are described in Chapter 7. "Working with Data (Advanced)," beginning on page 203.

Numeric Expressions

One of the most powerful features of EViews is the ability to use and to process mathematical expressions. EViews contains an extensive library of built-in operators and functions that allow you to perform complicated mathematical operations on your data with just a few keystrokes. In addition to supporting standard mathematical and statistical operations, EViews provides a number of specialized functions for automatically handling the leads, lags and differences that are commonly found in time series data.

An EViews expression is a combination of numbers, series names, functions, and mathematical and relational operators. In practical terms, you will use expressions to describe all mathematical operations involving EViews objects.

As in other programs, you can use these expressions to calculate a new series from existing series, to describe a sample of observations, or to describe an equation for estimation or forecasting. However, EViews goes far beyond this simple use of expressions by allowing you to use expressions virtually anywhere you would use a series. We will have more on this important feature shortly, but first, we describe the basics of using expressions.

Operators

EViews expressions may include operators for the usual arithmetic operations. The operators for addition (+), subtraction (–), multiplication (*), division (/) and raising to a power (^) are used in standard fashion so that:

```
5 + 6 * 7.0 / 3
7 + 3e-2 / 10.2345 + 6 * 10^2 + 3e3
3^2 - 9
```

are all valid expressions. Notice that explicit numerical values may be written in integer, decimal, or scientific notation.

In the examples above, the first expression takes 5 and adds to it the product of 6 and 7.0 divided by 3 (5 + 14 = 19); the last expression takes 3 raised to the power 2 and subtracts 9 (9 – 9 = 0). These expressions use the order of evaluation outlined below.

The "–" and "+" operators are also used as the unary minus (negation) and unary plus operators. It follows that:

```
2-2
-2+2
2+++++++++++-2
2---2
```

all yield a value of 0.

EViews follows the usual order in evaluating expressions from left to right, with operator precedence order as follows (from highest precedence to lowest):

- unary minus (–), unary plus (+)

- exponentiation (^)

- multiplication (*), division (/)

- addition (+), subtraction (–)

- comparison (<, >, <=, >=, =)

- and, or

The last two sets of operators are used in logical expressions.

To enforce a particular order of evaluation, you can use parentheses. As in standard mathematical analysis, terms which are enclosed in parentheses are treated as a subexpression and evaluated first, from the innermost to the outermost set of parentheses. We strongly recommend the use of parentheses when there is any possibility of ambiguity in your expression.

To take some simple examples,

- `-1^2`, evaluates to $(-1)\verb|^|2 = 1$ since the unary minus is evaluated prior to the power operator.

- `-1 + -2 * 3 + 4`, evaluates to –1 + –6 + 4 = –3. The unary minus is evaluated first, followed by the multiplication, and finally the addition.

- `(-1 + -2) * (3 + 4)`, evaluates to –3 * 7 = –21. The unary minuses are evaluated first, followed by the two additions, and then the multiplication.

- `3*((2+3)*(7+4) + 3)`, evaluates to 3 * (5*11 + 3) = 3 * 58 = 174.

A full listing of operators is presented in Chapter 13. "Operator and Function Reference," on page 561 of the *Command and Programming Reference*.

Series Expressions

Much of the power of EViews comes from the fact that expressions involving series operate on every observation, or element, of the series in the current sample. For example, the series expression:

```
2*y + 3
```

tells EViews to multiply every sample value of Y by 2 and then to add 3. We can also perform operations that work with multiple series. For example:

```
x/y + z
```

indicates that we wish to take every observation for X and divide it by the corresponding observation on Y, and add the corresponding observation for Z.

Series Functions

EViews contains an extensive library of built-in functions that operate on all of the elements of a series in the current sample. Some of the functions are "element functions" which return a value for each element of the series, while others are "summary functions" which return scalars, vectors or matrices, which may then be used in constructing new series or working in the matrix language (see Chapter 11. "Matrix Language," on page 261 of the *Command and Programming Reference* for a discussion of scalar, vector and matrix operations).

Most function names in EViews are preceded by the @-sign. For example, @mean returns the average value of a series taken over the current sample, and @abs takes the absolute value of each observation in the current sample.

All element functions return NAs when any input value is missing or invalid, or if the result is undefined. Functions which return summary information generally exclude observations for which data in the current sample are missing. For example, the @mean function will compute the mean for those observations in the sample that are non-missing.

There is an extensive set of functions that you may use with series:

- A list of mathematical functions is presented in Chapter 13. "Operator and Function Reference," on page 561 of the *Command and Programming Reference*.

- Workfile functions that provide information about observations identifiers or allow you to construct time trends are described in Chapter 15. "Workfile Functions" of the *Command and Programming Reference*.

- Functions for working with strings and dates are documented in "String Function Summary" on page 631 and "Date Function Summary" on page 633 both in the *Command and Programming Reference*.

The remainder of this chapter will provide additional information on some of these functions, then examples of expressions involving functions.

Series Elements

At times, you may wish to access a particular observation for a series. EViews provides you with a special function, @elem, which allows you to use a specific value of a series.

@elem takes two arguments: the first argument is the name of the series, and the second is a quoted date or observation identifier.

For example, suppose that you want to use the 1980Q3 value of the quarterly series Y, or observation 323 of the undated series X. Then the functions:

```
@elem(y, "1980Q3")
@elem(x, "323")
```

will return the values of the respective series in the respective periods.

Numeric Relational Operators

Relational comparisons may be used as part of a mathematical operation, as part of a sample statement, or as part of an if-condition in programs.

A numeric relational comparison is an expression which contains the " = " (equal), " > = " (greater than or equal), " < = " (less than or equal), " < > " (not equal), " > " (greater than), or " < " (less than) comparison operators. These expressions generally evaluate to TRUE or FALSE, returning a 1 or a 0, depending on the result of the comparison.

Comparisons involving strings are discussed in "String Relational Operators," beginning on page 79 of the *Command and Programming Reference*.

Note that EViews also allows relational comparisons to take the value "missing" or NA, but for the moment, we will gloss over this point until our discussion of missing values (see "Missing Values" on page 187).

We have already seen examples of expressions using relational operators in our discussion of samples and sample objects. For example, we saw the sample condition:

```
incm > 5000
```

which allowed us to select observations meeting the specified condition. This is an example of a relational expression—it is TRUE for each observation on INCM that exceeds 5000; otherwise, it is FALSE.

As described above in the discussion of samples, you may use the "and" and "or" conjunction operators to build more complicated expressions involving relational comparisons:

```
(incm>5000 and educ>=13) or (incm>10000)
```

It is worth emphasizing the fact that EViews uses the number 1 to represent TRUE and 0 to represent FALSE. This internal representation means that you can create complicated expressions involving logical subexpressions. For example, you can use relational operators to recode your data:

```
0*(inc<100) + (inc>=100 and inc<200) + 2*(inc>=200)
```

which yields 0 if INC < 100, 1 if INC is greater than or equal to 100 and less than 200, and 2 for INC greater than or equal to 200.

The equality comparison operator "=" requires a bit more discussion, since the equal sign is used both in assigning values and in comparing values. We consider this issue in greater depth when we discuss creating and modifying series (see "Series" on page 189). For now, note that if used in an expression:

```
incm = 2000
```

evaluates to TRUE if INCOME is exactly 2000, and FALSE, otherwise.

Descriptive Statistics

Standard descriptive statistic functions are available in EViews. These include, but are not limited to functions to calculate the mean (@mean), the median (@median), the standard deviation (@stdev), the variance (@var) and covariance (@cov). The descriptive statistic functions all take an optional sample as an argument. For a full list of descriptive statistics functions, details on the use of samples, and some examples, see "Descriptive Statistics," on page 567 of the *Command and Programming Reference*.

It should be noted that EViews offers two ways to calculate standard deviations, variances and covariances. The simple standard deviation function, @stdev, calculates the sample standard deviation, that is the square root of the sum-of-squares divided by $n-1$. To calculate the population standard deviation, that is division by n, use the @stdevp function. Note for symmetry purposes there is also a @stdevs which performs the same calculation as @stdev.

The @var and @cov functions calculate the population variance and covariance respectively, *i.e.*, they divide through by n. To calculate the sample variance or covariance use the @vars or @covs functions. Again, there are also @varp and @covp functions which do the same as @var or @cov.

Leads, Lags, Differences and Time Series Functions

It is easy to work with lags or leads of your series. Simply use the series name, followed by the lag or lead enclosed in parentheses. Lags are specified as negative numbers and leads as positive numbers so that,

```
income(-4)
```

is the fourth lag of the income series, while:

```
sales(2)
```

is the second lead of sales.

While EViews expects lead and lag arguments to be integers, there is nothing to stop you from putting non-integer values in the parentheses. EViews will automatically convert the number to an integer; you should be warned, however, that the conversion behavior is not guaranteed to be systematic. If you must use non-integer values, you are strongly encouraged to use the @round, @floor, or @ceil functions to control the lag or lead behavior.

In many places in EViews, you can specify a range of lead or lag terms. For example, when estimating equations, you can include expressions of the form:

```
income(-1 to -4)
```

to represent all of the INCOME lags from 1 to 4. Similarly, the expressions:

```
sales sales(-1) sales(-2) sales(-3) sales(-4)
sales(0 to -4)
sales(to -4)
```

are equivalent methods of specifying the level of SALES and all lags from 1 to 4.

The @lag function can also be used to specify lags. Thus the expressions:

```
@lag(sales,1)
sales(-1)
```

are equivalent. Note one useful function of @lag is that it will take the lag of everything within parenthesis. @lag can therefore be used to find the lag of an expression. Typing:

```
@lag((sales-income)/sales,4)
(sales(-4)-income(-4))/sales(-4)
```

yields identical results.

EViews also has several built-in functions for working with difference data in either levels or in logs. The "D" and "DLOG" functions will automatically evaluate the differences for you. For example, instead of taking differences explicitly,

```
income - income(-1)
log(income) - log(income(-1))
```

you may use the equivalent expressions,

```
d(income)
dlog(income)
```

You can take higher order differences by specifying the difference order. For example, the expressions:

```
d(income,4)
dlog(income,4)
```

represent the fourth-order differences of INCOME and log(INCOME).

If you wish to take seasonal differences, you should specify both the ordinary, and a seasonal difference term:

```
d(income,1,4)
dlog(income,1,4)
```

These commands produce first order differences with a seasonal difference at lag 4. If you want only the seasonal difference, specify the ordinary difference term to be 0:

```
d(income,0,4)
dlog(income,0,4)
```

Other time series functions provided by EViews include a number of percentage change type functions. The simplest of these, @pc calculates a simple one-period percentage change in a series. For example typing:

```
@pca(income)
```

calculates the annual percentage change in INCOME.

Two special types of time series functions, moving functions and cumulative functions are also available in EViews, and are described below.

Mathematical details of lags, leads, differences and percentage change functions are provided in Chapter 13. "Operator and Function Reference," on page 561 of the *Command and Programming Reference*.

Cumulative and Moving Statistic Functions

Cumulative and moving statistic functions provide information over a range, or "window" of observations. The cumulative functions come in two types, those that move forwards and those that move backwards. The forwards functions, which take the form @cum[*stat*], have a window that starts at the start of the workfile (or if a sample is given in the function, from the start of the sample) up until the current observation.

The backwards functions, which take the form @cumb[*stat*], start at the end of the workfile, or sample, and move backwards until the current observation.

Note for both type of cumulative function the length of the window is different for each observation. The cumulative functions may be thought of as perform "running total" type calculations. Missing values are not propagated in the cumulative functions, *i.e.*, observations with a value equal to NA are simply skipped.

The moving statistic functions have a shorter, user specified, window length. They provide information on the n observations up to, and including, the current observation, where n is chosen by the user.

The moving functions come in two types, those that propagate missing values and those that do not. For the functions that do propagate missing values, which take the form @mov[*stat*], if any of the observations within the window contain an NA the function will return NA. The functions that do not propagate, which take the form @m[*stat*], will simply skip any NA observations.

For more information on missing values see "Missing Values" on page 187.

As an example, you could find out the maximum value of INCOME from the start of the workfile to each observation by typing:

```
show @cummax(income)
```

If the first, say, four observations of INCOME are 100, 120, 110, 140 then this command will show a series as 100, 120, 120, and 140 as the first four observations.

If you wanted to know at each observation the average of the previous 3 years (including the current year) SALES figures you could type:

```
show @movav(sales,3)
```

Note this is equal to:

```
show (sales + sales(-1) + sales(-2))/3
```

Note that the lag or lead operators can be used inside a moving statistic function to allow you to control the exact start and end point of your window. For example, if you wanted to know, at each observation, the sum of SALES from three years ago, two years ago and last year (*i.e.* the sum of SALES(-1), SALES(-2) and SALES(-3)) you could type:

```
show @movsum(sales(-1),3)
```

Further details and a complete list of cumulative functions can be found in "Cumulative Statistic Functions" on page 571, and for moving functions in "Moving Statistic Functions" on page 574 both in the *Command and Programming Reference*.

Ranking Series

EViews has an @ranks function which will generate a series based upon the ranking of another series. Ranking can be either ascending or descending depending upon whether "a"

or "d" is used as an option in the function. For example to create series, ARANK, which contains the ascending ranks of the observations in the series SALES you could type:

```
series arank = @ranks(sales,a)
```

and to create a series containing the descending ranks you could type:

```
series drank = @ranks(sales,d)
```

EViews provides a number of different ways of handling ties in the ranking. For more details see @ranks in "Descriptive Statistics" on page 567 of the *Command and Programming Reference*.

Missing Values

Occasionally, you will encounter data that are not available for some periods or observations, or you may attempt to perform mathematical operations where the results are undefined (*e.g.*, division by zero, log of a negative number). EViews uses the code NA (not available) to represent these missing values.

For the most part, you need not worry about NAs. EViews will generate NAs for you when appropriate, and will automatically exclude observations with NAs from statistical calculations. For example, if you are estimating an equation, EViews will use the set of observations in the sample that have no missing values for the dependent and all of the independent variables.

There are, however, a few cases where you will need to work with NAs, so you should be aware of some of the underlying issues in the handling of NAs.

First, when you perform operations using multiple series, there may be alternative approaches for handling NAs. EViews will usually provide you with the option of *casewise exclusion* (common sample) or *listwise exclusion* (individual sample). With casewise exclusion, only those observations for which *all* of the series have non-missing data are used. This rule is always used, for example, in equation estimation. For listwise exclusion, EViews will use the maximum number of observations possible for each series, excluding observations separately for each series in the list of series. For example, when computing descriptive statistics for a group of series, you have the option to use a different sample for each series.

If you must work directly with NAs, just keep in mind that EViews NAs observe all of the rules of IEEE NaNs. This means that performing mathematical operations on NAs will generate missing values. Thus, each of the following expressions will generate missing values:

```
@log(-abs(x))
1/(x-x)
(-abs(x))^(1/3)
3*x + NA
exp(x*NA)
```

For the most part, comparisons involving NA values propagate NA values. For example, the commands:

```
series y = 3
series x = NA
series equal = (y = x)
series greater = (y > x)
```

will create series EQUAL and GREATER that contain NA values, since the comparison between observations in a series involving an NA yields an NA.

Note that this NA handling behavior differs from EViews 4 and earlier in which NAs were treated as ordinary values for purposes of equality (" = ") and inequality (" < > ") testing. In these versions of EViews, the comparison operators " = " and " < > " always returned a 0 or a 1. The change in behavior was deemed necessary to support the use of string missing values. In all versions of EViews, comparisons involving ordering (" > ", " < ", " < = ", " > = ") propagate NAs.

It is still possible to perform comparisons using the previous methods. One approach is to use the special functions @eqna and @neqna for performing equality and strict inequality comparisons without propagating NAs. For example, you may use the commands:

```
series equal1 = @eqna(x, y)
series nequal = @neqna(x, y)
```

so that NAs in either X or Y are treated as ordinary values for purposes of comparison. Using these two functions, EQUAL1 will be filled with the value 0, and NEQUAL will be filled with the value 1. Note that the @eqna and @neqna functions do not compare their arguments to NA, but rather facilitate the comparison of values so that the results are guaranteed to be 0 or 1. See also "Version 4 Compatibility Mode" on page 176 of the *Command and Programming Reference* for settings that enable the previous behavior for element comparisons in programs.

To test whether individual observations in a series are NAs, you may use the @isna function. For example,

```
series isnaval = @isna(x)
```

will fill the series ISNAVAL with the value 1, since each observation in X is an NA.

There is one special case where direct comparison involving NAs does not propagate NAs. If you test equality or strict inequality against the literal NA value:

```
series equal2 = (x = NA)
series nequal2 = (y <> NA)
```

EViews will perform a special test against the NA value without propagating NA values. Note that these commands are equivalent to the comparisons involving the special functions:

```
series equal3 = @eqna(x, NA)
series nequal3 = @neqna(y, NA)
```

If used in a mathematical operation, a relational expression resulting in an NA is treated as an ordinary missing value. For example, for observations where the series X contains NAs, the mathematical expression

```
5*(x>3)
```

will yield NAs. However, if the relational expression is used as part of a sample or IF-statement, NA values are treated as FALSE.

```
smpl 1 1000 if x>y
smpl 1 1000 if x>y and not @isna(x) and not @isna(y)
```

are equivalent since the condition x>3 implicitly tests for NA values. One consequence of this behavior is that:

```
smpl 1 1000 if x<NA
```

will result in a sample with no observations since less-than tests involving NAs yield NAs.

Very early versions of EViews followed the IEEE rules for missing data with one important exception. In EViews 2 and earlier, multiplying any number by zero (including NAs) yielded a zero. In subsequent versions, the value NA times zero equals NA. Thus, an earlier recommended method of recoding (replacing) NA values in the series X no longer worked so that the command for replacing NA values with the values in Y:

```
x = (x<>na)*x + (x=na)*y
```

works in EViews 2, but does not work subsequent versions. The `@nan` function has been provided for this purpose.

```
x = @nan(x,y)
```

recodes NA values of X to take the values in the series Y. See of the *Command and Programming Reference*.

Series

One of the primary uses of expressions is to generate new series from existing data or to modify the values in an existing series. Used in combination with samples, expressions allow you to perform sophisticated transformations of your data, saving the results in new or existing series objects.

The current discussion focuses on the basic numeric series object. Users who wish to work with alphanumeric or advanced series features should see Chapter 7. "Working with Data (Advanced)," on page 203 and Chapter 8. "Series Links," on page 233.

To create or modify a series, select **Quick/Generate Series…** or click on the **Genr** button on the workfile toolbar. EViews opens a window prompting you for additional information.

You should enter the assignment statement in the upper edit box, and the relevant sample period in the lower edit box.

The assignment statement is actually an implicit loop over observations. Beginning with the first observation in the sample, EViews will evaluate the assignment statement for each included observation.
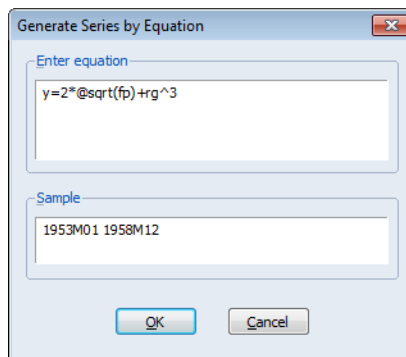
Basic Assignment

You can type the series name, followed by an equal sign and then an expression. For every element of the sample, EViews will evaluate the expression on the right-hand side of the equality, and assign the value to the *destination series* on the left-hand side, creating the series if necessary.

For example, if there is no series named Y,

```
y = 2*x + 37*z
```

will first create the Y series and fill it with NAs. Then, for every observation in the current sample, EViews will fill each element of the Y series with the value of the expression. If Y does exist, EViews will only replace Y values in the current sample with the value of the expression. All observations not in the sample will be unchanged.

One special form of assignment occurs when the right-hand side of the assignment statement is a constant expression:

```
y = 3
y = 37 * 2 + 3
```

EViews will simply assign the value of the constant to all of the observations in the sample.

Using Samples

By modifying the sample of observations used in assignment, you can splice together series using multiple series assignment commands. For example, if we enter three assignment commands with different samples: first

Upper window: `y = z`

Lower window: `@all if z<=1 and z>-1`

followed by an assignment with,

    Upper window: `y = -2 + 3*z`

    Lower window: `if z>1`

and finally,

    Upper window: `y = -.9 + .1*z`

    Lower window: `if z<=-1`

we can generate Y as a piecewise linear function of the series Z. Note that the "@ALL" is implicit in the latter two assignments.

While it is possible to perform these types of operations using loops and IF-statements (see Chapter 6. "EViews Programming," on page 117 of the *Command and Programming Reference*), we strongly urge you to use `genr` and sample statements where possible, since the latter approach is much more efficient.

Dynamic Assignment

Since EViews evaluates the assignment expression for each observation in the sample, you can perform dynamic assignment by using lagged values of the destination series on the right side of the equality. For example, suppose we have an annual workfile that ranges from 1945 to 1997. Then if we enter:

    Upper window: `y = y + y(-1)`

    Lower window: `1946 1997`

EViews will replace the Y series with the cumulative sum of Y. We begin with 1946, since we do not want to transform the first value in the workfile. Then for each period, EViews will take the current value of Y and add it to the lagged value of Y. The assignment is dynamic because as we successively move on to the next period, the lagged value of Y contains the cumulative sum.

Note that this procedure destroys the original data. To create a new series with the cumulative sums, you will have to perform the assignment in two steps, first making a copy of the original series, and then performing the dynamic assignment.

Implicit Assignment

You can make an implicit assignment by putting a simple formula on the left-hand side of the equal sign. EViews will examine your expression and select, as the destination series, the first valid series name on the left-hand side of the equality. Then for every observation in the sample, EViews will assign values using the implicit relationship. For example, if you enter:

    `log(y) = x`

EViews will treat Y as the destination series, and evaluate `y=exp(x)` for every observation in the sample.

The following are examples of valid assignment statements where Y is the destination series:

```
1/y = z
log(y/x)/14.14 = z
log(@inv(y)*x) = z
2+y+3*z = 4*w
d(y) = nrnd
```

In general, EViews can solve for, or *normalize*, equations that use the following on the left-hand side of the equality: `+, –, *, /, ^, log(), exp(), sqr(), d(), dlog(), @inv()`.

Since **Genr** is not a general equation solver, there will be situations in which EViews cannot normalize your equation. You cannot, for example, use the assignment statement:

```
@tdist(y, 3) = x
```

since `@tdist` is not one of the functions that EViews knows how to invert. Similarly, EViews cannot solve for equations where the destination series appears more than once on the left side of the equality. For example, EViews cannot solve the equation:

```
x + 1/x = 5
```

In both cases, EViews will display the error message "Unable to normalize equation".

Note that the destination series can appear on both sides of the equality. For example:

```
log(x) = x
```

is a legal assignment statement. EViews will normalize the expression and perform the assignment

```
x = exp(x)
```

so that X will be assigned the exponential of the original value of X. EViews will *not* solve for the values of X satisfying the equality "LOG(X) = X".

Using the Command Window

You can create series and assign values from the command window. First, set the workfile sample using the `smpl` statement, then enter the assignment statement.

There are alternative forms for the assignment statement. First, if the series does not exist, you must use either the `series` or the `genr` keyword, followed by the assignment expression. The two statements:

```
series y = exp(x)
genr y = exp(x)
```

are equivalent methods of generating the series Y. Once the series has been created, subsequent assignment statements do not require the `series` or the `genr` keyword:

```
smpl @all
series y = exp(x)
smpl 1950 1990 if y>300
y = y/2
```

This set of commands first sets the series to equal EXP(X) for all observations, then assigns the values Y/2 for the subset of observations from 1950 to 1990 if Y > 300.

Auto-series

Another important method of working with expressions is to use an expression *in place of* a series. EViews' powerful tools for expression handling allow you to substitute expressions virtually any place you would use a series—as a series object, as a group element, in equation specifications and estimation, and in models.

We term expressions that are used in place of series as *auto-series*, since the transformations in the expressions are *automatically* calculated without an explicit assignment statement.

Auto-series are most useful when you wish to see the behavior of an expression involving one ore more series, but do not want to keep the transformed series, or in cases where the underlying series data change frequently. Since the auto-series expressions are automatically recalculated whenever the underlying data change, they are never out-of-date.

See for a more advanced method of handling series and expressions.

Creating Auto-series

It is easy to create and use an auto-series—anywhere you might use a series name, simply enter an EViews expression. For example, suppose that you wish to plot the log of CP against time for the period 1953M01 to 1958M12. There are two ways in which you might plot these values.

One way to plot these values is to generate an ordinary series, as described earlier in , and then to plot its values. To generate an ordinary series containing the log of CP, say with the name LOGCP, select **Quick/Generate series...** from the main menu, and enter,

```
logcp = log(cp)
```

or type the command,

```
series logcp = log(cp)
```

in the command window. EViews will evaluate the expression LOG(CP) for the current values of CP, and will place these values into the series LOGCP. To view a line graph view of the series, open the series LOGCP and select **View/Graph/Line**.

Note that the values of the ordinary series LOGCP will not change when CP is altered. If you wish to update the values in LOGCP to reflect subsequent changes in CP, you will need to issue another `series` or `genr` assignment statement.
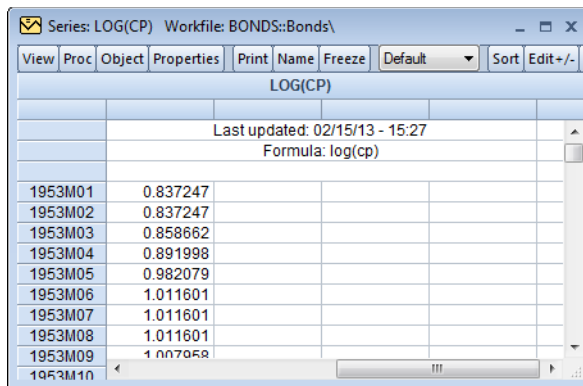
Alternatively, you may create and use an auto-series by clicking on the **Show** button on the toolbar, or selecting **Quick/Show…** and entering the command,

```
log(cp)
```

or by typing

```
show log(cp)
```

in the command window. EViews will open a series window in spreadsheet view:

Note that in place of an actual series name, EViews substitutes the expression used to create the auto-series.

An auto-series may be treated as a standard series window so all of the series views and procedures are immediately available. To display a time series graph of the LOG(CP) auto-series, simply select **View/Graph…** from the series window toolbar and click **OK** to create a line graph:
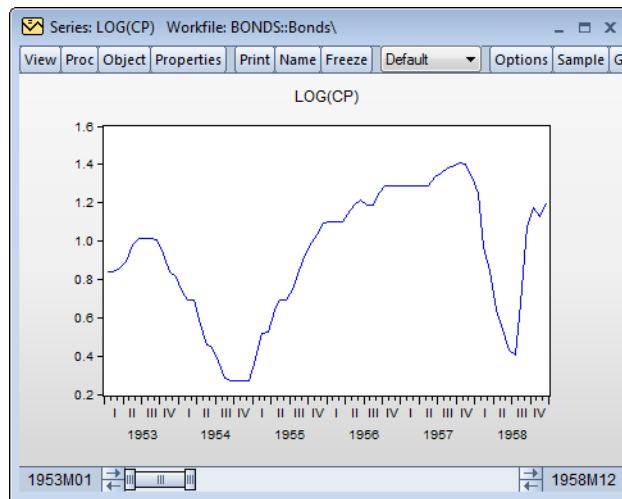
All of the standard series views and procedures are also accessible from the menus.

Note that if the data in the CP series are altered, the auto-series will reflect these changes. Suppose, for example, that we take the first four years of the CP series, and multiply theme by a factor of 10:

```
smpl 1953m01 1956m12
cp = cp*10
smpl 1953m01 1958m12
```

The auto-series graph will automatically change to reflect the new data:
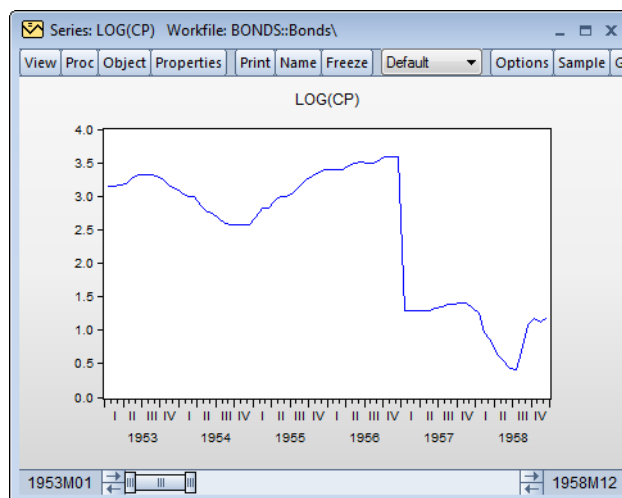
In contrast, the values of the ordinary series LOGCP are not affected by the changes in the CP data.

Similarly, you may use an auto-series to compute a 12-period, backward-looking, geometric moving average of the *updated* CP data. The command:

```
show @exp(@movav(@log(cp),12))
```

will display the auto-series containing the geometric moving average. To view a line graph of the auto-series, select **View/Graph...** from the series window toolbar and click **OK**:
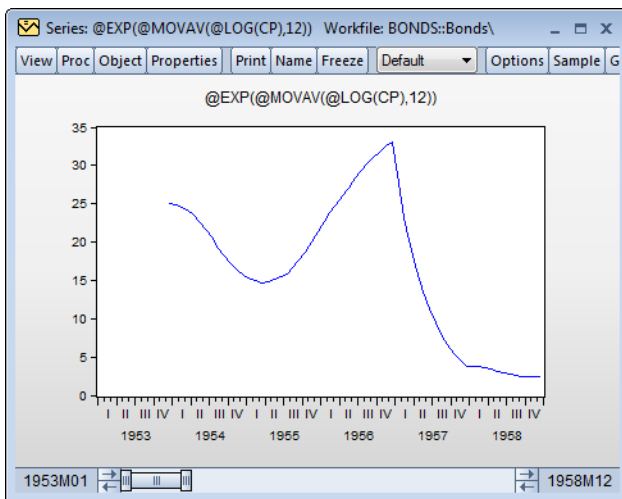
Auto-series Across Pages

You may define use series in different workfile pages using the syntax *pagename\seriesname*. Examples of useful places to use this functionality include group definitions, equation specifications, or when generating a new series.

If the two workfile pages are of differing frequency, EViews will use the default frequency conversion method for the series to convert the frequencies. There is no way to specify a different conversion method for auto-series. If you wish to fine-tune the frequency conversion, you must create a series link (Chapter 8. "Series Links," on page 233) or copy the series from the other page. Notably, auto-series defined between panel and non-panel workfiles may not frequency convert as intended so we recommend the use of links in this setting.

An auto-series across pages may only be used to refer to the series itself and may not be used in an auto-series expression. You may not, for example, use an expression which takes the log of an auto-series across pages as a regressor in an equation. You may, however, use the auto-series in a series assignment, as in

```
series myser = log(otherpage\yourser)
```

Naming an Auto-series

The auto-series is deleted from your computer memory when you close the series window containing the auto-series. For more permanent expression handling, you may convert the auto-series into an auto-updating series that will be kept in the workfile, by assigning a name to the auto-series.

Simply click on the **Name** button on the series toolbar, or select **Object/Name...** from the main menu, and provide a name. EViews will create an auto-updating series with that name in the workfile, and will assign the auto-series expression as the formula used in updating the series. For additional details, see "Auto-Updating Series" on page 203.

Using Auto-series in Groups

One of the more useful ways of working with auto-series is to include them in a group. Simply create the group as usual, using an expression in place of a series name, as appropriate. For example, if you select **Object/New Object.../Group**, and enter:

```
cp @exp(@movav(@log(cp),12))
```

you will create a group containing two series: the ordinary series CP, and the auto-series representing the geometric moving average. We may then use the group object graphing routines to compare the original series with the smoothed series. Select **View/Graph...** from the group window toolbar and click **OK**:

"Groups" on page 198 below describes other useful techniques for working with auto-series.

Using Auto-Series in Estimation

One method of using auto-series in estimation is to allow expressions as right-hand side variables. Thus, you could estimate an equation with $\log(x)$ or $\exp(x+z)$ as an explanatory variable.
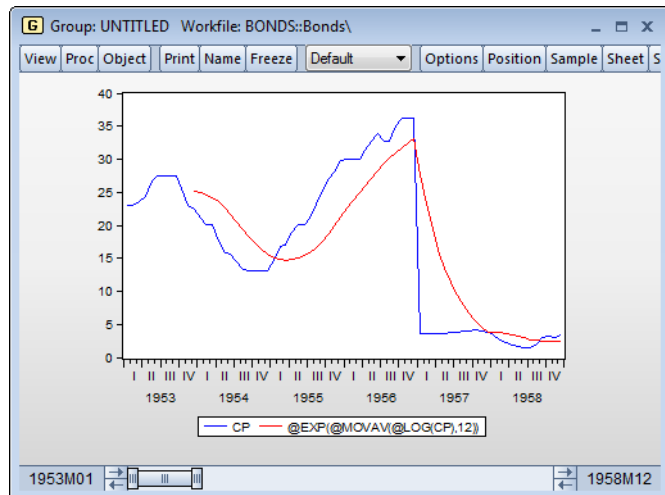
EViews goes a step beyond this use of auto-series, by allowing you to use auto-series as the dependent variable in estimation. Thus, if you want to regress the log of Y on explanatory variables, you don't have to create a new variable LOGY. Instead, you can use the expression $\log(y)$ as your dependent variable.

When you forecast using an equation with an auto-series dependent variable, EViews will, if possible, forecast the untransformed dependent variable and adjust the estimated confidence interval accordingly. For example, if the dependent variable is specified as `log(y)`, EViews will allow you to forecast the level of Y, and will compute the asymmetric confidence interval. See Chapter 23. "Forecasting from an Equation," on page 147 of *User's Guide II* for additional details.

Groups

EViews provides specialized tools for working with groups of series that are held in the form of a group object. In "Importing Data" on page 146, we used groups to import data from spreadsheets into existing workfiles. Briefly, a group is a collection of one or more series identifiers or expressions. Note that a group does not contain the data in the individual series, only references to the data in the series.

To create a group, select **Object/New Object…/Group** and fill in the dialog with names of series and auto-series. Or you may select **Show** from the workfile toolbar and fill out the dialog. Alternatively, type the command `group` in the command window, followed by a name to be given to the group and then the series and auto-series names:

```
group macrolist gdp invest cons
```

creates the group MACROLIST containing the series GDP, INVEST and CONS. Similarly,

```
group altlist log(gdp) d(invest) cons/price
```

creates the group ALTLIST containing the log of the series GDP, the first difference of the series INVEST, and the CONS series divided by the PRICE series.

There are a few features of groups that are worth keeping in mind:

- A group is simply a list of series identifiers. It is not a copy of the data in the series. Thus, if you change the data for one of the series in the group, you will see the changes reflected in the group.

- If you delete a series from the workfile, the series identifier will be maintained in all groups. If you view the group spreadsheet, you will see a phantom series containing NA values. If you subsequently create or import the series, the series values will be restored in all groups.

- Renaming a series changes the reference in every group containing the series, so that the newly named series will still be a member of each group.

- There are many routines in EViews where you can use a group name in place of a list of series. If you wish, for example, to use X1, X2, and X3 as right-hand side variables in a regression, you can instead create a group containing the series, and use the group in the regression.

We describe groups in greater detail in Chapter 12. "Groups," on page 547.

Accessing Individual Series in a Group

Groups, like other EViews objects, contain their own views and procedures. For now, note that you can access the individual elements of a named group as individual series.

To refer the n-th series in the group, simply append "(n)" to the group name. For example, consider the MACROLIST group, defined above. The expression MACROLIST(1) may be used to refer to GDP and MACROLIST(2) to refer to INVEST.

You can work with MACROLIST(1) as though it were any other series in EViews. You can display the series by clicking on the **Show** button on the toolbar and entering MAC-ROLIST(1). You can include GDP in another group directly or indirectly. A group which contains:

```
macrolist(1) macrolist(2)
```

will be identical to a group containing

```
gdp invest
```

We can also use the individual group members as part of expressions in generating new series:

```
series realgdp = macrolist(1)/price
series y = 2*log(macrolist(3))
```

or in modifying the original series:

```
series macrolist(2) = macrolist(2)/price
```

Note that in this latter example the series keyword is required, despite the fact that the INVEST series already exists. This is true whenever you access a series as a member of a group.

Other tools allow you to retrieve the number of series in a group using the @count group data member:

```
scalar numgroup = macrolist.@count
```

To retrieve the names of each of the series, you may use the group data member @seriesname. These tools are described in greater detail in "Group" on page 290 of the *Object Reference*.

Group Row Functions

EViews allows you to generate a series based upon the rows, or observations, in a group. The most simple of these is the @columns function which simply returns a series where every observation is equal to the number of series in a group. This function provides exactly the same information as the @count data member of a group. Thus the expression:

```
series numgroup = @columns(macrolist)
```

produces the same result as:

```
series numgroup = macrolist.@count
```

There are also functions that will calculate the mean of a group's rows (`@rmean`), their standard deviation (`@rstdev`) and variance (`@rvar`).

The `@rvalcount` function can be used to find how many times a specific value occurs within the rows of a group. For example:

```
series numvals = @valcount(macrolist,5)
```

will create a series where each row of that series will be the count of how many of the series within the MACROLIST group contain the value "5" for that particular row. Note that the value argument for this function can be a scalar or a series.

A full list of the group row functions can be found in of the *Command and Programming Reference.*

Creating a Group By Expanding a Series

The `@expand` function allows you to create a group of dummy variables by expanding out one or more series into individual categories. For example, if the series UNION contains values equal to either "union", "non-union", then using:

```
group g1 @expand(union)
```

will create a group, G1, with two series, the first series containing 1 where-ever union is equal to "union" and zero elsewhere, the second series containing 1 where-ever union is equal to "non-union" and zero elsewhere.

`@expand` may also be used on more than one series to give the cross-interaction of different series. Thus if you have a second series called MARRIED that contains either "married" or "single" then entering:

```
group g2 @expand(union,married)
```

will create a group, G2, with four series, the first containing 1 where-ever UNION is equal to "union" and MARRIED is equal to "married", the second series containing a 1 where-ever UNION is equal to "union" and MARRIED is equal to "single", and so on.

The `@expand` function can be used as part of a mathematical expression, so that a command of:

```
group g3 2*@expand(union)
```

will create a group where the first series contains a 2 where-ever UNION is equal to "union". Further,

```
group g4 log(income)*@expand(married)
```

creates a group where the first series is equal to the values of the log of INCOME where-ever MARRIED is equal to "married" and so on.

The of the most useful applications of the `@expand` function is when specifying an equation object, since it can be used to automatically create dummy variables.

See also "Automatic Categorical Dummy Variables" on page 28 of *User's Guide II* for additional discussion.

An Illustration

Auto-series and group processing provides you with a powerful set of tools for working with series data. As we saw above, auto-series provide you with dynamic updating of expressions. If we use the auto-series expression:

```
log(y)
```

the result will be automatically updated whenever the contents of the series Y changes.

A potential drawback of using auto-series is that expressions may be quite lengthy. For example, the two expressions:

```
log(gdp)/price + d(invest) * (cons + invest)
12345.6789 * 3.14159 / cons^2 + dlog(gdp)
```

are not suited to use as auto-series if they are to be used repeatedly in other expressions.

You can employ group access to make this style of working with data practical. First, create groups containing the expressions:

```
group g1 log(gdp)/price+d(invest)*(cons+invest)
group g2 12345.6789*3.14159/cons^2+dlog(gdp)
```

If there are spaces in the expression, the entire contents should be enclosed in parentheses.

You can now refer to the auto-series as G1(1) and G2(1). You can go even further by combining the two auto-series into a single group:

```
group myseries g1(1) g2(1)
```

and then referring to the series as MYSERIES(1) and MYSERIES(2). If you wish to skip the intermediate step of defining the subgroups G1 and G2, make certain that there are no spaces in the subexpression or that it is enclosed in parentheses. For example, the two expressions in the group ALTSERIES,

```
group altseries (log(gdp)/price) 3.141*cons/price
```

may be referred to as ALTSERIES(1) and ALTSERIES(2).

Scalars

Scalar objects are different from series and groups in that they hold a single number instead of data for each observation in the sample. Scalars are created by commands of the form:

```
scalar scalar_name = number
```

where you assign a number to the scalar name. The number may be an expression or special functions that return a scalar.

To examine the contents of a scalar, you may enter the command show, followed by the name of the scalar. For example:

```
scalar logl1 = eq1.@logl
show logl1
```

stores the log likelihood value of the equation object named EQ1 in a scalar named LOGL1, and displays the scalar window.

Chapter 7. Working with Data (Advanced)

In addition to the basic tools for working with numeric data outlined in Chapter 6. "Working with Data," EViews provides additional tools and objects for more advanced data handling, or for working with different kinds of data.

Auto-Updating Series

One of the most powerful features of EViews is the ability to use a series expression in place of an existing series. These expressions generate auto-series in which the expression is calculated when in use, and automatically recalculated whenever the underlying data change, so that the values are never out of date.

Auto-series are designed to be discarded after use. The resulting downside to auto-series is that they are quite transitory. You must, for example, enter the expression wherever it is used; for example, you must type "LOG(X)" every time you wish to use an auto-series for the logarithm of X. For a single use of a simple expression, this requirement may not be onerous, but for more complicated expressions used in multiple settings, repeatedly entering the expression quickly becomes tedious.

For more permanent series expression handling, EViews provides you with the ability to define a series or alpha object that uses a formula. The resulting auto-updating series is simply an EViews numeric series or alpha series that is defined, not by the values currently in the object, but rather by an expression that is used to compute the values. In most respects, an auto-updating series may simply be thought of as a named auto-series. Indeed, naming an auto-series is one way to create an auto-updating series.

The formula used to define an auto-series may contain any of the standard EViews series expressions, and may refer to series data in the current workfile page, or in EViews databases on disk. It is worth emphasizing that in contrast with link objects, which also provide dynamic updating capabilities, auto-updating series are designed to work with data in a single workfile page.

Auto-updating series appear in the workfile with a modified version of the series or alpha series icon, with the numeric series icon augmented by an " = " sign to show that it depends upon a formula.

Defining an Auto-Updating Series

Using the Dialog

To turn a ordinary series into an auto-
updating series, you will assign an expres-
sion to the series and tell EViews to use
this expression to determine the series val-
ues. Simply click on the **Properties** button
on the series or alpha series object toolbar,
or select **View/Properties...** from the main
menu, then select the **Values** tab.

There are three radio buttons which con-
trol the values that will be placed in the
numeric or alpha series ("Alpha Series,"
beginning on page 208). The default set-
ting is either **Numeric data** or **Alphanu-
meric (text) data** (depending on the series
type) in which the series is defined by the values currently in the series; this is the tradi-
tional way that one thinks of defining a numeric or alpha series.
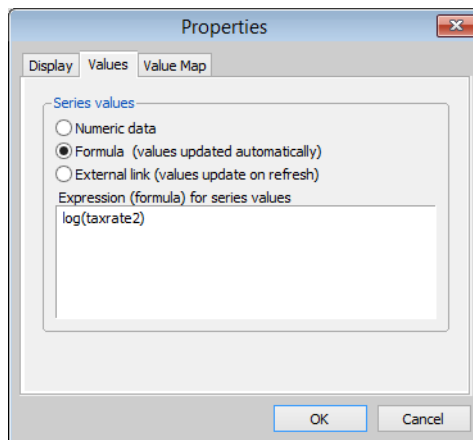
If instead you select **Formula**, enter a valid series expression in the dialog box, and click on
OK, EViews will treat the series as an auto-updating series and will evaluate the expression,
putting the resulting values in the series. Auto-updating numeric series appear with a new
icon in the workfile—a slightly modified version of the standard series icon, featuring the
series line with an extra equal sign, all on an orange background.

Lastly, **External link** indicates that the series is linked to data found outside of EViews as
described in the link specification. You will be prompted to update data in external links
whenever the workfile is opened, and you may update the external series links on demand
by right-clicking on the series, and selecting **Update...** or clicking on CTRL-F5.

In this example, we instruct
EViews that the existing series
LOGTAXRT should be an auto-
updating series that contains the
natural logarithm of the TAX-
RATE2 series. As with an auto-
series expression, the values in
LOGTAXRT will never be out of
date since they will change to
reflect changes in TAXRATE2. In
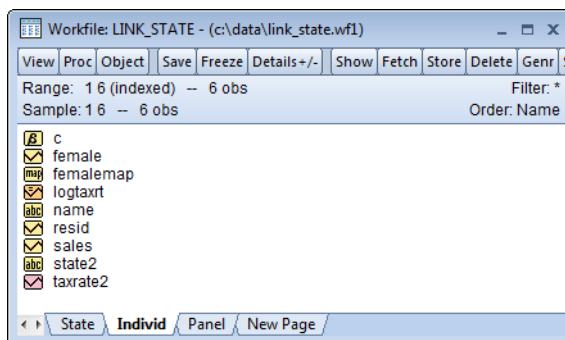
contrast to an auto-series, however, LOGTAXRT is a permanent series in the workfile which may be used like any other series.

Alternately, you may create an auto-updating series that links to a series in a external file or database by selecting **External link**. In contrast to an auto-updating series based on a formula which updates whenever the underlying data change, an auto-updating series based on an external link will update only when the workfile is first loaded (you will be prompted for whether to refresh the data series or not) or when you manually update the links by clicking on **Object/Manage Links & Formulae…** in the workfile window, then either selecting the source database above or selecting the individual objects that are linked to that database and then clicking the associated **Refresh** button. You may also update the link by selecting **Update…** in the series **Object** menu.

You may, at any time, change an auto-updating series into an standard numeric series by bringing up the **Values** page of the **Properties** dialog, and clicking on the **Numeric data** setting. EViews will define then define the series by its current values. In this way you may freeze the formula series values at their existing values, a procedure that is equivalent to performing a standard series assignment using the provided expression.

Note that once an expression is entered as a formula in a series, EViews will keep the definition even if you specify the series by value. Thus, you make take a series that has previously been frozen, and return it to auto-updating by selecting **Formula** definition.

Issuing a Command

To create an auto-updating series using commands, you should use the formula keyword, `frml`, followed by an assignment statement. The following example creates a series named LOW that uses a formula to compute values. The auto-updating series takes the value 1 if either INC is less than or equal to 5000 or EDU is less than 13, and takes the value 0 otherwise:

```
frml low = inc<=5000 or edu<13
```

LOW is now an auto-updating series that will be reevaluated whenever INC or EDU change.

You may also define auto-updating alpha series using the `frml` keyword. If FIRST_NAME and LAST_NAME are alpha series, then the declaration:

```
frml full_name = first_name + " " + last_name
```

creates an auto-updating alpha series, FULL_NAME.

The same syntax should be used when you wish to apply a formula to an existing series.

```
series z = rnd
frml z =(x+y)/2
```

makes Z an auto-updating series that contains the average of series X and Y. Note that the previous values of Z are replaced, and obviously lost. Similarly, we may first define an alpha series and then apply an updating formula:

```
alpha a = "initial value"
frml a = @upper(first_name)
```

You may not, however, apply an alpha series expression to a numeric series, or vice versa. Given the series Z and A defined above, the following two statements:

```
frml z = @upper(first_name)
frml a = (x+y)/2
```

will generate errors.

Note that once a numeric series or alpha series is defined to be auto-updating, its values may not be modified directly, since they are determined from the formula. Thus, if Z is an auto-updating series, the assignment command:

```
z = log(x)
```

will generate an error since an auto-updating series may not be modified. To modify Z you must either issue a new `frml` assignment or you must first set the values of Z to their current values by turning off auto-updating, and then issue the assignment statement.

To reset the formula in Z, you may simply issue the command:

```
frml z = log(x)
```

to replace the formula currently in the series.

To turn off auto-updating for a series, you may use the special expression "@CLEAR" in your `frml` assignment. When you turn off auto-updating, EViews freezes the numbers or strings in the series at their current values. Once the series is set to current values, it is treated as an ordinary series, and may be modified as desired. Thus, the commands:

```
frml z = @clear
z = log(x)
```

are allowed since Z is converted into an ordinary series prior to performing the series assignment.
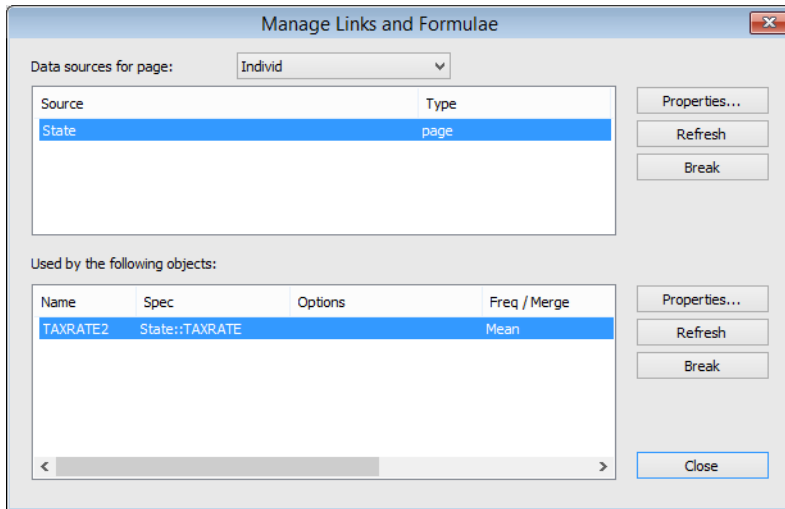
Alternately, you may convert a named auto-updating series into an ordinary series by selecting **Object/Manage Links & Formulae...** from the workfile window and using the dialog to break the links in the auto-updating series.

One particularly useful feature of auto-updating series is the ability to reference series in databases. The command:

```
frml gdp = usdata::gdp
```

creates a series in the workfile called GDP that gets its values from the series GDP in the database USDATA. Similarly:

```
frml lgdp = log(usdata::gdp)
```

creates an auto-updating series named LGDP that contains the log of the values of GDP in the database USDATA.

Series that reference data in databases may be *refreshed each time a workfile is loaded from disk*. Thus, it is possible to setup a workfile so that its data are current relative to a shared database.

Naming an Auto-Series

If you have previously opened a window containing an ordinary auto-series, you may convert the auto-series into an auto-updating series by assigning a name. To turn an auto-series into an auto-updating series, simply click on the **Name** button on the toolbar, or select **Object/Name...** from the main menu, and enter a name. EViews will assign the name to the series object, and will apply the auto-series definition as the formula to use for auto-updating.

Suppose, for example, that you have opened a series window containing an auto-series for the logarithm of the series CP by clicking on the **Show** button on the toolbar, or selecting **Quick/Show…** and entering "LOG(CP)". Then, simply click on the **Name** button in the auto-series toolbar, and assign a name to the temporary object to create an auto-updating series in the workfile.

Additional Issues

Auto-updating series are designed to calculate their values when in use, and automatically update values whenever the underlying data change. An auto-updating series will assign a value to every observation in the current workfile, irrespective of the current values of the workfile sample.

In most cases, there is no ambiguity in this operation. For example, if we have an auto-updating series containing the expression "LOG(CP)", we simply take each observation on CP in the workfile, evaluate the log of the value, and use this as the corresponding auto-updating series value.

However, in cases where the auto-updating series contains an expression involving descriptive statistics, there is ambiguity as to whether the sample used to calculate the values is the sample at the time the auto-updating series was created, the sample at the time the series is evaluated, the entire workfile range, or some other sample.

To resolve this ambiguity, EViews will enter the current workfile sample into the expression at the time the auto-updating series is defined. Thus, if you enter "@MEAN(CP)" as your auto-updating series expression, EViews will substitute an expression of the form "@MEAN(CP, *smpl*)" into the definition. If you wish to evaluate the descriptive statistics for a given sample, you should enter an explicit sample in your expression.

Alpha Series

An *alpha series* object contains a set of observations on alphanumeric string values. Alpha series should be used when you wish to work with variables that contain alphanumeric data, such as names, addresses, and other text. If any of these types of data were entered into an ordinary series, EViews will replace the string with the numeric missing value, NA.

You may, for example, have an alpha series that contains the two-character U.S. Postal Service abbreviations for the 50 states, D.C., and Puerto Rico. Here, we show the alpha series, STATE, that contains the appropriate 2-character string values. STATE will be identified in the workfile with the alpha series icon labeled "abc", and by the designation **Alpha** in the titlebar of the alpha series window.
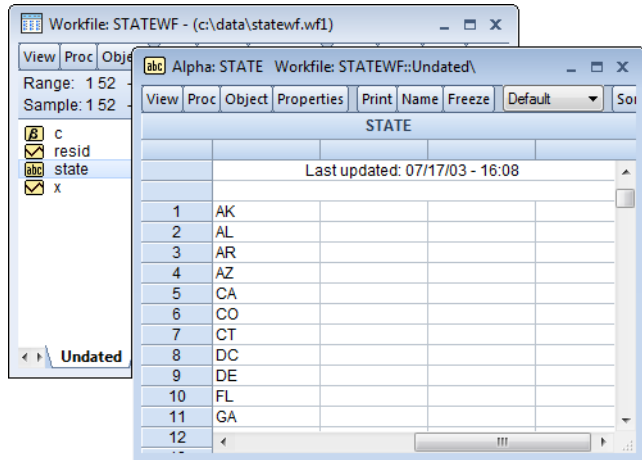
Similarly, alpha series may be used to hold identifying information such as the names and addresses of individuals, social security and telephone numbers, or classifying labels such as "male" and "female", or "high", "medium", and "low".

Declaring an Alpha Series

To create a new alpha series, you may select **Object/New Object...** from the main EViews window or workfile button bar, and then click on **Series Alpha** and optionally enter a name to be given to your alpha series. If you provide a name, EViews will create a new alpha series object in the workfile. If you do not supply a name, EViews will open an UNTITLED alpha series window.

Alternatively, you may type the keyword "ALPHA", followed by an optional series name, in the command window. The command:

```
alpha
```

will create a new untitled alpha series and will display the series in an object window. Likewise:

```
alpha myseries
```

will create a new alpha series MYSERIES. To open the alpha series windows for MYSERIES or SETSERIES, simply double-click on the corresponding alpha series icon in the workfile window directory, or enter the command "SHOW MYSERIES".

In both of the cases described above, the alpha series will be initialized to contain missing values. For alpha series, the empty string (the null string, "") is used to designate a missing value. If you are declaring an alpha series using a command, you may combine the declara-

tion with the assignment of the values in the series. We explore alpha series assignment in .

For the most part, you need not worry about the lengths of string values in your alpha series since EViews will automatically resize your series as required, up to the limit specified in the global defaults. Beyond that point, EViews will truncate the values of the alpha series. To modify the truncation length, select **Options/General Options...** from the main menu and go to the **Series and Alphas/Alpha truncation** page. Enter the desired length, and subsequent alpha series creation and assignment will use the new truncation length.
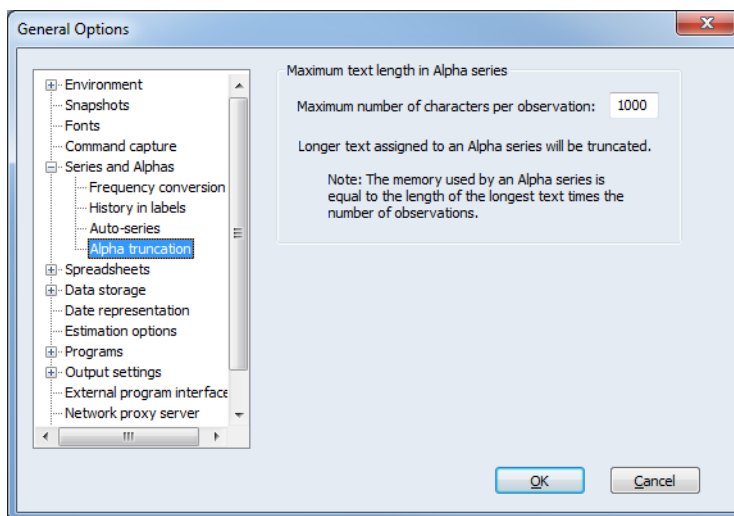
You should bear in mind that the strings in EViews alpha series are of fixed length so that the size of each observation is equal to the length of the longest string. If you have a series with all short strings with the exception of one very long string, the memory taken up by the series will be the number of observations times the longest string. In settings of this sort, efficiency suggests that you consider using value maps () to encode the values of the long string.

Editing an Alpha Series

There is no difference between editing an ordinary numeric series and editing an alpha series. Make certain that the alpha series is in edit mode by verifying the existence of the edit field in the series window. If not, click on the **Edit +/−** button to enable edit mode.

To edit a specific value, click on
the desired cell. The existing
value in the cell will appear in the
edit window for you to modify or
delete.

Simply type the new value in the
edit window. Once you have
entered the desired value, move to
a new cell by clicking or using the
arrow keys, or press the return
key. This action will accept the
entered value and prepare you for
editing the newly selected cell.
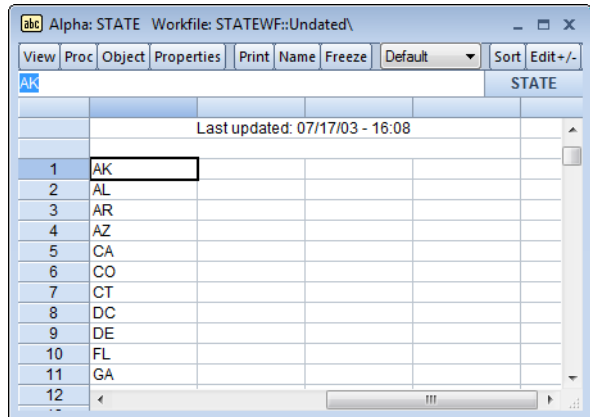
Note that when editing the values of an alpha series, EViews does not require you to delimit
your strings. You may simply type the relevant value in the edit field. EViews will remove
any leading and trailing spaces from the value that you enter; if you wish to retain those
characters, enclose your string in double quotes. To enter the double quote character as a
part of your string, you should escape the character with another double quote so that you
enter two consecutive double quotes.

Assigning values to Alpha Series

You may assign values to an alpha series using string expressions. An alpha series assign-
ment has the form:

> *alpha_name = string_expr*

where *alpha_name* is the name of an existing alpha series and *string_expr* is any expression
containing a combination of string literals, alpha series, and functions or operators that
return strings (see "Strings" on page 77 of the *Command and Programming Reference* for
details). As with ordinary series, we may combine the declaration and assignment steps so
that the commands:

> `alpha` *alpha_name = string_expr*

or

> `genr` *alpha_name = string_expr*

first create the alpha series *alpha_name* and then will assign the values using *string_expr*. In
the latter command, EViews notes that the right-hand side expression is a string so that it
knows to create an alpha series.

Alternatively, assuming that the alpha series exists, you may reassign the series values by clicking on **Quick/Generate Series...** in the main menu and entering the assignment and sample statements in the dialog. For example, if you enter the expression:

```
myalpha = string_expr
```

in the dialog, EViews will assign the values of the *string_expr* to the existing alpha series MYALPHA. Alternatively, you may enter the expression in the command line. In both cases, EViews will assign the corresponding values for all observations in the current workfile sample, overwriting the existing values.

Let us consider a simple example. Suppose that we have data on the company name (NAME), ticker symbol (SYMBOL), time of last trade (LAST_TIME), and closing price (LAST_TRADE) for each of the stocks in the Dow Jones Industrial Average on September 10, 2003.

Clicking on the icon for NAME, we can display the alpha series spreadsheet view. Note here that the default column width is not wide enough to display the contents of every observation, a condition that is signaled by the trailing "..." in the display for several of the observations. We may increase the column width by dragging the column header separators (the lines separating the column headers located just below the name of the series), by clicking on the **Properties** button and entering a larger number in the width field, or by double clicking on the column header separator to adjust the column width to the minimum width that displays all of the observation values without truncation.

Suppose now that we wish to create an alpha series containing the name of each company followed by its ticker symbol (enclosed in parentheses). A simple assignment statement generates the desired series:

```
alpha namesymb = name + " (" + symbol + ")"
```

EViews will create a new alpha series NAMESYMB if one doesn't exist. Then, for every observation in the workfile sample, the contents of the alpha series NAME are concatenated with the literal strings for the parentheses, and the contents of the SYMBOL series.
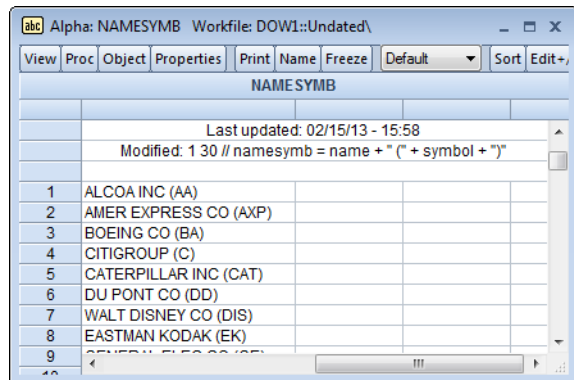
Working with Alpha Series

Once created, an alpha series is used in two primary ways: (1) to generate numeric values and (2) to provide identifiers for the observations in the workfile.

Generating Numeric Values

By definition, an alpha series contains a string value for each observation. This means that if you use an alpha series in a setting requiring numeric input, all of the values of the alpha series will be treated as NAs. For example, if you attempt to compute the mean of the STATE alpha series or use the Dow company NAME in an equation regression specification, EViews will generate an error saying that there are an insufficient number of observations, since all of the numeric values are missing.

You may, however, use the string relational operators (see "String Relational Operators" on page 79 of the *Command and Programming Reference*) to generate a series of numeric values. For the data from our Dow Jones example, the commands:

```
smpl @all
series wname = (@lower(@left(NAME, 1)) = "w")
```

generate the numeric series WNAME containing the value 1 if the company name begins with the letter "W", and 0 otherwise.

Similarly, the relational operators may be used when specifying a subsample. The command:

```
smpl @all if gender = "Male"
```

will restrict the workfile sample to include only observations where the string value in the alpha series GENDER is "Male".

You may also use the various functions described in "String Information Functions" on page 82 of the *Command and Programming Reference* to generate numeric values.
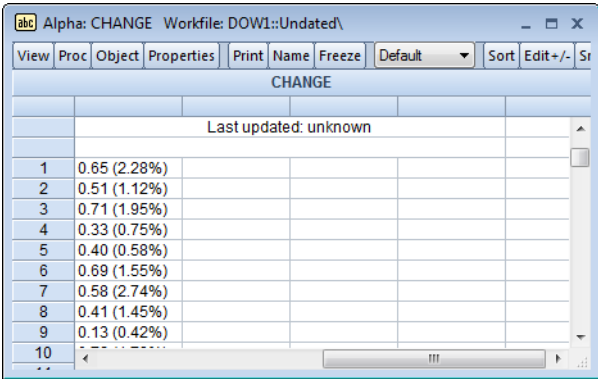
Two examples are of particular importance. First, you may have an alpha series that contains string representations of numbers, such as "3.14159". In order to use the strings as actual numbers, you must translate them into numbers, using either the string evaluation function `@val`.

Suppose, in our Dow Jones example, that we have the alpha series CHANGE containing information on the stock price change, expressed in both levels and percentages.

If we wish to extract only the levels information from the alpha series, the `@left` function may be used to extract the leftmost four characters of each string. The `@val` function may then be used to obtain the numeric value for each observation. Putting this together, the command:
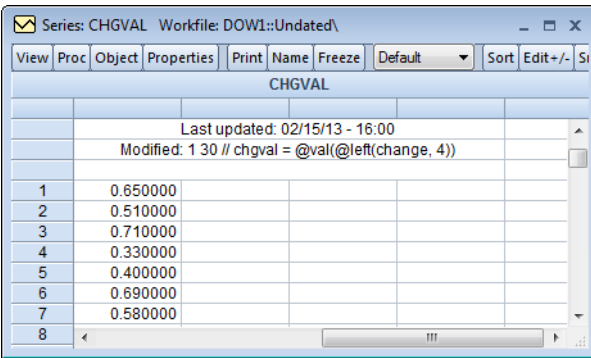
```
series chgval = @val(@left(change, 4))
```

converts the leading four characters of the CHANGE series into numeric values, and places the results in the series CHGVAL.

Second, you may have an alpha series that contains a text representation of dates. Here, we have a series DATES that contains text representations of dates in "dd-Mon-YY" format (one or two-digit day, dash, three-character month abbreviation, dash, two-digit year). For example, "12-Jun-03" represents June 12, 2003.

Alpha: DATES Workfile: DATES::Untitled\				
437	12-Jun-03			
438	13-Jun-03			
439	16-Jun-03			
440	17-Jun-03			
441	18-Jun-03			
442	19-Jun-03			
443	20-Jun-03			
444	23-Jun-03			
445	24-Jun-03			
446	25-Jun-03			
447	26-Jun-03			
448	27-Jun-03			
449	20 Jun 02			

To convert every element of this series into a numeric series containing date values, simply issue the command:

```
series dval = @dateval(dates)
```

The newly created series DVAL will contain date numbers associated with each of the string values in DATES.

Additional Issues

The Spreadsheet View

By default, the alpha series spreadsheet will display your data left-justified, with a column width of approximately 12 characters. You may change the justification and column width settings by clicking on the **Properties** button in the toolbar, then selecting a new justification setting and entering a new column width. Alternatively, the column width may be changed by dragging the separator in the column header to the desired position, or by double-clicking on the separator to adjust the column width to the minimum width that displays all of the observation values without truncation.

Auto-series

You should note that like ordinary series, you may also work directly with a series expression that produces an alpha series. For example, if ALPHA1 is an alpha series, the command:

```
show @lower(alpha1)
```

will result in an alpha series containing the contents of ALPHA1 with the text converted to all lowercase characters.

Auto-series expressions involving alpha series may also evaluate to ordinary series. For example, if NUMER1 is a numeric series and ALPHA1 is an alpha series, you may enter:

```
show numer1+@len(alpha1)+(alpha1>"cat")
```

to open a series window containing the results of the operation. Note that the parentheses around the relational comparison are required for correct parsing of the expression.

Date Series

A *date series* is a standard EViews numeric series that contains valid date values (see "Dates" on page 94 of the *Command and Programming Reference*). There is nothing that distinguishes a date series from any other numeric series, except for the fact that the values it contains may be interpreted as dates.

Creating a Date Series

There is nothing special about creating a date series. Any method of creating an EViews series may be used to create a numeric series that will be used as a date series.

Displaying a Date Series

The numeric values in a date series are generally of interest only when performing calendar operations. For most purposes, you will wish to see the values of your date series as date strings.

For example, the following series QDATES in our quarterly workfile is a numeric series containing valid date values for the start of each quarter. The numeric values of QDATES depicted here show the number of days since 1 January A.D. 1.

Obviously, this is not the way that most people will wish to view their date series. Accordingly, EViews provides considerable control over the display of the date values in your series. To change the display, click on the **Properties** button in the series toolbar, or select **View/Properties...** from the main menu.

EViews will display a dialog prompting you to change the display properties of the series. While you may change a variety of display settings such as the column width, justification, and indentation, here, for the moment, we are more interested in setting the properties of the **Numeric Display**.

For a date series, there are four settings of interest in the **Numeric Display** dropdown menu (**Period**, **Day**, **Day-Time**, and **Time**), with each corresponding to specific information that we wish to display in the series spreadsheet. For example, the **Day** selection allows you to display date information up to the day in various formats; with year, month, and day all included in the representation.

Let us consider our quarterly workfile example. Here we have selected **Period** and chosen a specific **Date format** entry ("YYYY[Q]Q"), which tells EViews that you wish to display the year, followed by the "Q" separator, and then the quarter number. Note also that when **Period** is selected, there is a **Current Workfile** setting in the **Date format** dropdown menu which tells EViews to use the current workfile display settings.

The two checkboxes below the **Date format** dropdown menu may be used to modify the selected date format. If you select

Two digit year, EViews will only display the last two-digits of the year (if the selected format is "YYYY[Q]Q", the actual format used will be "YY[Q]Q"); if you select **Day/month** order, days will precede months in whichever format is selected (if you select "mm/dd/YYYY" as the format for a day display, EViews will use "dd/mm/YYYY").

Applying this format to the QDATES series, the display changes to show the data in the new format:
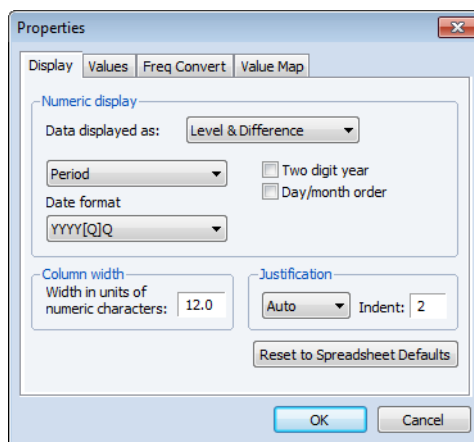
If instead we select the **Day** display, and choose the "YYYY-MM-DD" format, the QDATES spreadsheet will show:

There is one essential fact to remember about the QDATES series. Despite the fact that we have changed the display to show a text representation of the date, QDATES still contains the underlying numeric date values. This is in contrast to using an alpha series to hold a text representation of the date.

If you wish to convert a (numeric) date series into an alpha series, you will need to use the `@datestr` function. If you wish to convert an alpha series into a numeric date series, you will need to use the `@dateval` function. See "Translating between Date Strings and Date Numbers" on page 100 of the *Command and Programming Reference* for details.

Editing a Date Series

You may edit a date series either by either date numbers, or if the series is displayed using a date format, by entering date strings directly.

Suppose, for example, that we have our date series from above and that we wish to change a value. If we are displaying the series with date formatting, we may enter a date string, which EViews will automatically convert into a date number.

For example, we may edit our QDATES series by entering a valid date string ("April 10, 1992"), which EViews will convert into a date number (727297.0), and then display as a date string ("1992-04-10"). See "Free-format Conversion" on page 101 of the *Command and Programming Reference* for details on automatic translation of strings to date values.

Note, however, that if we were to enter the same string value in the series when the display is set to show numeric values, EViews will not attempt to interpret the string and will enter an NA in the series.

Value Maps

You may use the valmap object to create a *value map* (or *map*, for short) that assigns descriptive labels to values in numeric or alpha series. The practice of mapping observation values to labels allows you to hold your data in encoded or abbreviated form, while displaying the results using easy-to-interpret labels.

Perhaps the most common example of encoded data occurs when you have categorical identifiers that are given by integer values. For example, we may have a numeric series FEMALE containing a binary indicator for whether an individual is a female (1) or a male (0).

Numeric encodings of this type are commonly employed, since they allow one to work with the numeric values of FEMALE. One may, for example, compute the mean value of the FEMALE variable, which will provide the proportion of observations that are female.

On the other hand, numeric encoding of categorical data has the disadvantage that one must always translate from the numeric values to the underlying categorical data types. For example, a one-way tabulation of the FEMALE data produces the output:

```
Tabulation of FEMALE
Date: 09/30/03  Time: 11:36
Sample: 1 6
Included observations: 6
Number of categories: 2
```

Value	Count	Percent	Cumulative Count	Cumulative Percent
0	3	50.00	3	50.00
1	3	50.00	6	100.00
Total	6	100.00	6	100.00

Interpretation of this output requires that the viewer remember that the FEMALE series is encoded so that the "0" value represents "Male" and the "1" represents "Female". The example above would be easier to interpret if the first column showed the text representations of the categories in place of the numeric values.

Valmaps allow us to combine the benefits of descriptive text data values with the ease of working with a numeric encoding of the data. In cases where we define value maps for alphanumeric data, the associations allow us to use space saving abbreviations for the underlying data along with more descriptive labels to be used in presenting output.

Defining a Valmap

To create a valmap object, select **Object/New Object.../ValMap** from the main menu and optionally enter an object name, or enter the keyword "VALMAP" in the command line, followed by an optional name. EViews will open a valmap object.
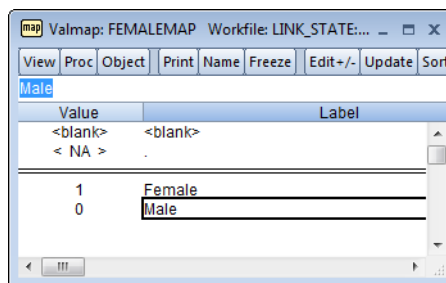
You will enter your new mappings below the double line by typing or by copy-and-pasting. In the **Value** column, you should enter the values for which you wish to provide labels; in the **Label** column, you will enter the corresponding text label. Here, we define a valmap named FEMALEMAP in which the value 0 is mapped to the string "Male", and the value 1 is mapped to the string "Female".

The two special entries above the double line should be used to define mappings for blank strings and numeric missing values. The default mapping is to represent blank strings as themselves, and to represent numeric missing values with the text "NA". You may change these defaults by entering the appropriate text in the **Label** column. For example, to change

the representation of missing numeric values to, say, a period, simply type the "." character in the appropriate cell.

We caution that when working with maps, EViews will look for exact equality between the value in a series and the value in the valmap. Such an equality comparison is subject to the usual issues associated with comparing floating point numbers. To mitigate these issues and to facilitate mapping large numbers of values, EViews allows you to define value maps using intervals.

To map an interval, simply enter a range of values in the **Value** column and the associated label in the **Label** column. You may use round and square parentheses, to denote open ("(", ")") or closed ("[", "]") interval endpoints, and the special values "–INF" and "INF" to represent minus and plus infinity.
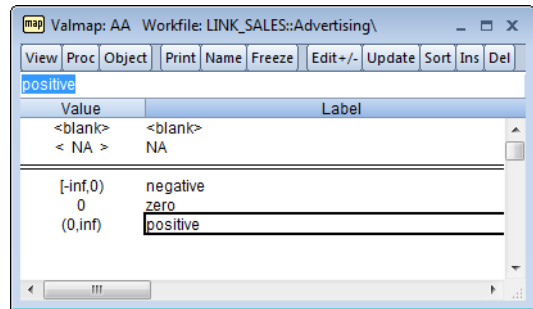
Using interval mapping, we require only three entries to map all of the negative values to the string "negative", the positive values to the string "positive", and the value 0 to the string "zero". Note that the first interval in our example, "[–inf, 0)", is mathematically incorrect since the lower bound should not be closed, but EViews allows the closed interval syntax in this case since there is no possibility of confusion.

While the primary use for valmaps will be to map numeric series values, there is nothing stopping you from defining labels corresponding to alpha series values (note that value and label matching is case sensitive). One important application of string value mapping is to expand abbreviations. For example, one might wish to map the U.S. Postal Service state abbreviations to full state names.
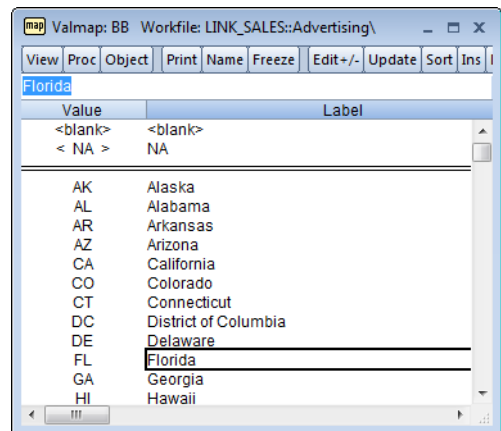
Since valmaps may be used with both numeric and alpha series, the text entries in the **Value** column may generally be used to match both numeric and alphanumeric values. For example, if you enter the text "0" as your value, EViews treats the entry as representing either a numeric 0 or the string value "0". Similarly, entering the text string "[0,1]" will match both numeric values in the interval, as well as the string value "[0,1]".

There is one exception to this dual interpretation. You may, in the process of defining a given valmap, provide an entry that conflicts with a previous entry. EViews will automatically identify the conflict, and will convert the latter entry into a string-only valmap entry.
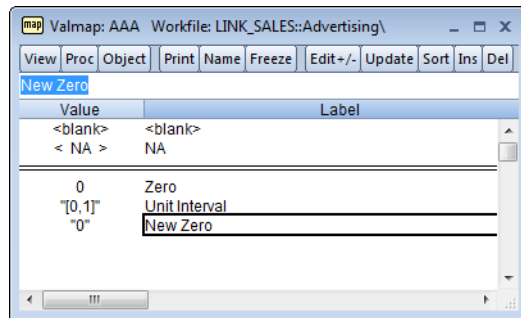
For example, if the first line of your valmap maps 0 to "Zero", a line that maps 0 to "New Zero", or one that maps "[0, 1]" to "Unit interval" conflicts with the existing entry. In the latter cases, the conflicting maps will be treated as text maps. Such a map is identified by enclosing the entry with quotation marks. Here, EViews has automatically added the enclosing quotation marks to indicate that the latter two label entries will only be interpreted as string maps, and not as numeric maps.

Once you have defined your mappings, click on the **Update** button on the toolbar to validate the object. EViews will examine your valmap and will remove entries with values that are exact duplicates. In this example, the last entry, which maps the string "0" to the value "New Zero" will be removed since it conflicts with the first line. The second entry will be retained since it is not an exact duplicate of any other entry. It will, however, be interpreted only as a string since the numeric interpretation would lead to multiple mappings for the value 0.

Assigning a Valmap to a Series

To use a valmap, you need to instruct EViews to display the values of the map in place of the underlying data. Before working with a valmap, you should be certain that you have updated and validated your valmap by pressing the **Update** button on the valmap toolbar.

First, you must assign the value map to your series by modifying the series properties. Open the series window and select **View/ Properties...** or click on the **Properties** button in the series toolbar to open the properties dialog. Click on the **Value Map** tab to display the value map name edit field.

If the edit field is blank, a value map has not been associated with this series. To assign a valmap, simply enter the name of a valmap object in the edit field and click on **OK**.

EViews will validate the entry and apply the specified map to the series. Note that to be valid, the valmap must exist in the same workfile page as the series to which it is assigned.
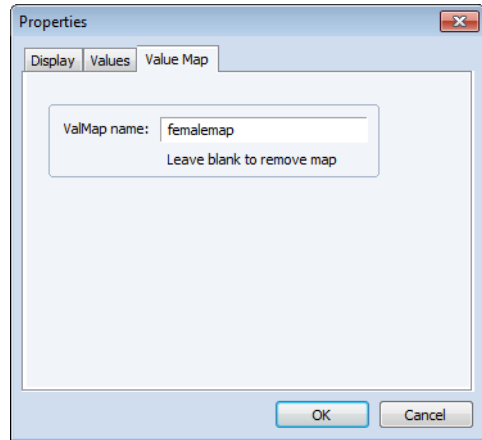
Using the Valmap Tools

EViews provides a small set of object-specific views and procedures that will aid you in working with valmaps.

Sorting Valmap Entries

You may add your valmap entries in any order without changing the behavior of the map. However, when viewing the contents of the map, you may find it useful to see the entries in sorted order.

To sort the contents of your map, click on **Proc/Sort...** from the main valmap menu. EViews provides you with the choice of sorting by the value column using numeric order (**Value - Numeric**), sorting by the value column using text order (**Value - Text**) or sorting by the label column (**Label**).
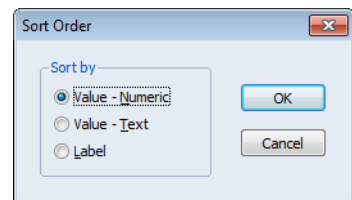
In the first two cases, we sort by the values in the first column of the valmap. The difference between the choices is apparent when you note that the ordering of the entries "9" and "10" depends upon whether we are interpreting the sort as a numeric sort, or as a text sort. Selecting **Value - Numeric** tells EViews that, where possible, you wish to interpret strings as numbers when performing comparisons (so that "9" is less than "10"); selecting **Value - Text** says that all values should be treated as text for purposes of comparison (so that "10" is less than "9").

Click on **OK** to accept the sort settings.

Examining Properties of a Valmap

You may examine a summary of your valmap by selecting **View/Statistics** in the valmap window. EViews will display a view showing the properties of the labels defined in the object.

The top portion of the view shows the number of mappings in the valmap, and the number of unique labels used in those definitions. Here we see that the valmap has four definitions, which map four values into four unique labels. Two of the four definitions are the special entries for blank strings and the numeric NA value.

The remaining portions of the view provide a detailed summary of the valmap describing the properties of the map when applied to numeric and to text values.
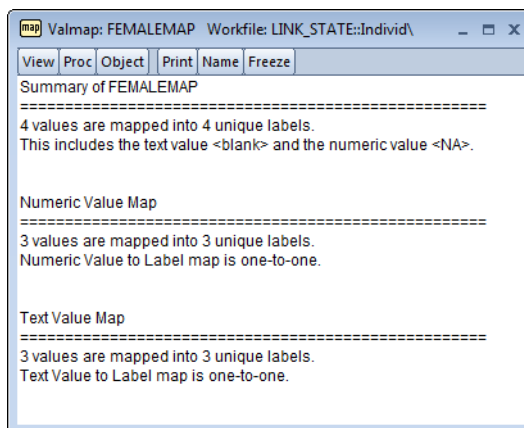
When applied to an ordinary numeric series, our FEMALEMAP example contains three relevant definitions that provide labels for the values 0, 1, and NA. Here, EViews reports that the numeric value mapping is one-to-one since there are no two values that produce the same value label.

The output also reports that the FEMALEMAP has three relevant definitions for mapping the three text values, "0", "1", and the blank string, into three unique labels. We see that the text interpreted maps are also one-to-one.

Note that in settings where we map an interval into a given label, or where a given text label is repeated for multiple values, EViews will report a many-to-one mapping. Knowing that a valmap is many-to-one is important since it implies that the values of the underlying source series are not uniquely identified by the label values. This lack of identification has important implications in editing mapped series and in interpreting the results from various statistical output (see "Editing a Mapped Series" on page 226 and "Valmap Definition Cautions" on page 230).

Tracking Valmap Usage

A single valmap may be applied to more than one series. You may track the usage of a given valmap by selecting **View/Usage** from the valmap main menu. EViews will examine every numeric and alpha series in the workfile page to determine which, if any, have applied the specified valmap.

The valmap view then changes to show the number and names of the series that employ the valmap, with separate lists for the numeric and the alpha series. Here we see that there is a single numeric series named FEMALE that uses FEMALEMAP.
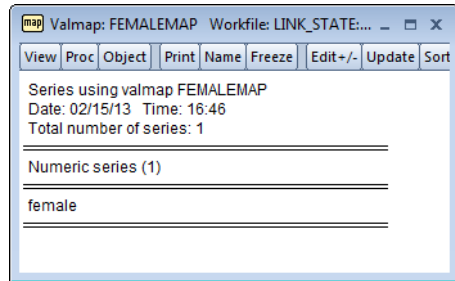
Working with a Mapped Series

Once you assign a map to a series, EViews allows you to display and edit your series using the mapped values and will use the labels when displaying the output from selected procedures.

Displaying series values

By default, once you apply a value map to a series, the EViews spreadsheet view will change to display the newly mapped values.
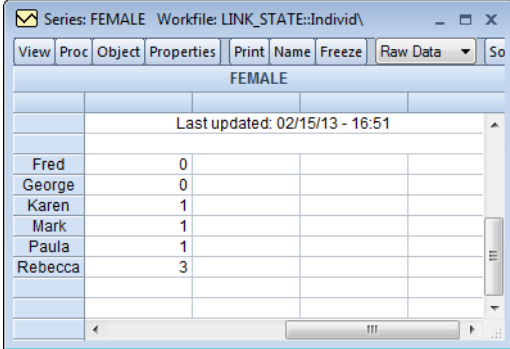
For example, after applying the FEMALEMAP to our FEMALE series, the series spreadsheet view changes to show the labels associated with each value instead of the underlying encoded values. Note that the display format drop-down menu usually visible in series toolbar indicates that EViews is displaying the **Default** series values, so that it shows the labels "Male" and "Female" rather than the underlying 0 and 1 values.

Note that if any of the values in the series does not have a corresponding valmap entry, EViews will display a mix of labeled and unlabeled values, with the unlabeled value "showing through" the mapping. For example, if the last observation in the FEMALE series had the value 3, the series spreadsheet will show observations with "Male" and "Female" corresponding to the mapped values, as well as the unmapped value 3.

There may be times when you wish to view the underlying series values instead of the labels. There are two possible approaches. First, you may remove the valmap assignment from the series. Simply go to the **Properties** dialog and delete the name of the valmap object from the **Value Map** page. The display will revert to showing the underlying values. Less drastically, you may use the display method dropdown menu to change the display format for the spreadsheet view. If you select **Raw Data**, the series spreadsheet view will change to show the underlying series data.

Editing a Mapped Series

To edit the values of your mapped series, first make certain you are in edit mode, then enter the desired values, either by typing in the edit field, or by pasting from the clipboard. How EViews interprets your input will differ depending upon the current display format for the series.

If your mapped series is displayed in its original form using the **Raw Data** setting, EViews will interpret any input as representing the underlying series values, and will place the input directly into the series. For example, if our FEMALE series is displayed using the **Raw Data** setting, any numeric input will be entered directly in the series, and any string input will be interpreted as an NA value.

In contrast, if the series is displayed using the **Default** setting, EViews will use the attached valmap both in displaying the labeled values and in interpreting any input. In this setting, EViews will first examine the attached valmap to determine whether the given input value is also a label in the valmap. If a matching entry is found, and the label matches a unique underlying value, EViews will put the value in the series. If there is no matching valmap label entry, or if there is an entry but the corresponding value is ambiguous, EViews will put the input value directly into the series. One implication of this behavior is that so long as the underlying values are not themselves valmap labels, you may enter data in either mapped or unmapped form. Note, again, that text value and label matching is case-sensitive.

Let us consider a simple example.

Suppose that the FEMALE series is set to display mapped values, and that you enter the value "Female". EViews will examine the assigned valmap, determine that "Female" corresponds to the underlying value "1", and will assign this value to the series. Since "1" is a valid form of numeric input, the numeric value 1 will be placed in the series. Note that even though we have implicitly entered 1 into the series, the mapped spreadsheet view will continue to show the value "Female".

Alternatively, we could have directly entered the "1" corresponding to the underlying numeric value. Since "1" is not a valmap label, EViews will put the value 1 in the series, which will be displayed using the label "Female".

While quite useful, entering data in mapped display mode requires some care, as your results may be somewhat unexpected. For one, you should bear in mind that the required reverse lookup of values associated with a given input requires an exact match of the input to a label value, and a one-to-one correspondence between the given label and a valmap value. If this condition is not met, the original input value will be placed in the series.

Consider, for example, the result of entering the string "female" instead of "Female". In this case, there is no matching valmap label entry, so EViews will put the input value, "female", into the series. Since FEMALE is a numeric series, the resulting value will be an NA, and the display will show the mapped value for numeric missing values.

Similarly, suppose you enter "3" into the last observation of the FEMALE series. Again, EViews does not find a corresponding valmap label entry, so the input is entered directly into the series. In this case, the input represents a valid number so that the resulting value will be a 3. Since there is no valmap entry for this value, the underlying value will be displayed.

Lastly, note that if the matching valmap label corresponds to multiple underlying values, EViews will be unable to perform the reverse lookup. If, for example, we modify our valmap so that the interval "[1, 10]" (instead of just the value 1) maps to the label "Female", then when you enter "Female" as your input, it is impossible to determine a unique value for the series. In this case, EViews will enter the original input, "Female", directly into the series, resulting in an NA value.

See "Valmap Definition Cautions" on page 230 for additional cautionary notes.

Using a Mapped Series

You may use a mapped series as though it were any series. We emphasize the fact that the mapped values of a series are not replacements of the underlying data; they are only labels to be used in output. Thus, when performing numeric calculations with a series, EViews will always use the underlying values of the series, not the label values. For example, if you map the numeric value -99 to the text "NA", and take the absolute value of the mapped numeric series containing that value, you will get the value 99, and not a missing value.

In appropriate settings (where the series values are treated as categories), EViews routines will use the labels when displaying output. For example, a one-way frequency tabulation of the FEMALE series with the assigned FEMALEMAP yields:

```
Tabulation of FEMALE
Date: 09/30/03   Time: 11:36
Sample: 1 6
Included observations: 6
Number of categories: 2
```

Value	Count	Percent	Cumulative Count	Cumulative Percent
Female	3	50.00	3	50.00
Male	3	50.00	6	100.00
Total	6	100.00	6	100.00

Similarly, when computing descriptive statistics for the SALES data categorized by the values of the FEMALE series, we have:

```
Descriptive Statistics for SALES
Categorized by values of FEMALE
Date: 10/01/03   Time: 19:30
Sample: 1 6
Included observations: 6
```

FEMALE	Mean	Std. Dev.	Obs.
Female	323.3333	166.2328	3
Male	263.3333	169.2139	3
All	293.3333	153.5795	6

Valmap Functions

To facilitate working with valmaps, three new `genr` functions are provided which allow you to translate between unmapped and mapped values. These functions may be used as part of standard series or alpha expressions.

First, to obtain the mapped values corresponding to a set of numbers or strings, you may use the command:

> `@map`(*arg[, map_name]*)

where *arg* is a numeric or string series expression or literal, and the optional *map_name* is the name of a valmap object. If *map_name* is not provided, EViews will attempt to determine the map by inspecting *arg*. This attempt will succeed only if *arg* is a numeric series or alpha series that has previously been mapped.

Let us consider our original example where the FEMALEMAP maps 0 to "Male" and 1 to "Female". Suppose that we have two series that contain the values 0 and 1. The first series, MAPPEDSER, has previously applied the FEMALEMAP, while the latter series, UNMAPPED-SER, has not.

Then the commands:

```
alpha s1 = @map(mappedser)
alpha s2 = @map(mappedser, femalemap)
```

are equivalent. Both return the labels associated with the numeric values in the series. The first command uses the assigned valmap to determine the mapped values, while the second uses FEMALEMAP explicitly.

Alternately, the command:

```
alpha s3 = @map(unmappedser)
```

will generate an error since there is no valmap assigned to the series. To use `@map` in this context, you must provide the name of a valmap, as in:

```
alpha s4 = @map(unmappedser, femalemap)
```

which will return the mapped values of UNMAPPEDSER, using the valmap FEMALEMAP.

Conversely, you may obtain the numeric values associated with a set of string value labels using the `@unmap` function. The `@unmap` function takes the general form:

`@unmap(`*arg, map_name*`)`

to return the numeric values that have been mapped into the string given in the string expression or literal *arg*, where *map_name* is the name of a valmap object. Note that if a given label is associated with multiple numeric values, the missing value NA will be returned. Note that the *map_name argument is required with the* `@unmap` *function.*

Suppose, for example, that you have an alpha series STATEAB that contains state abbreviations ("AK", "AL", etc.) and a valmap STATEMAP that maps numbers to the abbreviations. Then:

```
series statecode = @unmap(stateab, statemap)
```

will contain the numeric values associated with each value of STATEAB.

Similarly, you may obtain the string values associated with a set of string value labels using:

`@unmaptxt(`*arg, map_name*`)`

where *arg* is a string expression or literal, and *map_name* is the name of a valmap object. `@unmaptxt` will return the underlying string values that are mapped into the string labels provided in *arg*. If a given label is associated with multiple values, the missing blank string "" will be returned.

Valmap Definition Cautions

EViews allows you to define quite general value maps that may be used with both numeric and alpha series. While potentially useful, the generality comes with a cost, since if used carelessly, valmaps can cause confusion. Accordingly, we caution you that there are many features of valmaps that should be used with care. To illustrate the issues, we list a few of the more problematic cases.

Many-to-one Valmaps

A many-to-one valmap is a useful tool for creating labels that divide series values into broad categories. For example, you may assign the label "High" to a range of values, and the label "Low" to a different range of values so that you may, when displaying the series labels, easily view the classification of an observation.

The downside to many-to-one valmaps is that they make interpreting some types of output considerably more difficult. Suppose, for example, that we construct a valmap in which several values are mapped to the label "Female". If we then display a one-way frequency table for a series that uses the valmap, the label "Female" may appear as multiple entries. Such a table is almost impossible to interpret since there is no way to distinguish between the various "Female" values.

A series with an attached many-to-one valmap is also more difficult to edit when viewing labels since EViews may be unable to identify a unique value corresponding to a given label. In these cases, EViews will assign a missing value to the series, which may lead to confusion (see "Editing a Mapped Series" on page 226).

Mapping Label Values

Defining a map in which one of the label values is itself a value that is mapped to a label can cause confusion. Suppose, for example, that we have a valmap with two entries: the first maps the value 6 to the label "six", and the second maps the value "six" to the label "high".

Now consider editing an alpha series that has this valmap attached. If we use the **Default** display, EViews will show the labeled values. Thus, the underlying value "six" will display as the value "high"; while the value "6" will display as "six". Since the string "six" is used both as a label and as a value, in this setting we have the odd result that it must be entered indirectly. Thus, to enter the string "six" in the alpha series, we have the counterintuitive result that you must type "high" instead of "six", since entering the latter value will put "6" in the series.

Note, however, that if you display the series in **Raw Data** form, all data entry is direct; entering "six" will put the value "six" into the series and entering "high" will put the value "high" in the series.

Mapping Values to Numbers

Along the same lines, we strongly recommend that you not define value maps in which numeric values can be mapped to labels that appear to be numeric values. Electing, for example, to define a valmap where the value 5 is mapped to the label "6" and the value 6 is mapped to the label "5", is bound to lead to confusion.

Chapter 8. Series Links

The *series link* object (or *link*, for short) provides you with powerful tools for combining information from different workfile pages. Links provide an easy-to-use interface to a wide range of sophisticated data operations such as:

- merging data from one workfile page into another
- saving "by-group" summary statistics into a workfile page
- matching observations between dated workfile pages
- performing frequency conversion between regular dated workfile pages

Links operate both dynamically and on demand, so that the desired operation is performed only when needed, and is updated automatically whenever your data change.

You may find that working with links is in many ways similar to working with data tables in a relational database management system. Indeed, links have specifically been designed to provide much of the power of these sophisticated systems. But you need not have worked with such a system to take advantage of the power, ease-of-use, and flexibility associated with link objects.

Note that you may also use auto-series to work with data on other workfile pages. This method is easier than creating a link object, but more limiting ("Auto-series Across Pages," on page 196).

We begin with a discussion of basic link concepts that outlines the basic operations supported by links. In later sections we document the use of links in EViews.

Basic Link Concepts

A link is a series-like object that exists in one workfile page, but "refers" to series data in another workfile page. At a basic level, a link is a description of how EViews should use data in a *source* workfile page to determine values of a series in the current, or *destination*, workfile page.

A link contains three fundamental components:

- First, there is the name of a *source series*. The source series identifies the series in the source workfile page that is used as a basis for obtaining values in the destination page.
- Second, the link contains the names of one or more link *identifier (ID)* series in both the source and destination pages. The source ID and destination ID series will be used to match observations from the two pages.

- Lastly, the link contains a description of how the source series should be used to construct link values for matching observations in the destination page.

The basic series link employs a method called *match merging* to determine the link values in the destination page. More advanced links combine match merging with automatic frequency conversion. We describe these two methods in detail below, in "Linking by general match merging" on page 234 and "Linking by date with frequency conversion" on page 243.

As the name suggests, the series link object shares most of the properties of a series. You may, in fact, generally use a series link as though it were a series. You may examine series views, perform series procedures, or use the series link to generate new data, or you may use the link as a regressor in an equation specification.

Another important property of links is that they are "live", in the sense that the values in the link change as its underlying data change. Thus, if you have a link in a given workfile page, the link values will automatically be updated when the source series or ID series values change.

Lastly, links are memory efficient. Since links are computed and updated as needed, the values of the series link are not held in memory unless they are in use. Thus, it is possible to create a page populated entirely by links that takes up only the minimum amount of memory required to perform all necessary operations.

Linking by general match merging

We begin our discussion of linking with a brief, and admittedly terse, description of how a basic link with match merging works. More useful, perhaps, will be the extended examples that follow.

The basic link first compares values for one or more source ID series with the values in the destination ID series. Observations in the two pages are said to match if they have identical ID values. When matches are observed, values from the source series are used to construct values of the link for the corresponding observations in the destination page.

Each link contains a description of how the source series should be used to construct link values in the destination page. Constructing values for a basic match merge link involves two steps:

- First, we perform a *contraction* of the source series to ensure that there is a single value associated with each distinct source ID value. The contraction method employed describes how the (possibly) multiple source series observations sharing a given ID value should be translated into a single value.

- Next, we take the distinct source IDs and contracted source series values, and perform a match merge in which each contracted value is repeated for all matching observations in the destination page.

This basic method is designed to handle the most general cases involving many-to-many match merging by first computing a many-to-one contraction (by-group summary) of the source series, and then performing a one-to-many match merge of the contracted data.

All other match merges are handled as special cases of this general method. For a many-to-one match merge, we first compute the contraction, then perform one-to-one matching of the contracted data into the destination page. In the more common one-to-many or one-to-one match merge, the contraction step typically has no practical effect since the standard contractions simply return the original source series values. The original values are then linked into the destination page using a simple one-to-one or one-to-many match merge.

While all of this may seem a bit abstract, a few simple examples should help to fix ideas. Suppose first that we have a state workfile page containing four observations on the series STATE1 and TAXRATE:

State1	TaxRate
Arkansas	.030
California	.050
Texas	.035
Wyoming	.012

In the same workfile, we have a second workfile page containing individual level data, with a name, NAME, state of residence, STATE2, and SALES volume for six individuals:

Name	State2	Sales
George	Arkansas	300
Fred	California	500
Karen	Arkansas	220
Mark	Texas	170
Paula	Texas	120
Rebecca	California	450

We wish to link the data between the two pages. Note that in this example, we have given the state series different names in the two pages to distinguish between the two. In practice there is no reason for the names to differ, and in most cases, the names will be the same.

One-to-many match merge

Our first task will be to create, in the page containing individual information, a series containing values of the TAXRATE faced by every individual. We will determine the individual rates by examining each individual's state of residence and locating the corresponding tax

rate. George, for example, who lives in Arkansas, will face that state's tax rate of 0.030. Similarly, Mark, who lives in Texas, has a tax rate of 0.035.

We will use a series link to perform a one-to-many match merge in which we assign the TAXRATE values in our source page to multiple individuals in our destination page.

For the three basic components of this link, we define:

- the source series TAXRATE
- the source identifier STATE1 and destination identifier STATE2
- the merge rule that the values of TAXRATE will be repeated for every individual with a matching STATE2 value in the destination page

This latter merge rule is always used for basic links involving one-to-many match merges. Here, the rule leads to the natural result that each individual is assigned the TAXRATE value associated with his or her state.

After performing the link, the individual page will contain the merged values for the tax rate in TAXRATE2. We use the "2" in the TAXRATE2 name to denote the fact that these data are generated by merging data using STATE2 as the destination ID series:

Name	State2	Sales	TaxRate2
George	Arkansas	300	.030
Fred	California	500	.050
Karen	Arkansas	220	.030
Mark	Texas	170	.035
Paula	Texas	120	.035
Rebecca	California	450	.050

We mention one other issue in passing that will become relevant in later discussion. Recall that all basic links with match merging first contract the source series prior to performing the match merge. In this case, the specified merge rule implicitly defines a contraction of the source series TAXRATE that has no effect since it returns the original values of TAXRATE. It is possible, though generally not desirable, to define a contraction rule which will yield alternate source values in a one-to-many match merge. See "Link calculation settings" on page 249.

Many-to-one match merge

Alternatively, we may wish to link data in the opposite direction. We may, for example, choose to link the SALES data from the individual page to the destination state page, again matching observations using the two state IDs. This operation is a many-to-one match

merge, since there are many observations with STATE2 ID values in the individual page for each of the unique values of STATE1 in the state page.

The components of this new link are easily defined:

- the source series SALES

- the source identifier STATE2 and destination identifier STATE1

- a merge rule stating that the values of SALES will first be contracted, and that the contracted values will be placed in matching observations in the destination page

Specifying the last component, the merge rule, is a bit more involved here since there are an unlimited number of ways that we may contract the individual data. EViews provides an extensive menu of contraction methods. Obvious choices include computing the mean, variance, sum, minimum, maximum, or number of observations for each source ID value. It is worth noting here that only a subset of the contraction methods are available if the source is an alpha series.

To continue with our example, suppose that we choose to take the sum of observations as our contraction method. Then contraction involves computing the sum of the individual observations in each state; the summary value for SALES in Arkansas is 520, the value in California is 950, and the value in Texas is 290. Wyoming is not represented in the individual data, so the corresponding contracted value is NA.

Given this link definition, the many-to-one match merge will result in a state page containing the match merged summed values for SALES1:

State1	TaxRate	Sales1	Sales1ct
Arkansas	.030	520	2
California	.050	950	2
Texas	.035	290	2
Wyoming	.012	NA	0

Similarly, we may define a second link to the SALES data containing an alternative contraction method, say the count of non-missing observations in each state. The resulting link, SALES1CT, shows that there are two individual observations for each of the first three states, and none for Wyoming.

Many-to-many match merge

Lastly, suppose that we have a third workfile page containing a panel structure with state data observed over a two year period:

Year	State3	TaxRate
1990	Arkansas	.030
1991	Arkansas	.032
1990	California	.050
1991	California	.055
1990	Texas	.035
1991	Texas	.040
1990	Wyoming	.012
1991	Wyoming	.035

Linking the SALES data from the *individual* page to the panel page using the STATE2 and STATE3 identifiers involves a many-to-many match merge since there are multiple observations for each state in both pages.

The components of this new link are easily defined:

- the source series SALES

- the source identifier STATE2 and destination identifier STATE3

- a merge rule stating that the values of SALES will first be contracted, and that the contracted values will be repeated for every observation with a matching STATE3 value in the destination page

This merge rule states that we perform a many-to-many merge by first contracting the source series, and then performing a one-to-many match merge of the contracted results into the destination. For example, linking the SALES data from the individual page into the panel state-year page using the sum and count contraction methods yields the link series SALES3 and SALES3A:

Year	State3	TaxRate	Sales3	Sales3a
1990	Arkansas	.030	520	2
1991	Arkansas	.032	520	2
1990	California	.050	950	2
1991	California	.055	950	2
1990	Texas	.035	290	2
1991	Texas	.040	290	2
1990	Wyoming	.012	NA	0
1991	Wyoming	.035	NA	0

It is worth noting that this many-to-many match merge is equivalent to first performing a many-to-one link from the individual page into the state page, and then constructing a one-to-many link of those linked values into the panel page. This two-step method may be achieved by first performing the many-to-one link into the state page, and then performing a one-to-many link of the SALES1 and SALES1CT links into the panel page.

Linking by date match merging

To this point, we have primarily considered simple examples involving a single categorical link identifier series (states). You may, of course, construct more elaborate IDs using more than one series. For example, if you have data on multinational firms observed over time, both the firm and date identifiers may be used as the link ID series.

The latter example is of note since it points to the fact that dates may be used as valid link identifiers. The use of dates as identifiers requires special discussion, as the notion of a match may be extended to take account of the calendar.

We begin our discussion of merging using dates by noting that a date may be employed as an identifier in two distinct ways:

- First, an ID series containing date values or alphanumeric representations of dates may be treated like any other ID series. In this case, the value in one workfile page must be *identical* to the value in the other page for a match to exist.

- Alternatively, when we are working with regular frequency data, we may take advantage of our knowledge of the frequency and the calender to define a broader notion of date matching. This broader form of matching, which we term *date matching*, involves comparing dates by first rounding the date ID values down to the lowest common regular frequency and then comparing the rounded values. Note that date matching requires the presence of at least one regular frequency for the rounding procedure to be well-defined.

In practical terms, date matching produces the outcomes that one would naturally expect. With date matching, for example, the quarterly observation "2002Q1" matches "2002" in a regular annual workfile, since we round the quarterly observation down to the annual frequency, and then match the rounded values. Likewise, we would match the date "March 3, 2001" to the year 2001 in an annual workfile, and to "2001Q1" in a quarterly workfile. Similarly, the date "July 10, 2001" also matches 2001 in the annual workfile, but matches "2001Q3" in the quarterly workfile.

Basic links with date matching

Consider the following simple example of linking using date matching. Suppose that we have a workfile containing two pages. The first page is a regular frequency quarterly page containing profit data (PROFIT) for 2002 and 2003:

Quarter	Profit
2002Q1	120
2002Q2	130
2002Q3	150
2002Q4	105
2003Q1	100
2003Q2	125
2003Q3	200
2003Q4	170

while the second page contains irregular data on special advertising events (ADVERT):

Date	Advert
Jan 7, 2002	10
Mar 10, 2002	50
Apr 9, 2002	40
May 12, 2002	90
Mar 1, 2003	70
Dec 7, 2003	30
Dec 23, 2003	20

We would like to link the quarterly profit data to the irregular data in the advertising page. The quarterly values in the source page are unique so that we have a one-to-one or one-to-many match merge; accordingly, we may select any contraction method that leaves the original PROFIT data unchanged (mean, unique, *etc.*).

We first employ date matching by using the "@DATE" and "@DATE" keywords as our ID series. This specification instructs EViews to use the knowledge about the date structures in the page to perform a sophisticated matching across pages. Using this approach, we construct a PROFIT1 link containing the values:

Date	Advert	Profit1
Jan 7, 2002	10	120
Mar 10, 2002	50	120
Apr 9, 2002	40	130
May 12, 2002	90	130
Mar 1, 2003	70	100

| Dec 7, 2003 | 30 | 170 |
| Dec 23, 2003 | 20 | 170 |

In evaluating the values in PROFIT1, we simply repeat the value of PROFIT for a given quarter for every matching observation in the advertising page. Since we are using date matching, we employ matching across pages that uses calendar knowledge to determine matches. For example, the observation for quarter "2002Q1" matches both "Jan 7, 2002" and "Mar 10, 2002" in the advertising page so that the latter observations are assigned the value of 120.

Conversely, using date matching to link the ADVERT series to the quarterly page, we have a many-to-one match merge since, after rounding down to the lower frequency, multiple observations in the advertising page have "@DATE" values that match the unique "@DATE" values in the quarterly page. If we choose to employ the mean contraction method in the link ADVERT1, we have:

Quarter	Profit	Advert1
2002Q1	120	30
2002Q2	130	65
2002Q3	150	NA
2002Q4	105	NA
2003Q1	100	70
2003Q2	125	NA
2003Q3	200	NA
2003Q4	170	25

Here, the values of ADVERT1 contain the mean values over the observed days in the quarter. For example, the value for ADVERT1 in 2002Q1 is taken by averaging the values of ADVERT for "Jan 7, 2002" and "Mar 10, 2002". Note that the value for quarter 2002Q3 is NA since there are no observations with matching DATE values, *i.e.*, there are no observations in the advertising page that fall within the quarter.

It is worth noting that in both of these examples, had we employed exact matching using by specifying our ID series as QUARTER and DATE, we would have observed no matches since the date numbers for the quarterly data do not match any of the irregular date numbers. As a result, all of the values in the resulting link would be assigned the value NA.

Panel links with date matching

When using date matching to link dated panel data to a page with a different frequency, you should pay particular attention to the behavior of the merge operation since the results may differ from expectations.

An example will illustrate the issue. Consider the following simple panel featuring quarterly revenue data from 2002Q1 to 2003Q4:

Firm	Quarter	Revenue
1	2002Q1	120
1	2002Q2	130
1	2002Q3	150
1	2002Q4	105
1	2003Q1	100
1	2003Q2	125
1	2003Q3	200
1	2003Q4	170
2	2002Q1	40
2	2002Q2	40
2	2002Q3	50
2	2002Q4	35
2	2003Q1	20
2	2003Q2	25
2	2003Q3	50
2	2003Q4	40

We will consider the results from linking the REVENUE data into an annual page using date matching of the QUARTER and the YEAR identifiers. Using date match merging (with the "@DATE" and "@DATE" keywords), and employing both the sum and number of observations contractions, we observe the results in REVENUE1 (sum) and REVENUE1A (obs):

Year	Revenue1	Revenue1a
2002	670	8
2003	730	8

The important thing to note here is that the sums for each year have been computed over all eight matching observations in the panel page.

The key to understanding the result is to bear in mind that date matching only changes the way that a match between observations in the two pages is defined; the remaining match merge operation remains unchanged. The outcome is simply the result of applying standard link behavior in which we first identify matches, compute a contraction over all matching observations, and perform the one-to-one match merge.

An alternative approach to obtaining annual revenue values from the panel data would be to first contract the panel data to a quarterly frequency by averaging across firms, and then to convert the quarterly data to an annual frequency by summing over quarters. This approach, which produces very different results from the first method, may be undertaken in two steps: by first linking the quarterly panel data into a quarterly page (using the mean contraction), and then frequency converting by linking the quarterly data to the annual frequency (summing over quarters).

In cases where you are linking between two panel pages at different frequencies there are yet more complication. See "Panel frequency conversion" on page 245 for a description of the issues involved in constructing these types of links.

Linking by date with frequency conversion

In the special case where we wish to link data between two regular frequency pages using dates as the sole identifier, EViews allows you to define your links in two ways. First, you may use the date match merging described in "Linking by date match merging" on page 239, or you can define special links that employ frequency conversion.

Basic frequency conversion

Links specified by date will primarily be used to perform automatic frequency conversion of simple regular frequency data. For example, you may choose to hold your quarterly frequency data in one page, your monthly frequency data in a second page, and to create links between pages which automatically perform the up or down frequency conversion as necessary.

You can instruct EViews to use the source series default methods for converting between frequencies, or you may use the link definition to specify the up and down conversion methods. Furthermore, the live nature of links means that changes in the source data will generate automatic updates of the frequency converted link values.

We divide our discussion of frequency conversion links into those that link data from high to low frequency pages and those that link from low to high frequency pages.

High to low frequency conversion

Frequency conversion linking from a simple regular high frequency page to a regular low frequency page is fundamentally the same as using a link with date matching to perform basic many-to-one match merging. In both cases, we match dates, compute a contraction of the source series, and then perform a one-to-one match merge.

Given the specialized nature of frequency conversion, links specified by date with frequency conversion offer a subset of the ordinary link contraction methods. All of the standard high to low frequency conversion methods (average, sum, first, last, maximum and minimum)

are supported, but the match merge methods which do not preserve levels, (such as the sum-of-squares or the variance) are not included.

Frequency conversion links also allow you to disable conversions for partially observed periods, so that a missing value for the source series in a given month generates a missing value for the corresponding quarterly observation. This option is not available for basic match merge links.

Low to high- frequency conversion

In contrast, linking from low to high frequency pages using frequency conversion differs substantively from linking using basic date match merging.

When linking using general date match merging, the frequency conversion implied by the one-to-many match merge may only be performed by repeating the low frequency observation for every matching high frequency observation. Thus, in a one-to-many date match merge, an annual observation is always repeated for each matching quarter, month, or day.

In contrast, EViews provides additional up-conversion methods for frequency conversion links. In addition to the simple repeated-observation (constant-match average) method, frequency conversion links support all of the standard frequency conversion methods including constant-match sum, quadratic-match sum, quadratic-match average, linear-match sum, linear-match last, and cubic-match last.

Suppose that, in addition to our regular frequency quarterly PROFIT workfile page , we have a regular frequency monthly page containing observations spanning the period from August 2002 to March 2003. Linking the PROFIT data from the quarterly page into the monthly page by date, with frequency conversion, requires that we specify an up-conversion method. Here, we show results of a frequency conversion link using both the simple constant-match average (PROFIT2) and quadratic-match average (PROFIT3) methods:

Month	Profit2	Profit3
Aug 2002	150	152.407
Sep 2002	150	144.630
Oct 2002	105	114.074
Nov 2002	105	103.519
Dec 2002	105	97.407
Jan 2003	100	97.222
Feb 2003	100	98.889
Mar 2003	100	103.889

Note that the PROFIT2 values are the same as those obtained by linking using simple date match merging, since the constant-match average method simply repeats the PROFIT obser-

vations for each matching month. Conversely, the PROFIT3 values are obtained using an interpolation method that is only available for linking by date with frequency conversion.

Panel frequency conversion

There are additional issues to consider when performing frequency conversion links in panel workfile settings:

- When working with two regular frequency panel pages, each defined by a single ID, frequency conversion links construct values by performing the frequency conversion separately for each value of the panel ID.

- If the source page is a regular frequency panel and the destination is an ordinary regular frequency page, we contract the source series by computing *means* across the panel identifiers to form a single time series (note that "mean" is the only contraction allowed). The resulting time series, which is in the source frequency, is then frequency converted to the destination frequency. Finally, the resulting series is date match merged to the destination page.

  Note that this is the behavior is the same as performing a general match merge using "@DATE" as the identifiers, with a mean contraction.

- If the source page is an ordinary regular frequency page, and the destination is a regular frequency panel, we frequency convert to the destination frequency, then date match merge to the corresponding dates in each ID in the destination. Thus given source value is *repeated* for all matching dates in the destination page (*i.e.*, all IDs will have the same time-series).

In all three of these cases, all of the high-to-low conversion methods are supported, but low-to-high frequency conversion only offers **Constant-match average** (repeating of the low frequency observations).

Lastly, frequency conversion involving a panel page with more than one dimension or an undated page will be performed using raw data copy unless you elect to employ general match merging, as described in "Panel links with date matching" on page 241.

An example will illustrate the general approach. Suppose again that we are working with the regular frequency, quarterly panel REVENUE data. For convenience, we repeat the data here:

Firm	Quarter	Revenue
1	2002Q1	120
1	2002Q2	130
1	2002Q3	150
1	2002Q4	105
1	2003Q1	100

1	2003Q2	125
1	2003Q3	200
1	2003Q4	170
2	2002Q1	40
2	2002Q2	40
2	2002Q3	50
2	2002Q4	35
2	2003Q1	20
2	2003Q2	25
2	2003Q3	50
2	2003Q4	40

We now wish to use frequency conversion to link these data into an annual panel, using the average frequency conversion method to go from high-to-low frequency. Then the panel-to-panel frequency conversion will simply perform a frequency conversion for each firm.

Firm	**Year**	**Revenue2**
1	2002	505
1	2003	595
2	2002	165
2	2003	135

Here we have performed an ID specific sum as our high-to-low conversion method. The observation for FIRM 1 in YEAR 2002 in the annual page is simply the sum of the quarterly data for that firm in 2002. Similarly, the value for FIRM 2 in 2003 is the sum of the four quarterly values for FIRM 2 in 2003.

It is important to note that these results differ significantly from the results obtained by general match merging using date processing of matches (with "@DATE" and "@DATE" as the two identifiers). Using the latter approach, we would have obtained:

Firm	**Year**	**Revenue3**
1	2002	670
1	2003	730
2	2002	670
2	2003	730

While these results may at first seem a bit odd, they simply follow the logic of the discussion in "Panel links with date matching" on page 241. Note that our link which matches dates

between the two panel workfile pages is an example of a many-to-many match merge, since there are multiple IDs with the same dates in each page. Thus, we will first contract across IDs to obtain a unique time series in the original frequency, then frequency convert, then one-to-many match in the destination page. In this case, the initial contraction involves summing over firms to obtain a quarterly time series, then frequency converting (summing) to the destination frequency to obtain annual values for 2002 (670) and 2003 (730). The final step match merges these converted values into the annual panel using a one-to-many match merge rule.

Creating a Link

Links may be created interactively either by copying-and-pasting or drag-and-dropping a series from the source to the destination page, or by issuing a link declaration in the destination page.

Creating a link using copy-and-paste or drag-and-drop

To define a link using copy-and-paste, first select one or more source series in the source workfile page, and either click on the right mouse button and select **Copy**, or select **Edit/Copy** from the main EViews menu. Next, switch to the destination page by clicking on the appropriate tab, and either click on the right mouse button and select **Paste Special...**, or select **Edit/Paste Special...** from the main menu.

Alternately, you may right-click to select the source series, then drag-and-drop to the destination page. EViews will display the **Paste Special** dialog.

General match merge links

Here we have used **Paste Special...** to copy-and-paste the series TAXRATE from the source page into a destination page. If neither the source nor the destination series are dated pages, EViews will display a dialog prompting you to merge by date or by match merge. Select **General match merge criteria** in the bottom left to specify match merge options.

Destination name

The field in the upper left-hand
portion of the dialog should be
used for specifying the name of the
destination object. Here, we have
the default wildcard value of "*"
indicating that the series named
TAXRATE in the source page will
be used in the destination page. We
may modify the name by typing an
explicit name such as "NEWTAX",
or by entering an expression con-
taining the wildcard character. For
example, if we wish to use the name "NEWTAXRATE" in the destination page, we may enter
"NEW*" in the edit field.
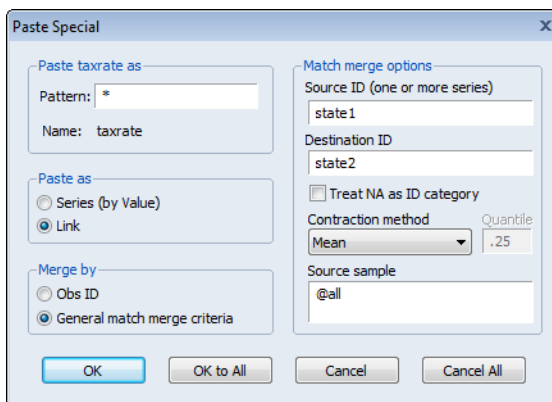
The wildcard processing is particularly useful if you are copying multiple series into a new
page since it facilitates batch renaming of series.

Destination type

Next, you will choose between pasting the series by value, or pasting the series as a link. If
you paste by value, EViews will create an ordinary series in the destination page, and will
fill it with the values from the link evaluation. If you paste your series as a link, EViews will
create an actual link object containing the desired specification. As you might expect, there
are significant differences between the two methods of copying your series.

In the first method, the link computations are performed immediately and the destination
series values are assigned at the time the series is created. This behavior follows the tradi-
tional model of match merging and frequency conversion in which the operation is per-
formed once to compute static values.

When you paste your series as a link, EViews defines a link object containing a specification
of the match merge or frequency conversion. At creation, the link object is not evaluated
and uses no memory. Then, whenever you access the values in the link series, EViews will
determine whether the object needs evaluation and if so, will allocate memory and perform
the link calculations.

With links, you gain the benefits of efficient memory use and dynamic updating of the val-
ues in the destination, at the cost of some speed since the link calculations may be per-
formed more than once. Along these lines, it is worth pointing out that links may be
converted into ordinary series at any time. Once a series is created, however, it may not be
converted back into a link.

Match merge options

Whether you elect to create a new series with fixed values or to create a new link series, you must specify link options.

Match ID information

First, you must specify the information that EViews will use to identify matches between observations in the two pages.

In the **Source ID** and **Destination ID** edit fields, you will enter the names of one or more source ID series and one or more destination ID series. The number and order of the names in the two fields should match. Thus, if you wish to match both CXID1 and PERIOD1 in the source page to CXID2 and PERIOD2 in the second page, you should enter the sets of names in parallel. Here, we choose to match observations using the values of the STATE1 series in the source page and the values of the STATE2 series in the destination page.

Next, there is a checkbox labeled **Treat NA as ID category** for whether to use observations which have NA values in the source and destination ID values. By default, observations are ignored if there are NAs in the ID series; by selecting this option, you instruct EViews to match observations with NA ID values from the source page to observations with NA ID values in the destination page.

Link calculation settings

The remaining options are used when computing the link values.

First, you should specify a source series contraction method. As described in "Linking by general match merging" on page 234, the first step in every match merge is to perform a contraction to ensure uniqueness of the source values. Since contraction is always performed, you should pay attention to your contraction method even when the source IDs are unique, since some settings will not yield the original source data.

There is an extensive list of contractions from which you may choose. For links involving numeric series you may choose to employ obvious methods such as the **Mean** (default) or the **Median** of the observations, or less obvious summary statistics such as the **Variance (population)**, **Std. Deviation (sample)**, **Kurtosis**, **Quantile**, **Number of obs**, or **Number of NAs**.

For links involving alpha series, you must select from a subset of the numeric contractions: **Unique values** (default), **No contractions allowed**, **First**, **Last**, **Maximum**, **Minimum**, **Number of obs**, **Number of NAs**.

Most of these options are self-explanatory, though a few comments about the choice of method may prove useful.

First, there are two options at the bottom of the list which deserve additional explanation. The last choice, **No contractions allowed,** may be used to ensure that contractions are never performed prior in the first step of a link match merge. The option is designed for cases where you believe that your source ID values are unique, and wish the link to generate an error if they are not.

The **Unique values** option provides a less strict version of the **No contractions allowed** setting, allowing for non-unique source ID values so long as any observations with matching IDs share the same source series value. In this case, the contraction will simply identify the unique source value associated with each unique source ID value. If there are observations with a single ID that have more than one source series value, the link will generate an error.

To see the difference between the two settings, note that contracting the following SOURCE and ID series

ID	Source
1	80
1	80
1	80
2	100
2	100

generates an error with the **Unique values** setting, but not with the **No contractions allowed** setting. Alternatively, the SOURCE and ID series

ID	Source
1	80
1	80
1	50
2	100
2	100

generate errors with both contractions.

Second, you should note that if you select **First** or **Last**, EViews will contract the source series by selecting the first or last observation in each set of observations with repeated source IDs. **First** or **Last** is defined here as depending on the order in which the observations appear in the original source workfile. Thus, selecting **First** means that the contracted value for each source ID value will be taken from the first observation in the workfile with that ID value.

Lastly, you should bear in mind that unless you select **No contractions allowed** or **Unique values**, EViews will perform a first stage contraction of the data using the specified settings. In cases where the source ID values are not unique, this contraction is a necessary step; in cases where the source ID values are unique, the contraction is not necessary for the resulting one-to-one or one-to-many match merge, but is performed so that EViews can support more complicated many-to-many merge operations.

For most of the choices, performing a contraction on the unique source data has no practical effect on the outcome of a one-to-one or one-to-many match merge. For example, a choice of any of the data preserving options: **Mean**, **Median**, **Maximum**, **Minimum**, **Sum**, **First**, **Last**, **Unique values**, or **No contractions allowed** will create a link that performs the standard one-to-one or one-to-many match merge of the values of the original source series into the destination page.

On the other hand, selecting a contraction method that alters the source values will create a link that performs a match merge of the summary values into the destination page. Thus, selecting **Sum of Squares**, **Variance (population)**, **Std. Deviation (sample)**, **Skewness**, **Kurtosis**, **Quantile**, **Number of obs**, or **Number of NAs**, will generate link values that differ from those obtained in a traditional one-to-one or one-to-many match merge.

It is worth emphasizing that the default contraction setting, **Mean,** preserves values for data with unique source IDs. Thus, unless you specifically set the contraction method to a non-preserving method, a one-to-one or one-to-many match merge will link the original values into the destination page. You may also ensure that EViews performs the traditional one-to-one or one-to-many match merge by selecting any of the other value preserving transformation methods, or even better by selecting **No contractions allowed** or **Unique values** to validate the IDs.

Finally, in the **Source sample** edit field, you should enter a description of the source sample to be used when constructing link values. By default, the full sample keyword "@ALL" is entered in the field so that EViews will use all of the observations in the source page.

One important application involving sample settings is to restrict the observations over which the contraction is performed prior to performing the match merge. Suppose, for example, that we have a workfile with observations on individuals with state of residence. Then we could construct two links from the individual page to a state page, one of which computes the mean INCOME for males in each state, and another which computes the mean INCOME for females.
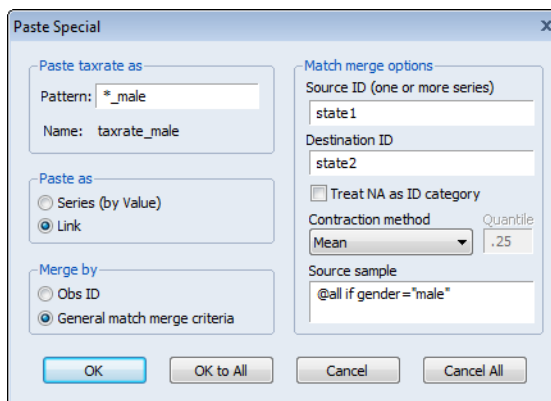
Date match merge links

Dates may be used in matching in two ways: exact matching or date matching (see "Linking by date match merging" on page 239 for details).

Suppose we have a workfile containing the quarterly data on PROFITS described earlier. The quarterly PROFITS data is contained in a regular frequency quarterly workfile page. Also contained in the page is a date series DT generated by taking the first instance in each quarter ("series dt = @date"). We show here DT formatted to show the day-month-year, alongside the PROFIT series.

Contained in a separate, unstructured page are advertising data ADVERT, and another series DT showing the corresponding irregular dates.

If we attempt to match merge these data using the DT date series as identifiers, EViews will use the first method, exact matching, to identify common observations. Thus, if we try to link the PROFIT data into the advertising page using the DT series as the identifiers, we will find that there

are no observations in the quarterly source page that match observations in the irregular daily destination page. The resulting link values will all be NAs.

When one or both of the pages follow a regular frequency, we may instruct EViews to employ date matching. We may do so by using the special ID keyword "@DATE" as an ID in the regular frequency page ID to indicate that we wish to use date matching with the built-in date identifiers given by the structure of the page. In this case, we will use "@DATE" as the ID for the regular frequency quarterly page, and match it against the values in the DT series in the destination page.

In this example, we use the **Paste Special** dialog to instruct EViews to copy the quarterly PROFIT series to a link named PROFIT1 in the destination page. We select **General match merge criteria** and employ date matching to match the quarters in the source page to the values in the DT series in the destination page, rounding to the lowest common frequency.
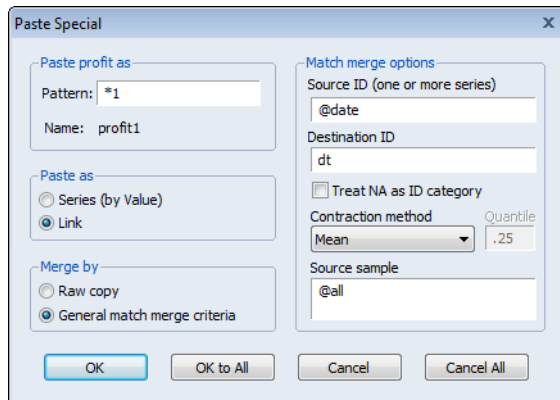
We first compute a **Mean** contraction of the source data for all observations, then match merge the contracted results into the destination. Note that since the match merge in this example is one-to-many, the **Mean** contraction method is irrelevant since it leaves the source data unchanged. If we wish to guarantee that the source IDs are unique, we may change the **Contraction** method to **No contractions allowed**.

In the special case where you have two dated structured pages, you may construct the link using the "@DATE" keyword for both page identifiers. Here, where the advertising page is structured as an (irregular) daily dated page, we could replace DT in the destination index field with the keyword "@DATE".

If "@DATE" is used as an ID in both pages, EViews will use the observation date identifiers associated with the structure of each page, round them to the lowest common frequency, and then find matching observations.

Frequency conversion links

In the special case where we link numeric series between two regular frequency pages, we may copy-and-paste to define a link (or a by value copy of the source series) that employs frequency conversion ("Linking by date with frequency conversion" on page 243). In this setting, the **Paste Special** dialog offers you an additional choice between linking by general match merge, or linking by date using frequency conversion.

If you select **General match merge criteria** in the **Merge by** section of the dialog, the right side of the dialog will change to show the standard match merge version described in "General match merge links" on page 247.

Alternately, to define a frequency conversion link, click on the **Date with frequency conversion** selection. The dialog will change to display the frequency conversion options for converting data both from high to low, and low to high frequency.
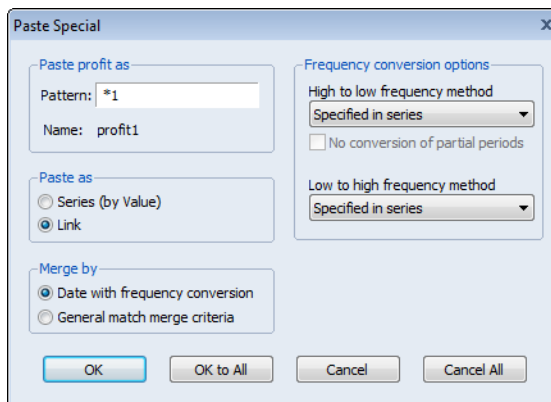
By default, EViews will use the high to low and the low to high conversion methods specified in the original source series.

If you wish to change the high to low conversion methods, simply select the desired setting from the drop-down menu. In addition, if you select one of the non-default methods, choose whether to select the **No conversion of partial periods** checkbox. If this setting is selected, EViews will propagate NAs when performing the frequency conversion so that the average of observations with an NA value will not drop the observation, and will instead generate an NA.

Note that the last conversion method, **No down conversions**, may be used to disallow down frequency conversion of the data. This setting allows you to ensure that when evaluated, the link involves same frequency (one-to-one) or low to high (one-to-many) frequency conversion, otherwise the link evaluation will generate an error.

To set the low to high conversion method, select the desired method from the drop-down menu. Once again, the last frequency conversion method, **No up conversions**, allows you to inform EViews that you expect the link to work only for same frequency,

or high-to-low frequency linking, and that the link evaluation should generate an error if it encounters data requiring up conversion.

Creating a link by command

While the copy-and-paste interface is the easiest approach to specifying a link, we note that you may also create links using the `link` declaration statement and the `linkto` procedure.

You may, at the command line, enter the keyword `link` followed by the name of a new link object. EViews will create a new, incompletely specified, link object in the current (destination) workfile page. The destination page should be active when you enter the command.

You may modify a link specification, defining link IDs, as well as contraction and in some cases, expansion methods using the `linkto` proc.

Consider our earlier example where we link the TAXRATE data from the state page to the individual page. The following command creates a link object in the current workfile page:

```
link taxrate2
```

You may modify the TAXRATE2 link by providing a link definition using the `linkto` procedure. The `linkto` keyword should be followed by the name of the source series and the source and destination IDs, with the latter separated by `@src` and `@dest` keywords. For example, if the link object TAXRATE2 exists in our individual page, the link proc:

```
taxrate2.linkto state::taxrate @src state1 @dest state2
```

instructs EViews to define the link TAXRATE2 so that it uses the TAXRATE series in the source page named "STATE" as the source series, and matches the source page STATE1 values to the current page STATE2 values.

In the special case where there is only one ID series in each page, we may, without introducing ambiguity, omit the `@src` and `@dest` keywords. Here, we may shorten our link definition statement to:

```
taxrate2.linkto state::taxrate state1 state2
```

Lastly, we may combine these declaration and definition statements into one. The command

```
link taxrate2.linkto state::taxrate state1 state2
```

both creates a link object in the active workfile page and defines the source and link ID series.

In this one-to-many example where we link state data to individuals, we need not consider contraction methods as the default (mean) contraction method preserves the original data. If you wish to disallow contractions, or to limit them to cases where the values of the source data are unique, you may use contraction options as in:

```
link taxrate2.linkto(c=none) state::taxrate state1 state2
```

or

```
link taxrate2.linkto(c=unique) state::taxrate state1 state2
```

Conversely, linking the SALES data from the individual page to the state page yields a many-to-one conversion in which the contraction method is important. In this setting, we may optionally specify a contraction method so that when the state page is active, the statement

```
link sales2.linkto(c=sum) indiv::sales state2 state1
```

links the SALES data from the "INDIV" source page, matching the source page STATE2 values to the current page STATE1 values, and contracting observations using the sum transformation. If the contraction option is not provided, EViews will use the mean contraction default.

In the special case where you wish to link your data using date matching, you must use the special keyword @date as an ID series for the regular frequency page. For example, when linking from our quarterly to our advertising page, we may specify:

```
link profit1.linkto quarterly::profit @date dt
```

to tell EViews to link the quarterly page PROFIT data, matching the built-in identifier for the quarter with the date series DT in the destination advertising page.

As in the copy-and-paste interface, the presence of the special @date keyword tells EViews that you wish to perform date matching using the date structure of the corresponding regular frequency page. If @date is not specified as an ID, EViews will employ a general match merge using the specified identifiers.

When linking data between dated regular frequency workfile pages, the linkto proc will perform a frequency conversion link between the two pages *unless* ID series are explicitly provided, or a general match merge specific conversion method (such as variance or kurtosis) is specified. Thus, issuing the command

```
link profit2.linkto quarterly::profit
```

in an annual page, creates a frequency conversion link PROFIT2 using the PROFIT data from the quarterly page. Since no conversion options are provided, EViews will use the default frequency conversion method specified in the quarterly PROFIT series.

If ID series are provided, EViews will perform the link using general match merging. Thus, the closely related command

```
link profit2a.linkto quarterly::profit @date @date
```

will produce a link named PROFIT2A that employs date match merging using the dates in the workfile page structures. Since no conversion options are provided. EViews will use the default match merge contraction method, taking means, to perform the conversion.

If no ID series are specified, but a match merge specific option is provided, "@DATE @DATE" is appended to the ID list, and general match merging is assumed. Thus, the command

```
link profit2b.linkto(c=med) quarterly::profit
```

is equivalent to

```
link profit2b.linkto(c=med) quarterly::profit @date @date
```

since "c = med" is a match merge specific conversion option. This link is evaluated using general match merging, with date matching.

For additional details see `Link` (p. 355) and `linkto` (p. 359) in the *Command and Programming Reference*.

Working with Links

Once a link is defined, you may, for all intents and purposes, use it as though it were an ordinary series or an alpha series.

Links may be identified in the workfile directory by the presence of a pink series or alpha series icon, or by an icon containing a "?". If a link definition uses an ordinary source series, it will appear in the workfile directory with a pink version of the series icon. If a link uses an alpha source series, it will appear with a pink alpha series icon. In both cases, the link may be used as though it were a series of the specified type.

If the link source series is not specified or if its type cannot be identified, the link icon will feature a "?" indicating that the link is unavailable. Undefined links will be classified as numeric series that generate NA values for every observation.

Using links

Links use virtually no memory until used. A link goes into use either when you are examining the contents of the link, when it is placed in a group which evaluates the link, or when the link is used in a series expression. Once a link goes out of use, the memory for the link is cleared and made available for other uses. In this way, links take up only the minimum amount of memory required to perform a given operation.

When links are in use, any modification to the data underlying the link will lead to a reevaluation of the link values. If you modify either the source series, or the source or destination ID series, EViews will automatically recompute the link values. In this way, you may use the link to define an automatically updating match merge or frequency conversion.

For example, suppose that we open a workfile containing the state and individual pages. Here we see the individual page containing the state TAXRATE data linked into the link series TAXRATE2. From the (colored) series icon, we see that TAXRATE2 is a link of a numeric series.

If the TAXRATE2 link is not in use, the link series contains no values and takes up no memory. Links are placed in use either by opening the link window, by placing the link in a group object, or by using the link in a series expression. Whenever the link comes into use, or one of the link components is changed, the link is evaluated, and its values updated as necessary.

For example, if we double click on the TAXRATE2 icon, we open a standard series spreadsheet view. At this point, EViews evaluates the link, performing the match merge operation, and assigning the values to the TAXRATE2 link. Note that the "Last updated" line will show the time that the link values were evaluated.

All of the menus and toolbars are those found in ordinary series—you may work with this link as though it were any ordinary series. Indeed, the only hint you will have that TAXRATE2 is not an ordinary series or alpha series is in the titlebar, which will indicate that we are working with a link object. For example, if you select **View/One-way tabulation...** uncheck the grouping settings, and click on **OK** to continue, EViews will display a frequency tabulation of the contents of the link, just as it would for an ordinary series.
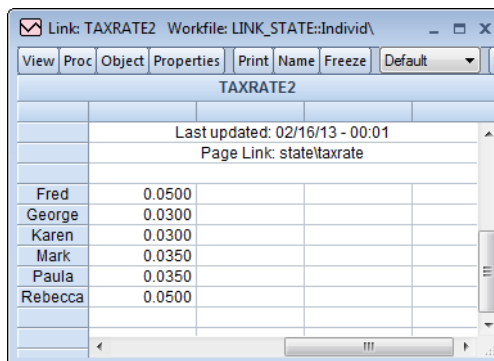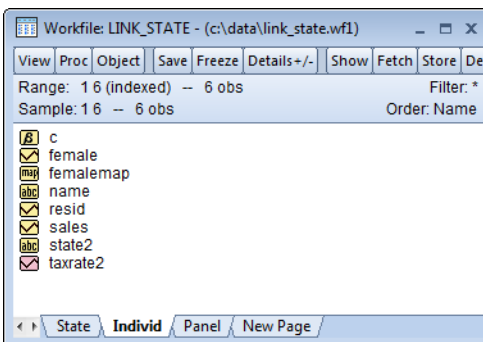
Tabulation of TAXRATE2
Date: 07/25/06   Time: 17:51
Sample: 1 6
Included observations: 6
Number of categories: 3

Value	Count	Percent	Cumulative Count	Cumulative Percent
0.0300	2	33.33	2	33.33
0.0350	2	33.33	4	66.67
0.0500	2	33.33	6	100.00
Total	6	100.00	6	100.00

If you then close the link window, EViews will examine any open windows or existing group objects to see whether the link is still in use. If the link is no longer used, its contents will be cleared and memory will be released. The next time you use the link, it will come into use and will be reevaluated.

Similarly, you may use TAXRATE2 in any place that a series may be used. For example, we may generate a new series, TAXPCT that contains the values of TAXRATE2 expressed in percentage terms:

```
series taxpct = taxrate2 * 100
```

Assuming that TAXRATE2 is not currently in use, EViews will evaluate the link and assign values to each observation, then will multiply each of these values by 100 and assign them to TAXPCT. When the series assignment operation is completed, the values of TAXRATE2 will no longer be used, so that EViews will clear the link contents.

If you attempt to open a link that is improperly defined, either because the source or ID series are not found, or because the observed data require a contraction or frequency conversion method that has been disallowed, EViews will display a link view showing the definition of the link and the error encountered. If you attempt to use this link, you will find that all of the link values are set to NA.
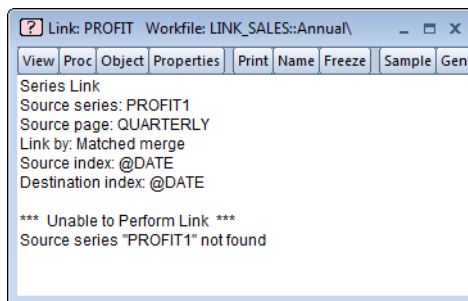
Modifying links

You may, at any time, modify the definition of a link by dialog or command.

To modify a link interactively, we must open the **Link Spec** dialog page. First open the desired link by double clicking on the icon in the workfile directory. Then click on the **Prop-**

erties toolbar button, or select **View/Properties...** from the main menu to bring up the link properties dialog. Lastly, select the **Link Spec** tab.

The **Link Spec** property page is a slightly modified version of the original **Paste Special** dialog used to create links. While the majority of the dialog is unchanged, in place of the destination name, we now have edit fields in which you specify the names of the source series and the source workfile page.
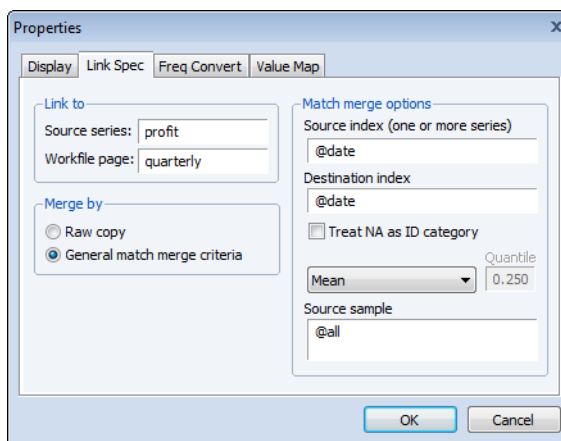
Here we see that the current link uses the PROFIT series in the QUARTERLY page as the source. The link is performed by general match merge, using date matching to link the quarterly dates to the destination series DT. The match merge first performs a mean contraction of the PROFIT series over the entire sample, and then performs the match merge.

To modify the link using the dialog, simply alter any of the dialog settings. For example, we may change the link contraction method from **Mean** to **Minimum** by changing the selection in the **Contraction method** dropdown menu, or we may change the source sample by entering a new sample in the edit box. More fundamental changes in the link will result from changing the source series or workfile page, or any of the match merge identifiers.

To modify a link by command, you may use the `linkto` proc. See "Creating a link by command" on page 255 for details. Issuing a `linkto` proc command for an existing link will replace the existing values with the new specification.

Breaking links

The auto-updating feature is one of the most important characteristics of links. Given the live nature of links, changes to either the source series, or the source or destination IDs will lead EViews to recalculate the values of the link. Links may be used to create auto-updating match merges or frequency conversion of series between workfile pages.

Suppose, for example, that while displaying the TAXRATE2 spreadsheet view, you elect to edit the values in the individual STATE2 ID series. Changing Mark's value for STATE2 from "Texas" to "Arkansas" changes the values of an ID series used to compute the values in TAXRATE2. EViews automatically recomputes TAXRATE2, changing the value for Mark from 0.35 to 0.30, and updates the open spreadsheet view accordingly. Furthermore, any future use of the TAXRATE2 link will use the updated values.

In some circumstances, you may wish to fix the values of the link so that future changes to the source or ID series do not alter the existing values. There are two ways in which you may achieve this result. First, you may simply generate a new series that contains the current values of the link, as in:

```
series fixrate = taxrate2
```

The new ordinary series FIXRATE contains the current values of TAXRATE2. Furthermore, FIXRATE remains unchanged in the face of changes in TAXRATE2. With this method, both the original link series and a new series will be kept in the workfile.

The second method of fixing values is to convert the link into a series. We term this process *unlinking* or *breaking the link*. In this case, the existing link is replaced by a series with the same name, containing the values in the link at the time the link is broken.
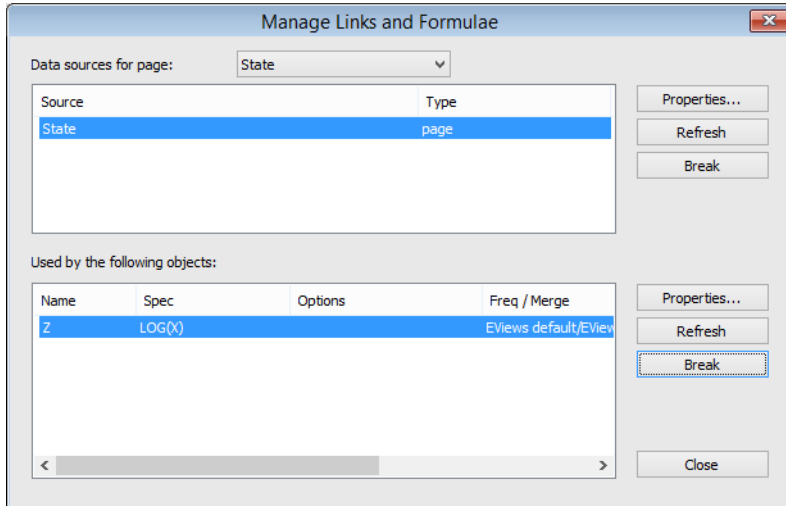
To break a link, simply select **Object/Manage Links & Formulae…** from the workfile window. EViews will display a dialog allowing you to specify the linked objects you wish to break. Either select the source above (to break all objects that use that source) or select the specific objects below then click the associated **Break** button. Bear in mind that the process of unlinking a series link object is irreversible.

Chapter 9. Advanced Workfiles

In Chapter 3. "Workfile Basics," we described the basics of workfiles; how to create and work with a workfile, as well as the basics of using multi-page workfiles. In this chapter, we describe advanced workfile types and tools for working with workfiles.

First, we describe the fundamentals of structured workfiles. You will need to understand the concepts underlying structured workfiles to work with irregular dated data, data involving cross-section identifiers, or panel structures.

Next, we outline various workfile level tools for managing your data. Among other things, we discuss the basics of resizing a workfile, saving a workfile to foreign formats, subsetting a workfile, and rearranging the format of the data in your workfile.

Structuring a Workfile

You may, at any time, change the underlying structure of an existing workfile or workfile page by applying structuring information. We call this process *structuring a workfile*. There are four primary types of structuring information that you may provide:

- regular date descriptions.
- variables containing observation identifiers for dated data.
- variables containing observation identifiers for cross-section data.
- variables containing observation identifiers defining a panel data structure.

The ability to structure your data is an important feature, and we will explore structured workfiles at some length.

Types of Structured Data

Before describing the process of structuring a workfile or workfile page, we define some concepts related to the various data structures.

Regular and Irregular Frequency Data

As the name suggests, *regular frequency data* arrive at regular intervals (daily, monthly, annually, etc.). Standard macroeconomic data such as quarterly GDP or monthly housing starts are examples of regular frequency data. This type of data is introduced in "Creating a Workfile by Describing its Structure" on page 43.

Unlike regular frequency data, *Irregular frequency data* do not arrive in a precisely regular pattern. An important example of irregular data is found in stock and bond prices,

where the presence of missing days due to holidays and other market closures means that the data do not follow a regular daily (7- or 5-day) frequency.

The most important characteristic of regular data is that there are no *structural gaps* in the data—all observations in the specified frequency *exist*, even if there are missing values that are not observed. Alternatively, irregular data allow for gaps between successive observations in the given regular frequency. This is a subtle distinction, but has important consequences for lag processing.

The distinction is best illustrated by an example. Suppose that we are working with a daily calendar and that we have two kinds of data: data on bond prices (BOND), and data on temperature in Los Angeles in Farenheit (TEMP):

Day	Day of Week	Bond	Temp
12/21	Sun	< mkt.closed >	68
12/22	Mon	102.78	70
12/23	Tues	102.79	NA
12/24	Wed	102.78	69
12/25	Thurs	< mkt.closed >	68
12/26	Fri	102.77	70

Notice that in this example, the bond price is not available on 12/21 and 12/25 (since the market was closed), and that the temperature reading was not available on 12/23 (due to equipment malfunction).

Typically, we would view the TEMP series as following a 7-day *regular* daily frequency with a missing value for 12/23. The key feature of this interpretation is that the day 12/23 *exists*, even though a temperature reading was not taken on that day. Most importantly, this interpretation implies that the lagged value of TEMP on 12/24 (the previous day's TEMP value) is NA.

In contrast, most analysts would view BOND prices as following an *irregular* daily frequency in which days involving market closures *do not exist*. Under this interpretation, we would remove weekends and holidays from the calendar so that the bond data would be given by:

Day	Day of Week	Bond
12/22	Mon	102.78
12/23	Tue	102.79
12/24	Wed	102.78
12/26	Fri	102.77

The central point here is that lags are defined differently for regular and irregular data. Given a regular daily frequency, the lagged value of BOND on 12/26 would be taken from the previous day, 12/25, and would be NA. Given the irregular daily frequency, the lagged value on 12/26 is taken from the previous observation, 12/24, and would be 102.78. In defining an irregular calendar, we explicitly skip over the structural gaps created by market closure.

You may always convert irregular frequency data into regular frequency data by adding any observations required to fill out the relevant calendar. If, for example, you have 7-day irregular data, you may convert it to a regular frequency by adding observations with IDs that correspond to any missing days.

Undated Data with Identifiers

Perhaps the simplest data structure involves undated data. We typically refer to these data as *cross-section data*. Among the most common examples of cross-section data are state data taken at a single point in time:

Obs	Year	State	TaxRate
1	2002	Alabama	.000
2	2002	Arkansas	.035
3	2002	Arizona	.035
…	2002	…	…
50	2002	Wyoming	.010

Here we have an alphabetically ordered dataset with 50 observations on state tax rates. We emphasize the point that these data are undated since the common YEAR of observation does not aid in identifying the individual observations.

These cross-section data may be treated as an unstructured dataset using the default integer identifiers 1 to 50. Alternatively, we may structure the data using the unique values in STATE as identifiers. These state name IDs will then be used when referring to or labeling observations. The advantages of using the state names as identifiers should be obvious— comparing data for observation labeled "Arizona" and "Wyoming" is much easier than comparing data for observations "3" and "50".

One last comment about the ordering of observations in cross-section data. While we can (and will) define the lag observation to be that "preceding" a given observation, such a definition is sensitive to the arbitrary ordering of our data, and may not be meaningful. If, as in our example, we order our states alphabetically, the first lag of "Arkansas" is taken from the "Arizona" observation, while if we order our observations by population, the lag of "Arkansas" will be the data for "Utah".

Panel Data

Some data involve observations that possess both cross-section (*group*) and within-cross-section (*cell*) identifiers. We will term these to be *panel data.* Many of the previously encountered data structures may be viewed as a trivial case of panel data involving a single cross-section.

To extend our earlier example, suppose that instead of observing the cross-section state tax data for a single year, we observe these rates for several years. We may then treat an observation on any single tax rate as having two identifiers: a single identifier for STATE (the group ID), and an identifier for the YEAR (the cell ID). The data for two of our states, "Kansas" and "Kentucky" might look like the following:

Obs	State	Year	TaxRate
...	...	...	...
80	Kansas	2001	.035
81	Kansas	2002	.037
82	Kansas	2003	.036
83	Kentucky	2001	.014
84	Kentucky	2003	.016
...	...	...	...

We emphasize again that identifiers must uniquely determine the observation. A corollary of this requirement is that the *cell IDs uniquely identify observations within a group.* Note that requiring cell IDs to be unique within a group does not imply that the cell IDs are unique. In fact, cell ID values are usually repeated across groups; for example, a given YEAR value appears in many states since the tax rates are generally observed in the same years.

If we observe repeated values in the cell identifiers within *any* one group, we must either use a different cell identifier, or we must redefine our notion of a group. Suppose, for example, that Kansas changed its tax rate several times during 2002:

Obs	State	Year	Cell_ID1	Cell_ID2	TaxRate
...	...	...	...	...	...
80	Kansas	2001	1	1	.035
81	Kansas	2002	2	1	.037
82	Kansas	2002	3	2	.038
83	Kansas	2002	4	3	.035
84	Kansas	2003	5	1	.036
85	Kentucky	2001	1	1	.014

86	Kentucky	2003	2	2	.016
…	…	…	…	…	…

In this setting, YEAR would not be a valid cell ID for groups defined by STATE, since 2002 would be repeated for STATE = "Kansas".

There are a couple of things we may do. First, we may simply choose a different cell identifier. We could, for example, create a variable containing a default integer identifier running within each cross-section. For example, the newly created variable CELL_ID1 is a valid cell ID since it provides each "Kansas" and "Kentucky" observation with a unique (integer) value.

Alternately, we may elect to subdivide our groups. We may, for example, choose to use *both* STATE *and* YEAR as the group identifier. This specification defines a group for each unique STATE and YEAR combination (*e.g.* — observations for which STATE = "Kansas" *and* YEAR = "2002" would comprise a single group). Given this new group definition, we may use *either* CELL_ID1 or CELL_ID2 as cell identifiers since they are both unique for each STATE and YEAR group. Notice that CELL_ID2 could not have been used as a valid cell ID for STATE groups since it does not uniquely identify observations within Kansas.

While it may at first appear to be innocuous, the choice between creating a new variable or redefining your groups has important implications (especially for lag processing). Roughly speaking, if you believe that observations within the original groups are closely "related", you should create a new cell ID; if you believe that the subdivision creates groups that are more alike, then you should redefine your group IDs.

In our example, if you believe that the observations for "Kansas" in "2001" and "2002" are both fundamentally "Kansas" observations, then you should specify a new cell ID. On the other hand, if you believe that observations for "Kansas" in "2002" are very different from "Kansas" in "2001", you should subdivide the original "Kansas" group by using both STATE and YEAR as the group ID.

Lags, Leads, and Panel Structured Data

Following convention, the observations in our panel dataset are always *stacked by cross-section*. We first collect the observations by cross-section and sort the cell IDs within each cross-section. We then stack the cross sections on top of one another, with the data for the first cross-section followed by the data for the second cross-section, the second followed by the third, and so on.

The primary impact of this data arrangement is its effect on lag processing. There are two fundamental principles of lag processing in panel data structures:

- First, lags and leads do not cross group boundaries, so that they never use data from a different group.

- Second, lags and leads taken within a cross-section are defined over the sorted values of the cell ID. This implies that lags of an observation are always associated with lower value of the cell ID, and leads always involve a higher value (the first lag observation has the next lowest value and the first lead has the next highest value).

Let us return to our original example with STATE as the group ID and YEAR as the cell ID, and consider the values of TAXRATE, TAXRATE(-1), and TAXRATE(1). Applying the two rules for panel lag processing, we have:

Obs	State	Year	TaxRate	TaxRate(-1)	TaxRate(1)
...	...	...	...	...	...
80	Kansas	2001	.035	NA	.037
81	Kansas	2002	.037	.035	.036
82	Kansas	2003	.036	.037	NA
83	Kentucky	2001	.014	NA	.016
84	Kentucky	2003	.016	.014	NA
...	...	...	...		

Note in particular, that the lags and leads of TAXRATE do not cross the group boundaries; the value of TAXRATE(-1) for Kentucky in 2001 is an NA since the previous value is from Kansas, and the value TAXRATE(1) for Kansas in 2003 is NA is the next value is from Kentucky.

Next, consider an example where we have invalid IDs since there are duplicate YEAR values for Kansas. Recall that there are two possible solutions to this problem: (1) creating a new cell ID, or (2) redefining our groups. Here, we see why the choice between using a new cell ID or subdividing groups has important implications for lag processing. First, we may simply create a new cell ID that enumerates the observations in each state (CELL_ID1). If we use CELL_ID1 as the cell identifier, we have:

Obs	State	Year	Cell_ID1	TaxRate	TaxRate(-1)
...	...	...	...	...	
80	Kansas	2001	1	.035	NA
81	Kansas	2002	2	.037	.035
82	Kansas	2002	3	.038	.037
83	Kansas	2002	4	.035	.038
84	Kansas	2003	5	.036	.035
85	Kentucky	2001	1	.014	NA
86	Kentucky	2003	2	.016	.014

...	...	...	...	...	

Note that the only observations for TAXRATE(-1) that are missing are those at the "seams" joining the cross-sections.

Suppose instead that we elect to subdivide our STATE groupings by using both STATE and YEAR to identify a cross-section, and we create CELL_ID2 which enumerates the observations in each cross-section. Thus, each group is representative of a unique STATE-YEAR pair, and the cell ID indexes observations in a given STATE for a specific YEAR. The TAXRATE(-1) values are given in:

Obs	State	Year	Cell_ID2	TaxRate	TaxRate(-1)
...	...	...	...	...	...
80	Kansas	2001	1	.035	NA
81	Kansas	2002	1	.037	NA
82	Kansas	2002	2	.038	.037
83	Kansas	2002	3	.035	.038
84	Kansas	2003	1	.036	NA
85	Kentucky	2001	1	.014	NA
86	Kentucky	2003	2	.016	.014
...	...	...	...	...	...

Once again, the missing observations for TAXRATE(-1) are those that span cross-section boundaries. Note however, that since the group boundaries are now defined by STATE and YEAR, there are more seams and TAXRATE(-1) has additional missing values.

In this simple example, we see the difference between the alternate approaches for handling duplicate IDs. Subdividing our groups creates additional groups, and additional seams between those groups over which lags and leads are not processed. Accordingly, if you wish your lags and leads to span all of the observations in the original groupings, you should create a new cell ID to be used with the original group identifier.

Types of Panel Data

Panel data may be characterized in a variety of ways. For purposes of creating panel workfiles in EViews, there are several concepts that are of particular interest.

Dated vs. Undated Panels

We characterize panel data as *dated* or *undated* on the basis of the cell ID. When the cell ID follows a frequency, we have a *dated panel* of the given frequency. If, for example, our cell IDs are defined by a variable like YEAR, we say we have an *annual panel*. Similarly, if the cell IDs are quarterly or daily identifiers, we say we have a quarterly or daily panel.

Alternatively, an *undated panel* uses group specific default integers as cell IDs; by default the cell IDs in each group are usually given by the default integers (1, 2, ...).

Regular vs. Irregular Dated Panels

Dated panels follow a regular or an irregular frequency. A panel is said to be a *regular frequency panel* if the cell IDs for every group follow a regular frequency. If one or more groups have cell ID values which do not follow a regular frequency, the panel is said to be an *irregular frequency panel*.

One can convert an irregular frequency panel into a regular frequency panel by adding observations to remove gaps in the calendar for all cross-sections. Note that this procedure is a form of internal balancing (see "Balanced vs. Unbalanced Panels" below) which uses the calendar to determine which observations to add instead of using the set of cell IDs found in the data.

See "Regular and Irregular Frequency Data" on page 263 for a general discussion of these topics.

Balanced vs. Unbalanced Panels

If every group in a panel has an identical set of cell ID values, we say that the panel is *fully balanced*. All other panel datasets are said to be *unbalanced*.

In the simplest form of balanced panel data, every cross-section follows the same regular frequency, with the same start and end dates—for example, data with 10 cross-sections, each with annual data from 1960 to 2002. Slightly more complex is the case where every cross-section has an identical set of irregular cell IDs. In this case, we say that the panel is balanced, but irregular.

We may balance a panel by adding observations to the unbalanced data. The procedure is quite simple—for each cross-section or group, we add observations corresponding to cell IDs that are not in the current group, but appear elsewhere in the data. By adding observations with these "missing" cell IDs, we ensure that all of the cross-sections have the same set of cell IDs.

To complicate matters, we may *partially* balance a panel. There are three possible methods—we may choose to balance between the starts and ends, to balance the starts, or to balance the ends. In each of these methods, we perform the procedure for balancing data described above, but with the set of relevant cell IDs obtained from a subset of the data. Performing all three forms of partial balancing is the same as fully balancing the panel.

Balancing data *between the starts and ends* involves adding observations with cell IDs that are not in the given group, but are both observed elsewhere in the data and lie between the start and end cell ID of the given group. If, for example, the earliest cell ID for a given group is "1985m01" and the latest ID is "1990m01", the set of cell IDs to consider adding is taken

from the list of observed cell IDs that lie between these two dates. The effect of balancing data between starts and ends is to create a panel that is *internally balanced,* that is, balanced for observations with cell IDs ranging from the *latest* start cell ID to the *earliest* end cell ID.

A simple example will better illustrate this concept. Suppose we begin with a two-group panel dataset with the following data for the group ID (INDIV), and the cell ID (YEAR):

Indiv	Year	Indiv	Year
1	1985	2	1987
1	1987	2	1989
1	1993	2	1992
1	1994	2	1994
1	1995	2	1997
1	1996	2	2001

For convenience, we show the two groups side-by-side, instead of stacked. As depicted, these data represent an unbalanced, irregular, annual frequency panel. The data are unbalanced since the set of observed YEAR identifiers are not common for the two individuals; *i.e.* — "1985" appears for individual 1 (INDIV = "1"), but does not appear for individual 2 (INDIV = "2"). The data are also irregular since there are gaps in the yearly data for both individuals.

To balance the data between starts and ends, we first consider the observations for individual 1. The earliest cell ID for this cross-section is "1985" and the latest is "1996". Next, we examine the remainder of the dataset to obtain the cell IDs that lie between these two values. This set of IDs is given by {"1987," "1989," "1992," "1994"}. Since "1989" and "1992" do not appear for individual 1, we add observations with these two IDs to that cross-section. Likewise, for group 2, we obtain the cell IDs from the remaining data that lie between "1987" and "2001". This set is given by {"1993," "1994," "1995," "1996"}. Since "1993," "1995," and "1996" do not appear for individual 2, observations with these three cell IDs will be added for individual 2.

The result of this internal balancing is an expanded, internally balanced panel dataset containing:

Indiv	Year	Indiv	Year
1	1985	2	—
1	1987	2	1987
1	*1989	2	1989
1	*1992	2	1992

1	1993	2	*1993
1	1994	2	1994
1	1995	2	*1995
1	1996	2	*1996
1	—	2	1997
1	—	2	2001

We have marked the five added observations with an asterisk, and arranged the data so that the cell IDs line up where possible. Observations that are not present in the dataset are marked as "—". Notice that the effect of the internal balancing is to fill in the missing cell IDs in the central portion of the data.

It is worth a digression to note here that an alternative form of internal balancing is to add observations to remove all gaps in the calendar between the starts and ends. This method of balancing, which converts the data from an irregular to a regular panel, uses the calendar to determine which observations to add instead of using the set of observed cell IDs found. If we are balancing the expanded dataset, we would add observations with the cell IDs for missing years: {"1986," "1988," "1990," "1991"} for individual 1, and {"1988," "1990," "1991," "1998," "1999," "2000"} for individual 2.

Lastly, we consider the effects of choosing to *balance the starts* or *balance the ends* of our data. In the former case, we ensure that every cross-section adds observations corresponding to observed cell IDs that come before the current starting cell ID. In this case, balancing the starts means adding an observation with ID "1985" to group 2. Similarly, balancing the ends ensures that we add, to every cross-section, observations corresponding to observed cell IDs that follow the cross-section end cell ID. In this case, balancing the ends involves adding observations with cell IDs "1997" and "2001" to group 1.

Nested Panels

While cell IDs must uniquely identify observations within a group, they typically contain values that are repeated across groups. A *nested panel* data structure is one in which the cell IDs are nested, so that they are unique both within and across groups. When cell IDs are nested, they uniquely identify the individual observations in the dataset.

Consider, for example, the following nested panel data containing identifiers for both make and model of automobile:

Make	**Model**
Chevy	Blazer
Chevy	Corvette
Chevy	Astro

Ford	Explorer
Ford	Focus
Ford	Taurus
Ford	Mustang
Chrysler	Crossfire
Chrysler	PT Cruiser
Chrysler	Voyager

We may select MAKE as our group ID, and MODEL as our cell ID. MODEL is a valid cell ID since it clearly satisfies the requirement that it uniquely identify the observations within each group. MODEL is also nested within MAKE since each cell ID value appears in exactly one group. Since there are no duplicate values of MODEL, it may be used to identify every observation in the dataset.

There are a number of complications associated with working with nested panel data. At present, EViews does not allow you to define a nested panel data structure.

Applying a Structure to a Workfile

To structure an existing workfile, select **Proc/"Structure/Resize Current Page…"** in the main workfile window, or double-click on the portion of the window displaying the current range ("Range:").

Selecting a Workfile Type

EViews opens the **Workfile structure** dialog. The basic structure of the dialog is quite similar to the **Workfile create** dialog ("Creating a Workfile" on page 42). On the left-hand side is a dropdown menu where you will select a structure type.
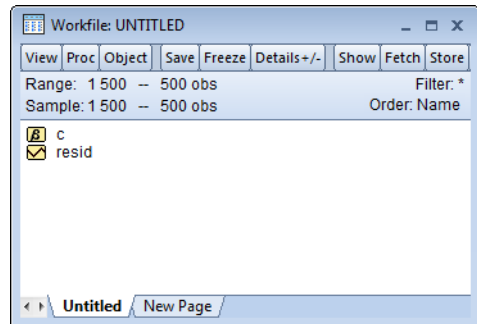
Clicking on the structure type dropdown menu brings up several choices. As before, you may choose between the **Unstructured/Undated**, and **Dated - regular frequency** types. There are, however, several new options. In the place of **Balanced Panel**, you have the option to select from **Dated - specified by date series**, **Dated Panel**, **Undated with ID series**, or **Undated Panel**.
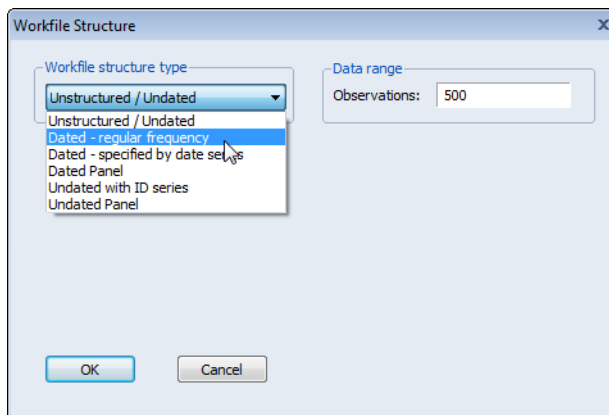
Workfile Structure Settings

As you select different workfile structure types, the right-hand side of the dialog changes to show relevant settings and options for the selected type. For example, if you select the **Dated - regular frequency type**, you will be prompted to enter information about the frequency of your data and date information; if you select an **Undated Panel**, you will be prompted for information about identifiers and the handling of balancing operations.

Dated - Regular Frequency

Given an existing workfile, the simplest method for defining a regular frequency structured workfile is to select **Dated - regular frequency** in the structure type dropdown menu. The right side of the dialog changes to reflect your choice, prompting you to describe your data structure.

You are given the choice of a **Frequency**, as well as a **Start date** and **End date**. The only difference between this dialog and the workfile create version is that the **End date** field is pre-filled with "@LAST". This default reflects the fact that given a start date and the number of observations in the existing workfile, EViews can calculate the end date implied
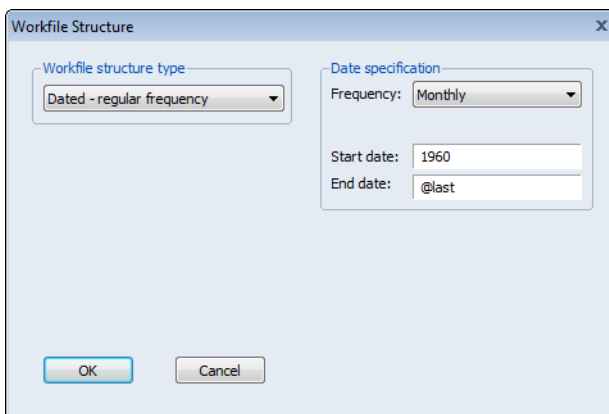
by "@LAST". Alternatively, if we provide an ending date, and enter "@FIRST" in the **Start date** field, EViews will automatically calculate the date associated with "@FIRST".

If we fill out the desired fields and click on **OK**, EViews will restructure the workfile. In this example, we have specified a monthly frequency starting in 1960m01 and continuing until "@LAST". There are exactly 500 observations in the workfile since the end date was calculated to match the existing workfile size.
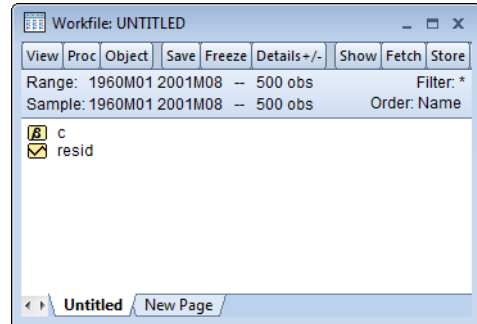
Alternatively, we might elect to enter explicit values for both the starting and ending dates. In this case, EViews will calculate the number of observations implied by these dates and the specified frequency. If the number does not match the number of observations in the existing workfile, you will be informed of this fact, and prompted to continue. If you choose to proceed, EViews will both restructure and resize the workfile to match your specification.

One consequence of this behavior is that resizing a workfile is a particular form of restructuring. To resize a workfile, simply call up the **Workfile structure** dialog, and change the beginning or ending date.

Here we have changed the **End date** from "2011m08" to "2011m12", thereby instructing EViews to add 4 observations to the end of the workfile. If you select **OK**, EViews will inform you that it will add 4 observations and prompt you to continue. If you proceed, EViews will resize the workfile to your specification.
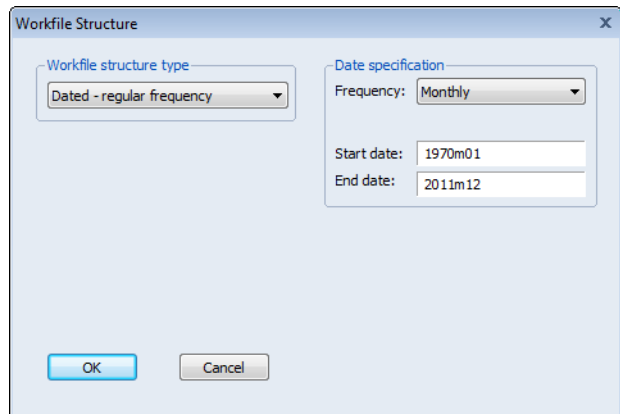
Specifying Start and End Times

The method used to specify the start and end dates or times should be discussed a bit further. Time periods are generally specified by intervals, such that an hour is represented by the specification 00:00 to 59:59. Thus, a full 24 hour day is defined with a start time of 00:00:00 and an end time of 23:59:59. You will notice that for intraday data, the defaults provided by EViews when you create a workfile reflect this. Similarly, a seven day week can be

defined from Monday at 00:00:00 through Sunday at 23:59:59. One second after this ending time, 24:00:00, refers to the following Monday at midnight and is the first second of the next period.
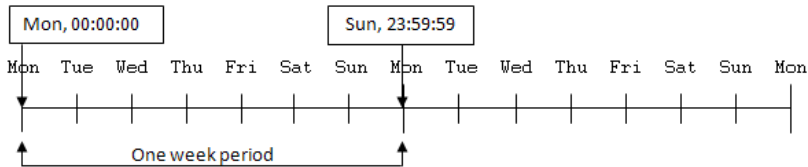
Not only does specifying an end time of 24:00:00 extend into the next day by one second, it will extend by an amount relative to the frequency being defined for the workfile. For instance, given an hourly workfile, an end time of 24:00:00 would add an extra hour to each day, since extending even a second into the next period adds a full interval.

In general, when specifying a range of time for observations, time can be looked at in terms of intervals or in terms of single measurements. For example, the time period 9 a.m. to 12 p.m. can be considered to be three hourly intervals: 9 a.m. to 10 a.m., 10 a.m. to 11 a.m, and 11 a.m. to 12 p.m. Alternately, it could be considered as four on-the-hour measurements: the first at 9 a.m., the second at 10 a.m., the third at 11 a.m., and the fourth at 12 p.m. While the interval model may be more frequently used for continuous measurements over time, it is really up to you to decide which model fits your data better.

In the first case, which defines three intervals from 9 a.m. to 12 p.m., you would specify your workfile with a start time of 9 a.m. and an end time of 11 a.m. This may not seem intuitive, but remember that specifying an end time of 12 p.m. would add an additional hour, defining the interval from 12 p.m. to 1 p.m. In fact, you could specify any end time from 11:00:00 to 11:59:59 for the third interval.

If we wish to look at our data in terms of four discrete hourly measurements, as in the second case, we would specify our workfile with a start time of 9 a.m. and an end time of 12 p.m. Our data points could then be measured at 9 a.m., 10 a.m., 11 a.m., and 12 p.m.

The distinction between thinking of time in terms of intervals or as discrete measurements is subtle. Generally, simply remember that the starting time will be defined as the first observation, and the following observations are defined by the length of time between the starting time and each subsequent time period. Specifying a start time of 9 a.m. and an end time of 9 a.m. will generate a workfile with one observation per day. To determine the number of observations, subtract the start time from the end time and add one for the first observation.

Dated - specified by date series

The second approach to structuring your workfile is to provide the name of a series containing the dates (or series than may be interpreted as dates) to be used as observation identifiers. Select **Dated - specified by date series** in the dropdown menu, and fill out the remainder of the dialog.

The first thing you must do is enter the name of one or more **Date series** that describe the unique date identifiers.

The series may contain EViews date values (a true date series), or the single or multiple series may contain numeric or string representations of unique dates. In the latter case, EViews will create a single date series containing the date values associated with the numeric or string representations. This new series, which will be given a name of the form DATEID##, will be used as the identifier series.

On the right side of the dialog, you will specify additional information about your workfile structure. In the first dropdown menu, you will choose one of the standard EViews workfile frequencies (annual, quarterly, monthly, etc.). As shown in the image, there is an additional (default) option, **Auto detect**, where EViews attempts to detect the frequency of your data from the values in the specified series. In most cases you should use the default; if, however, you choose to override the auto-detection, EViews will associate the date values in the series with observations in the specified frequency.

You may elect to use the EViews defaults, "@FIRST" and "@LAST", for the **Start date** and the **End date**. In this case, the earliest and latest dates found in the identifier series will be used to define the observations in the workfile. Alternatively, you may specify the start and end dates explicitly. If these dates involve resizing the workfile, you will be informed of this fact, and prompted to continue.

The last option is the **Insert empty obs** checkbox. This option should be used if you wish to ensure that you have a regular frequency workfile. If this option is selected, EViews will add any observations necessary to remove gaps in the calendar at the given frequency. If the option is not selected, EViews will use only the observed IDs in the workfile and the workfile may be structured as an irregular workfile.
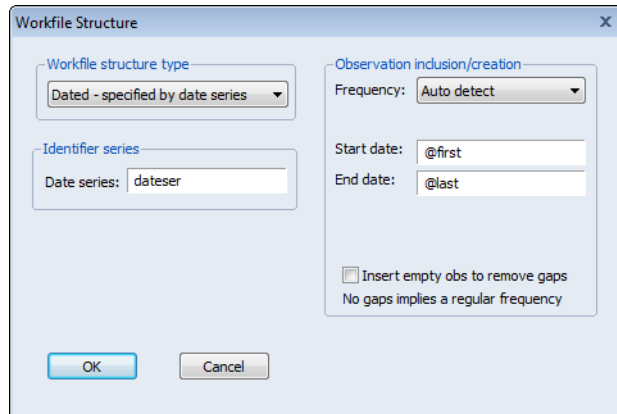
Suppose, for example, that you have observation with IDs for the quarters 1990Q1, 1990Q2, 1990Q4, but not 1990Q3. If **Insert empty obs** is checked, EViews will remove the gap in the calendar by adding an observation corresponding to 1990:3. The resulting workfile will be structured as a regular quarterly frequency workfile. If you do not insert observations, the workfile will be treated as an irregular quarterly workfile.

Once you click on **OK**, EViews will first look for duplicate observation IDs. If duplicates are not found, EViews will sort the data in your workfile by the values in the date series and define the specified workfile structure. In addition, the date series is locked so that it may not be altered, renamed, or deleted so long as it is being used to structure the workfile.

To illustrate the process of structuring a workfile by an ID series, we consider a simple example involving a 10 observation unstructured workfile.

Suppose that the workfile contains the alpha series B consisting of string representations of dates, as depicted. The first thing you should notice about B is that the years are neither complete, nor ordered—there is, for example, no "1962," and "1965" precedes "1961". You should also note that since we have an unstructured workfile, the observation identifiers used to identify the rows of the table are given by the default integer values.

From the workfile window we call up the **Workfile structure** dialog, select **Dated - specified by date series** as our workfile type, and enter the name "B" in the **Date series** edit box. We will start by leaving all of the other settings at their defaults: the frequency is set at **Auto detect**, and the start and end dates are given by "@FIRST" and "@LAST".

The resulting (structured) workfile window shown here indicates that we have a 10 observation irregular annual frequency workfile that ranges from an earliest date of 1960 to the latest date of 1976

Since the series B contained only text representations of dates, EViews has created a new series DATEID containing date values corresponding to

those in B. DATEID is locked and cannot be altered, renamed, or deleted so long as it is used to structure the workfile.

Here, we show a group containing the original series B, the new series DATEID, and the lag of B, B(-1). There are a few things to note. First, the observation identifiers are no longer integers, but instead are values taken from the identifier series DATEID. The formatting of the observation labels will use the display formatting present in the ID series. If you wish to change the appearance of the labels, you should set the display format for DATEID (see "Display Formats" on page 125).

Second, since we have sorted the contents of the workfile by the ID series, the values in B and DATEID are ordered by date. Third, the lagged values of series use the irregular calendar defined by DATEID—for example, the lag of the 1965 value is given by 1961.

Alternately, we could have chosen to restructure with the **Insert empty obs** checkbox selected, thus ensuring that we have a regular frequency workfile.
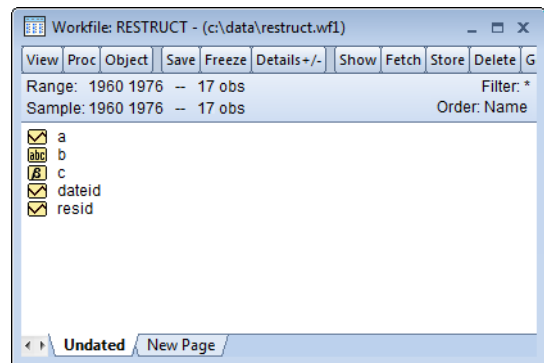
To see the effect of this option, we may reopen the **Workfile structure** dialog by double clicking on the "Range:" string near the top of the workfile window, selecting the **Insert empty obs to remove gaps** option, and then clicking on **OK**. EViews will inform us that the restructure option involves creating 7 additional observations, and will prompt us to continue. Click on **OK** again to proceed. The workfile will be converted to a regular frequency annual workfile with observations from 1960 to 1976.

We again show the group containing B, DATEID, and B(-1). Notice that while the observation identifiers and DATEID now include values for the previously missing dates, B and B(-1), do not. When EViews adds observations in the restructure operation, it sets all ordinary series values to NA or missing for those new observations. You are responsible for filling in values as desired.

Dated Panels

To create a dated panel workfile, you should call up the **Workfile structure** dialog then select **Dated Panel** as our structure type.

There are three parts to the specification of a dated panel. First, you must specify one or more **Date series** that describe date identifiers that are unique within each group. Next, you must specify the **Cross-section ID series** that identify members of a given group. Lastly, you should set options which govern the choice of frequency of your dated data, starting and end-ing dates, and the adding of observations for balancing the panel or ensuring a regular fre-quency.

Dated Panel Basics

We begin by considering the Grunfeld data that have been described in a number of places (see, for example, Baltagi (2005), *Econometric Analysis of Panel Data, Third Edition*, from which this version of the data has been taken). The data measure R&D expenditure and other economic measures for 10 firms for the years 1935 to 1954. These 200 observations form a balanced panel dataset. We begin by reading the data into an unstructured, 200 observation workfile.
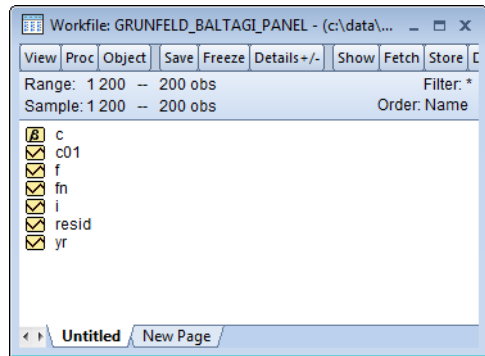
To structure the panel for these data, we call up the **Workfile structure** dialog, select **Dated Panel** as our structure type, and enter the name of the **Cross-section ID series** representing firm number, FN, along with the **Date series** (cell ID) representing the year, YR. If we leave the remaining settings at their default values, EViews will auto detect the frequency of the panel, setting the start and end dates on the basis of the values in the YR series, and will add any observations necessary so that the data between the starts and ends is balanced.

When you click on **OK** to accept these settings, EViews creates a DATEID series, sorts the data by ID and DATEID, locks the two series, and applies the structure. The auto detecting of the date frequency and endpoints yields an annual (balanced) panel beginning in 1935 and ending in 1954.
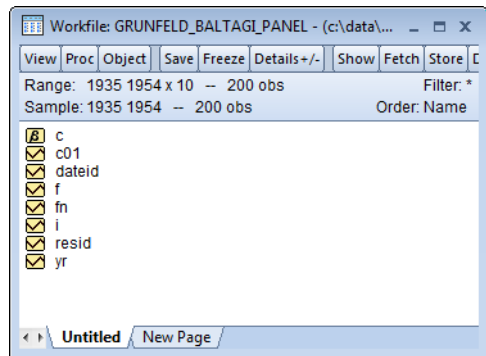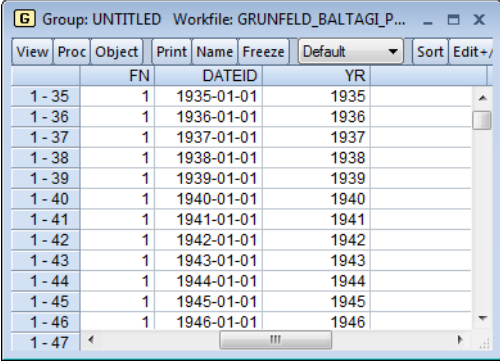
The basic information about this structure is displayed at the top of the workfile window. There are a total of 200 observations representing a balanced panel of 10 cross-sections with data from 1935 to 1954.

Notice that the observation labels for the structured panel workfile show both the group identifier and the cell identifier.

Dated Panel Balancing

In the basic Grunfeld example, the data originally formed a balanced panel so the various balance operations have no effect on the resulting workfile. Similarly, the option to insert observations to remove gaps has no effect since the data already follow a regular (annual) frequency with no gaps.

Let us now consider a slightly more complicated example involving panel data that are both unbalanced and irregular. For simplicity, we have created an unbalanced dataset by taking a 150 observation subset of the 200 observations in the Grunfeld dataset.

First, we call up the **Workfile structure** dialog and again select **Dated Panel**. We begin by using FN and YR as our group and cell IDs, respectively. Use **Auto detect** to determine the frequency, do not perform any balancing, and click on **OK**. With these settings, our workfile will be structured as an unbalanced, irregular, annual workfile ranging from 1935 to 1954.
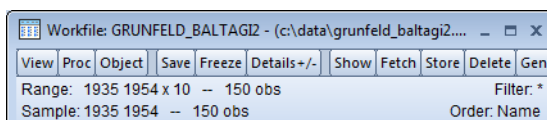
Alternatively, we can elect to perform one or more forms of balancing either at the time the panel structure is put into place, or in a restructure step. Simply call up the **Workfile structure** dialog and select the desired forms of balancing. If you have previously structured your workfile, the dialog will be pre-filled with the existing identifiers and frequency. In this example, we will have our existing annual panel structure with identifiers DATEID and FN.

In addition to choosing whether to **Balance starts** and **Balance ends**, you may choose, at most, one of the two options **Balance between starts and ends**, and **Insert obs to remove date gaps so date follows regular frequency**.

If balancing between starts and ends, the balancing procedure will use the observed cell IDs (in this case, the years encoded in DATEID for all cross-sections) between a given start and end date. All cross-sections will share the same possibly irregular calendar for observations between their starts and ends. If you also elect to insert observations to remove date gaps, EViews balances each cross-section between starts and ends using every date in the calendar for the given frequency. In the latter case, all cross-sections share the same regular calendar for observations between their starts and ends.

Selecting all three options, **Balance starts**, **Balance ends** and **Balance between starts and ends,** ensures a balanced panel workfile. If we substitute the option **Insert obs to remove date gaps so date follows regular frequency** for **Balance between starts and ends**, we further guarantee that the data follow a regular frequency.

In partly or fully balancing the panel workfile, EViews will add observations as necessary, and update the corresponding data in the identifier series. All other variables will have their values for these observations set to NA. Here, we see that EViews has added data for the two identifier series FN and DATEID while the ordinary series YR values associated with the added observations are missing.

Undated with ID series

If you wish to provide cross-section identifiers for your undated data, select **Undated with identifier series** in the dropdown menu.

EViews will prompt you to enter the names of one or more ID series. When you click on **OK**, EViews will first sort the workfile by the values of the ID series, and then lock the series so that it may not be altered so long as the

structure is in place. The values of the ID series will now be used in place of the default integer identifiers.
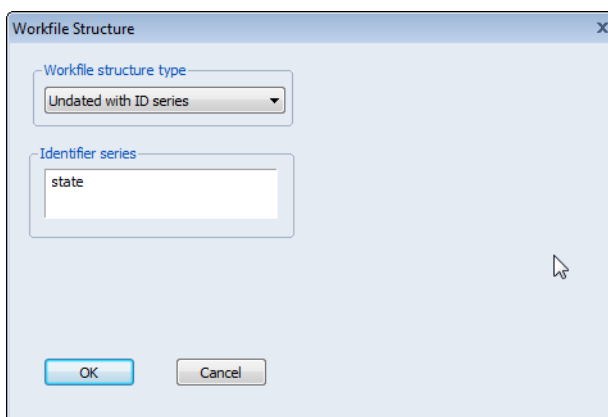
Let us consider a simple example. Suppose that we have a 52 observation unstructured workfile, with observations representing the 50 states in the U.S., D.C., and Puerto Rico.

We wish to use the values in the alpha series STATE (which contains the standard U.S. Postal Service abbreviations) to identify the observations. The data for STATE and a second series, X, are displayed here. Notice that the data are ordered from low to high values for X.
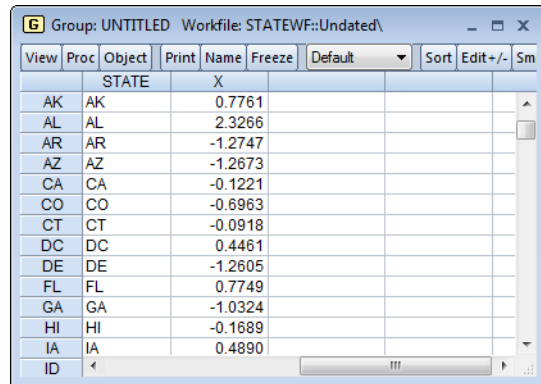
Simply select **Undated with identifier series**, enter "state" as the identifier series, and click **OK** to accept the settings. EViews will sort the observations in the workfile by the values in the ID series, and then apply the requested structure, using and locking down the contents of STATE.

Visually, the workfile window will change slightly with the addition of the description "(indexed)" to the upper portion of the window, showing that the workfile has been
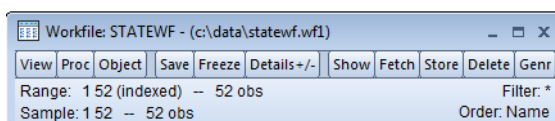
structured. Note, however, that since the dataset is still undated, the workfile range and sample are still expressed in integers ("1 52").

To see the two primary effects of structuring cross-section workfiles, we again examine the values of STATE and the variable X. Notice that the data have been sorted (in ascending order) by the value of STATE and that the observation identifiers in the left-hand border now use the values of STATE.
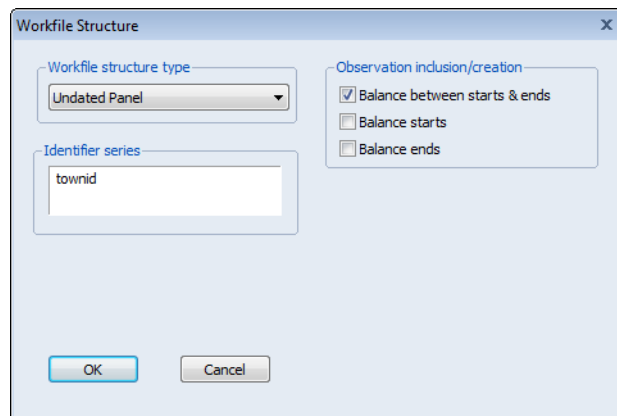
Note that as with irregular structured workfiles, the observation labels will adopt the characteristics of the classifier series display format. If you wish to change the appearance of the observation labels, you should set the spreadsheet display format for STATE (see "Changing the Spreadsheet Display" on page 124).

Undated Panels

To apply an undated panel structure to your workfile, you must specify one or more **Cross-section ID series** that identify members of a given group. First, select Undated Panel from the dropdown menu, and then enter the names of your **identifier series**. You may optionally instruct EViews to balance between the starts and ends, the starts, or the ends of your data.

As an example, we consider the Harrison and Rubinfeld data on house prices for 506 observations located in 92 towns and cities in the harbor area near New Bedford, MA (Harrison and Rubinfeld 1978; Gilley and Pace 1996).

The group identifiers for these data are given by the series TOWNID, in which the town for a given observation is coded from 1 to 92. Observations within a town are not further identified, so there is no cell ID within the data. Here we specify only the group identifier TOWNID.

When we click on **OK**, EViews analyzes the data in TOWNID and determines that there are duplicate observations—there are, for example, 22 observations with a TOWNID of 5. Since TOWNID does not uniquely identify the individual observations, EViews prompts you to create a new cell ID series.
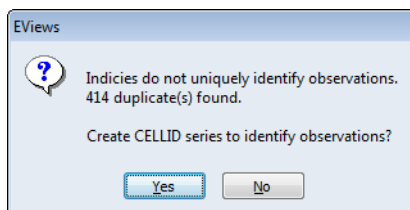
If you click on **No**, EViews will return you to the specification page where you may define a different set of group identifiers. If you choose to continue, EViews will create a new series with a name of the form CELLID## (*e.g.,* CELLID, CELLID01, CELLID02, *etc.*) containing the default integer cell identifiers. This series will automatically be used in defining the workfile structure.

There are important differences between the two approaches (*i.e.*, creating a new ID series, or providing a second ID series in the dialog) that are discussed in "Lags, Leads, and Panel Structured Data" on page 267. In most circumstances, however, you will click on **Yes** to continue. At this point, EViews will inform you that you have chosen to define a two-dimensional, undated panel, and will prompt you to continue. In this example, the data are unbalanced, which is also noted in the prompt.
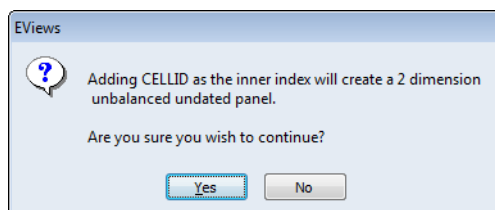
When you click on **Yes** to continue, EViews will restructure the workfile using the identifiers TOWNID and CELLID##. The data will be sorted by the two identifiers, and the two-dimensional panel structure applied. The workfile window will change to show this restructuring. As depicted in the upper portion, we have a 506 observation, undated panel with dimension (92, 30)—92 groups with a maximum of 30 observations in any group.

Note that in this example, balancing the starts or interiors has no effect on the workfile since CELLID## has cell IDs that begin at 1 and run consecutively for every group. If, however, we choose to balance the ends, which vary between 1 and 30, the corresponding resize operation would add 2254 observations. The final result would be a workfile with 2760 observations, comprised of 92 groups, each with 30 observations.
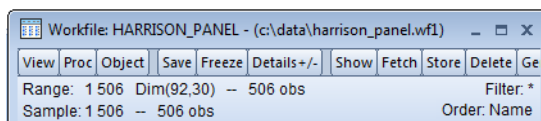
Common Structuring Errors

In most settings, you should find that the process of structuring your workfile is relatively straightforward. It is possible, however, to provide EViews with identifier information that

contains errors so that it is inconsistent with the desired workfile structure. In these cases, EViews will either error, or issue a warning and offer a possible solution. Some common errors warrant additional discussion.

Non-unique identifiers

The most important characteristic of observation identifiers is that they uniquely identify every observation. If you attempt to structure a workfile with identifiers that are not unique, EViews will warn you of this fact, will offer to create a new cell ID, and will prompt you to proceed. If you choose to proceed, EViews will then prompt you to create a panel workfile structure using both the originally specified ID(s) and the new cell ID to identify the observations. We have seen an example of this behavior in our discussion of the undated panel workfile type ("Undated Panels" on page 285).

In some cases, however, this behavior is not desired. If EViews reports that your date IDs are not unique, you might choose to go back and either modify or correct the original ID values, or specify an alternate frequency in which the identifiers are unique. For example, the date string identifier values "1/1/2002" and "2/1/2002" are not unique in a quarterly workfile, but are unique in a monthly workfile.

Invalid date identifiers

When defining dated workfile structures, EViews requires that you enter the name or names of series containing date information. This date information may be in the form of valid EViews date values, or it may be provided in numbers or strings which EViews will attempt to interpret as valid date values. In the latter case, EViews will attempt to create a new series containing the interpreted date values.

If EViews is unable to translate your date information into date values, it will issue an error indicating that the date series has invalid values or that it is unable to interpret your date specification. You must either edit your date series, or structure your workfile as an undated workfile with an ID series.

In cases where your date information is valid, but contains values that correspond to unlikely dates, EViews will inform you of this fact and prompt you to continue. Suppose, for example, that you have a series that contains 4-digit year identifiers ("1981," "1982," *etc.*), but also has one value that is coded as a 2-digit year ("80"). If you attempt to use this series as your date series, EViews will warn you that it appears to be an integer series and will ask you if you wish to recode the data as integer dates. If your proceed, EViews will *alter the values in your series* and create an *integer dated* (*i.e.*, not time dated) workfile, which may not be what you anticipated.

Alternately, you may cancel the restructure procedure, edit your date info series so that it contains valid values, and reattempt to apply a structure.

Missing value identifiers

Your identifier series may be numeric or alpha series containing missing values. How EViews handles these missing values depends on whether the series is used as a date ID series, or as an observation or group ID series.

Missing values are not allowed in situations where EViews expects date information. If EViews encounters missing values in a date ID series, it will issue a warning and will prompt you to delete the corresponding observations. If you proceed, EViews will remove the observations from the workfile. If removed, the observations may not be recovered, even if you subsequently change or remove the workfile structure.

If the missing values are observed in an observation or group ID series, EViews will offer you a choice of whether to keep or remove the corresponding observations, or whether to cancel the restructure. If you choose to keep the observations, the missing value, NA, for numeric series, and a blank string for alpha series, will be used as an observation or cross-section ID in the restructured workfile. If you choose to drop the observations, EViews will simply remove them from the workfile. These observations may not be recovered.

Removing a Workfile Structure

You may remove a workfile structure at any time by restructuring to an unstructured or regular frequency dated workfile. Call up the **Workfile structure** dialog and select **Unstructured/Undated** or **Dated - regular frequency** from the dropdown menu. Fill out the appropriate entries and click **OK**.

EViews will remove the workfile structure and will unlock any series used as date, group, or observation identifiers.

Resizing a Workfile

Resizing a workfile page is a special case of restructuring. Simply call up the **Workfile structure** dialog for any workfile page by selecting **Proc/"Structure/Resize Current Page…"** from a workfile window, or by clicking on the "Range:" description header near the top of the main workfile window. EViews will open the workfile structure dialog with your current settings displayed in the appropriate fields.

Dated - regular frequency / Unstructured

For workfile types where the structure of the data is described explicitly (dated with regular frequency, or unstructured), the **Start date** and **End date**, or **Observations** values will be filled out with actual values.

To change the sizes of regular frequency workfiles, enter the appropriate **Start date** and **End date** information using explicit dates or offsets from "@FIRST" and "@LAST".

To change the size of an unstructured workfile, change the number of **Observations**. Note that for unstructured data, you may only add or delete observations from the end of the workfile, you may not change the starting observation; if you wish to modify the starting observation you will need to work with an integer dated workfile.

EViews will inform you of the number of observations to be added and/or deleted, and will prompt you to continue.

For example, changing the **End date** for your annual workfile from "2001" to "2009", or the number of **Observations** in your unstructured workfile from "100" to "107" will both add 7 observations to the end of the respective workfiles. Likewise, changing the **Start date** of your monthly workfile from "1990:01" to "@FIRST-24" will add 24 months to the beginning of the workfile while changing the **End date** to "@LAST-3" removes (deletes) the last three observations.
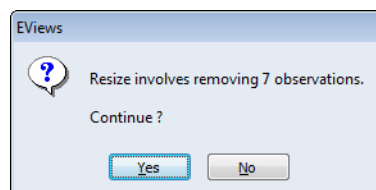
Dated - specified by date series

For a dated workfile that is structured using a date series, the dialog will open with pre-filled **Start date** and **End date** values containing "@FIRST" and "@LAST" as stand-ins for the earliest and latest observed dates. To change the size of a dated workfile structured by a date series, simply enter the appropriate information using explicit dates or offsets from "@FIRST" and "@LAST".

Given your start and end date values, EViews will analyze your date identifiers to determine whether you need to add or remove observations. If required, EViews will inform you of the number of observations to be added or deleted, and you will be prompted to continue. If observations are added, the date series will be modified to hold the corresponding date values. As with other forms of restructuring, deleted observations may not be recovered.

An observation will be deleted if the corresponding date ID falls outside the range implied by the start and end dates. If we enter "1970" as the **Start date** and "2010" as the **End date** in our annual workfile, any observations whose date series value is earlier than 1970 or later than 2010 will be removed from the workfile. If we enter "@FIRST + 2" and "@LAST-3" as our **Start date** and **End date**, EViews will delete the first two and last three observations from the workfile.

EViews will add observations to the workfile if the **Start date** is earlier than "@FIRST" or the **End date** is later than "@LAST". The observations to be added are determined by examining the regular frequency calendar to find all possible dates which fall in the desired range. If, in our annual workfile that ranges from 1980 to 2000, we specify a **Start date** of "1975", EViews will add observations for all of the years from 1975 to 1979, and will modify the date series so that it contains the associated date values. Alternatively, entering "@FIRST–2" and

"@LAST + 2" adds two observations corresponding to 1978 and 1979, and two observations corresponding to 2001 and 2002.

Note that there is a bit of asymmetry here in the use of offsets to "@FIRST" and "@LAST". Offsets that remove observations from the workfile simply count from the first or last observation, while offsets that add observations to the workfile use the regular frequency calendar to determine the dates to be added.

Dated Panel

For dated panel workfiles, the prefilled **Start date** and **End date** values will contain "@FIRST" and "@LAST" as stand-ins for the cross-section specific earliest and latest observed dates. To resize a dated panel workfile, you may enter an explicit date value in one or both of those fields. If you elect to use offsets, you must take care to understand the inherent complexities involved.

When you enter "@FIRST + 2" and "@LAST–2", EViews trims off 2 *observations* from the beginning and end of each cross-section. Used in this fashion, "@FIRST" refers to the earliest date for each cross-section, and the offsets are in observation space.

If we combine this trimming with balancing starts or ends, balancing occurs prior to the trimming of observations. Interestingly, this means that the starts or ends will not necessarily be balanced following trimming.

In order to use "@FIRST–2" or "@LAST + 2", EViews must balance starts or ends. The interpretation of the offsets that extend beyond the range of observations differs since they are evaluated in regular date space. If you enter "@FIRST–2" and choose to balance starts, the behavior is: first balance starts, then add two observations to the beginning in date space. Note that this operation is the same as adding two observations in regular date space to the cross-section with the earliest observed date and then balancing starts.

This behavior means that you cannot easily add two observations (in date space) to the start or end of each cross-section, without possibly adding more via start or end balancing. The panel data will have balanced starts or ends following the operation.

Undated with ID series / Undated Panel

Resizing an undated workfile that is structured using an ID series requires several distinct operations, since there is no simple way to describe the restructure operation. At a deep level, resizing these types of workfiles involves modifying your identifiers, and then adding or deleting observations with specific identifier values.

To alter the identifier series you must first remove the workfile structure. Call up the **Workfile structure** dialog and select **Unstructured/Undated** from the dropdown menu. Click on **OK**. EViews will remove the existing workfile structure and will unlock the ID series.

If you wish to remove observations, you should edit one of the ID series so that the desired observations have missing IDs. If you reapply the original **Undated with ID series** or **Undated Panel** structure, EViews will prompt you to remove observations with the missing ID values. We remind you that this step will remove all observations with missing values for the identifiers; if you originally used the missing value as a valid identifier, the corresponding observation will also be removed.

To add observations, you must first append observations to the workfile by expanding the unstructured workfile and then editing the ID series to add unique identifiers for the new values, or by using the built-in tools to append to the workfile page described below. Once you have added the new observations, you may reapply the workfile structure. EViews will sort your data using the identifier values, lock down the ID series, and then apply the structure to the expanded workfile.

Appending to a Workfile

One method of combining two workfile pages is to append observations from a source workfile page to the end of a target workfile page. When appending data, EViews first removes any structure from the target page, then expands its range to encompass the combined range of the original page and the appended data. The data from the source page are then copied to the expanded part of the target workfile range, either in existing series or alpha objects, or in newly created objects.

When appending, you should first make certain that the workfiles containing both the source and target page are open in EViews. In some cases (for example, concatenating a workfile page with itself), you only need to have a single open workfile since the source and target workfiles are the same.

To open the **Workfile Append** dialog, click on the **Proc** button on the target workfile toolbar and select **Append to Current Page...**, or select **Proc/Append to Current Page...** from the main menu.

Selecting Data to Append

You should begin by selecting a workfile page containing data to be appended to the target page. The first dropdown menu contains a list of all workfiles currently in memory from which you should select the source workfile; in the second dropdown menu, you should choose a page from those in the workfile you have selected. Here, we have instructed EViews to append data from the ANNUAL page in the workfile LINK_SALES.
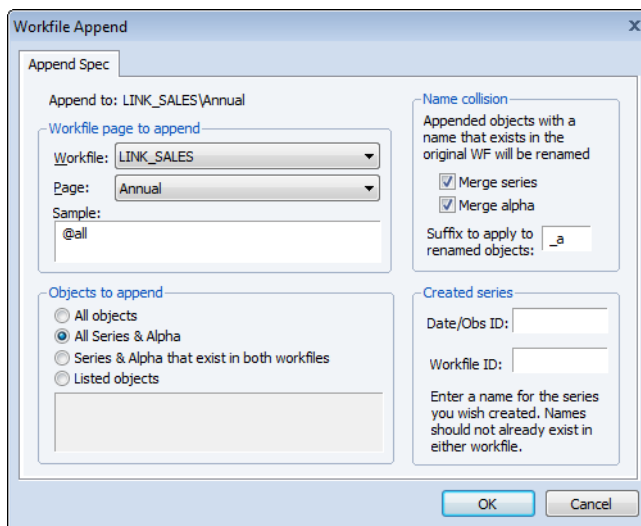
Next, you should specify a sample of observations in the source page to be appended; any valid EViews sample may be provided. Here, we have specified the default sample "@ALL", which ensures that we use all of the observations in the source page.

If you wish, you may use the **Objects to append** settings to specify the objects to be appended or copied. By default (**All series & alpha**), EViews will append all series and alphas (and links) from the source page into the destination page. If you select **All objects,** EViews will append all series and alphas, and will copy all other objects into the destination. Alternatively, choosing **Listed objects** allows you to specify the individual objects to be copied by name, using wildcards if convenient. To append only those data objects that exist in both pages, you should select **Series & alpha that exist in both workfiles**. If this setting is selected, a series or numeric link Y in the source page will only be appended if a series Y exists in the active page, and an alpha or alpha link X in the source will only be appended if an alpha series X exists in the destination.

Handling Name Collision

The settings in **Name collision** control the method EViews uses to append data when a source object name is present in the target page. To understand the effects of the various settings, consider the three possible scenarios that may occur when appending from an object into a workfile page:

- there is no object with the same name in the target page.

- an object with the same name exists in the target, but the object type is not compatible.

- an object with the same name exists in the target, and the object type is compatible with the source object.

In saying that the source and destination objects are compatible, we indicate that the source data may be added to the end of the existing object. Series and numeric link data may only be added to the end of series objects, while alpha and alpha link data may only be added to the end of alpha objects. All other combinations of objects are said to be incompatible.

Suppose that we wish to append the source series X or numeric link to the target page. If there is no object with the same name in the target page, EViews will create a new series, X, containing NA values for the original target page observations, and the values of the source series X for observations in the expanded part of the range.

If there is an incompatible matching object, a new object will be created with a name formed from the original name and the text specified in the **Suffix to apply to renamed objects** edit field. If, for example, the target page contains an incompatible X (*e.g.*, it contains the equation X), EViews will create a new series using the original name, and the specified suffix, for example, "X_A" (using the default suffix, "_A").

If there is a compatible matching object, EViews will examine your dialog settings to determine the appropriate behavior. By default, EViews will append the data from a compatible source object to the end of the existing object. Thus, data from the series or numeric link X will be copied to the expanded part of the range of the target series X, and data from the alpha or alpha link Y will be copied to the end of the alpha series Y. You may override this default so that EViews creates a new object even when the matching objects are compatible, by unselecting the **Merge series** or **Merge alpha** checkboxes.

Creating Identifier Series

The optional **Created series** settings in the dialog allow you to save series containing information about each observation in the combined workfile.

To save a series containing the date or observation ID associated with each observation in the combined workfile, you should enter a unique name in the edit field labeled **Date/Obs ID**. The specified series will be created in the target page, and will contain the observation or cell identifiers given by the structures associated with the source and the original target pages. Saving the IDs is particularly useful since appending to a workfile removes the existing page structure.

The optional **Workfile ID** series identifies the source of the observation in the combined workfile: observations in the original target page are assigned the value 0, while observations in the appended portion of the target will be given the value 1.

Contracting a Workfile

Samples are an important tool for restricting the set of observations used by EViews when performing calculations. You may, for example, set an estimation sample to restrict the observations used in a regression to only include females, or to only use the observations with dates between 1990 and 2003. An important advantage to working with samples is that the exclusion of observations is temporary, and may be reset simply by providing a new sample specification. Note also that even as they are excluded from calculations, out-of-sample observations still exist, and are used for lag processing.

There may be times, however, when you wish to drop or remove observations from a workfile page. For example, if you have daily data on stock trades and want lags to skip holidays, you must permanently remove holidays from the workfile. Similarly, if the focus of your analysis is on female labor force participation, you may wish to subset your workfile by excluding all males. Contracting the workfile in this fashion both reduces the size of the workfile and makes it easier to work with, since you no longer have to remember to set all samples to exclude males.

To contract a workfile page in place, you should click on the **Proc** button on the workfile toolbar and select **Contract Current Page...,** or select **Proc/Contract Current Page...** from the main menu.

EViews will open the **Workfile Contract** dialog prompting you to input a valid sample specification. Simply enter a sample specification and EViews will drop all observations in the current page that do not meet the specified criteria. Here, we drop all observations where the ID series is greater than 7 or where K lies between 100 and 200 (inclusive).
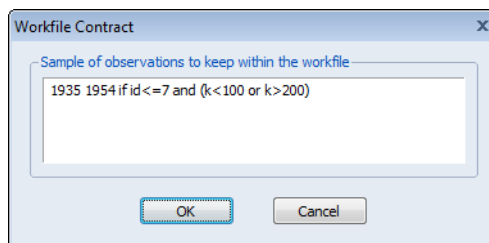
We emphasize that the workfile contraction occurs in place so that the existing workfile page will no longer exist. If you wish to keep the original page, you should make a copy of the page, or save it to disk.

Copying from a Workfile

EViews provides you with convenient tools for copying or extracting subsamples of observations and series objects from existing workfiles and creating new pages containing the extracted data or links to the data. You may, for example, wish to create separate workfile pages for the males and females in your cross-section workfiles, or to keep only non-holiday dates from your regular frequency daily-7 data. Similarly, you may wish to create a page containing a small subset of the series found in your original workfile.

Copying or extracting the series object data may be performed in two distinct ways: by creating links in a new page in the same workfile, or by copying the series objects into a new page in the existing or an alternate workfile.

The first method uses link objects to create memory efficiently, dynamically updating copies of the data in your series, link, and alpha objects, but requires that the new destination page be in the same workfile.

The second method copies the actual values in the objects. Since links are not involved, you may use this approach to copy data into new pages in different workfiles. In addition, when copying by value, you may copy other types of EViews objects and you will have access to built-in tools for creating random samples of the observations in the source workfile.

Copying by Link

To copy all or part of the data in a workfile by creating links, you should select **Proc/"Copy Extract from Current Page"/By Link to New Page...**. EViews will open the **Workfile Copy By Link** dialog in which you will specify the data to be copied.

There are two principal ways that you can specify a subset of the data to be copied: you may specify a subsample of observations in the workfile or you may specify a subset of the series objects.

EViews will copy all of the observations in the sample specified in the edit box labeled **Sample - observations to copy**. To specify a subsample of observations, you should replace the default "@ALL" with a valid EViews sample.

You may elect to copy all series, alphas, and valmaps, or you may select the **Listed Series - Alphas - Valmaps** radio button and enter a list of the series to be copied, with wildcards, if desired.

If the **Include Links** checkbox is selected, EViews will copy series and alpha links along with ordinary series and alphas. If you uncheck **Include Links**, EViews will drop all link objects from the copy list.
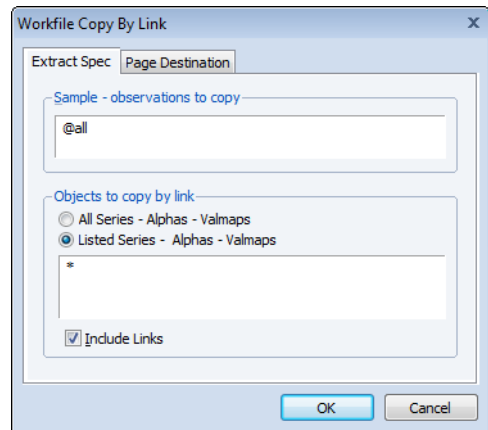
The copy by link procedure will create the links in a new page in the existing workfile. By default, the page will be given a name based on the page structure (e.g., "Annual", or "Daily5"). You may provide a name for this destination page by clicking on the **Page Destination** tab and enter the desired name. If a page with that name already exists in the workfile, EViews will create a new page using the next available name. Note that since we are copying by link, you may not create a page in a different workfile.
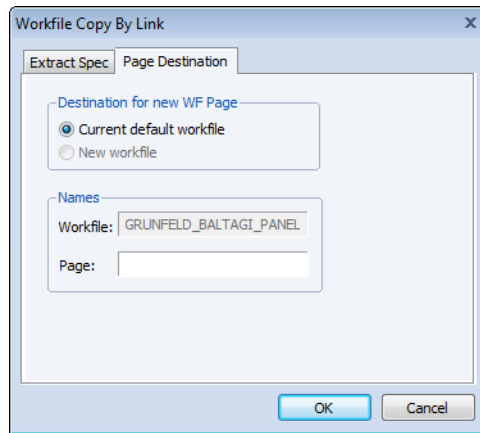
When you click on **OK** to accept the dialog settings, EViews first examines your source workfile and the specified sample, and then creates a new page with the appropriate number of observations.

Next, EViews will copy, by value, the ID series used to structure the source workfile page for the specified sample of observations. Using the new series, EViews will structure the new workfile in a manner similar to the source workfile page. If, for example, you have an undated workfile that is structured using an ID series COUNTRIES, EViews will create a series in the destination page, copy the relevant values, and structure the page as an undated workfile using the new ID series COUNTRIES. Similarly, if the original page has an annual panel structure that is defined using multiple ID series, all of the ID series will be copied to the new page, and the page will be structured as an annual panel using these new series.

Lastly, EViews will create links in the new page for all of the specified series objects. The links will be defined as general match merge links using the source and destination ID series. Since the new page is a subset of the original page, the contraction methods will be set to **No contractions allowed** (see "Link calculation settings" on page 249).

Copying by Value

To copy all or part of the workfile by value, you should select **Proc/"Copy/Extract from Current Page"/By Value to New Page or Workfile...**. EViews will open the **Workfile Copy By Value** dialog.

You should first specify an EViews sample describing the observations to be copied. By default, EViews will use the sample "@ALL".

Next, you should use the dropdown menu to select a **Random subsample** method. By default, all of the observations in the sample will be used (**No random subsampling**), but you may choose to extract a random sample in one of three ways:

- You may extract a subsample with a fixed number of observations (**Fixed subsample size - number of observations**). If the specified subsample size is larger than the number of observations, the entire sample is used.

- You may select a subsample with a fixed size, where the number of observations is specified as a percent of the total number of observations (**Fixed subsample size - % of observations**).

- You may take a simple random sample in which every observation has a fixed probability of being selected (**Random subsample size - % applied to each obs**). As the label suggests, the number of observations in the resulting subsample is itself random.

In the remainder of the dialog page you should specify the objects to be copied. There are two parts to the object specification: a list of object names, and a set of modifiers for object types.

By default, the **All objects** radio button is selected so that the list of object names provided to EViews will include every object in the source workfile. You may instead provide an explicit list by clicking on the **Listed objects** radio button and entering the names of objects (using wildcards if appropriate).

The type matching checkboxes (**Series - Alphas - Valmaps**, **Links**, **Estimation & Model Objects**, **All others**) may be used to restrict the object list on the basis of broad classifications for type; an object will be copied only if it is in the list of object names provided in the edit box, and if its type matches a classification that you elect to copy. If, for example, you wish to remove all objects that are not series objects or valmaps from your list, you should uncheck the **Estimation & Model objects** and the **All others** checkboxes.
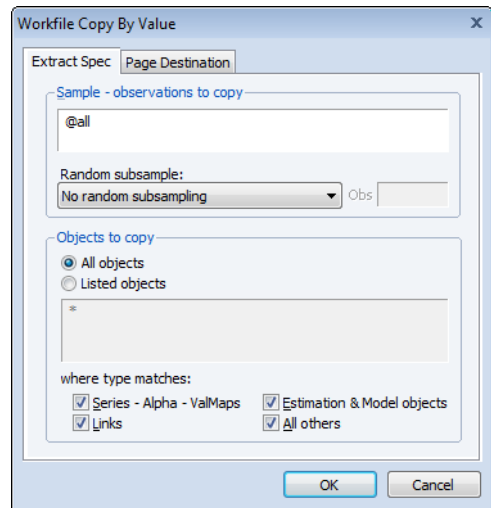
Lastly, you may optionally provide a destination workfile page. By default, EViews will copy the data to a new workfile in a page named after the workfile page structure (*e.g.*, "Quarterly," "Monthly"). You may provide an alternative destination by clicking on the **Page Destination** tab in the dialog, and entering the desired destination workfile and/or page.

When you click on **OK,** EViews examines your source workfile and the specified sample, and creates a new page with the appropriate number of observations. EViews then copies the ID series used to structure the source workfile, and structures the new workfile in identical fashion. Lastly, the specified objects are copied to the new workfile page.

Reshaping a Workfile

In a typical study, each subject (individual, firm, period, *etc.*) is observed only once. In these cases, each observation corresponds to a different subject, and each series, alpha, or link in the workfile represents a distinct variable.

In contrast, *repeated measures data* may arise when the same subject is observed at different times or under different settings. The term *repeated measures* comes from the fact that for a given subject we may have repeated values, or measures, for some variables. For example, in longitudinal surveys, subjects may be asked about their economic status on an annual basis over a period of several years. Similarly, in clinical drug trials, individual patient health may be observed after several treatment events.

It is worth noting that standard time series data may be viewed as a special case of repeated measures data, in which there are repeated higher frequency observations for each lower frequency observation. Quarterly data may, for example, be viewed as data in which there are four repeated values for each annual observation. While time series data are not typically viewed in this context, the interpretation suggests that the reshaping tools described in this section are generally applicable to time series data.

There are two basic ways that repeated measures data may be organized in an EViews workfile. To illustrate the different formats, we consider a couple of simple examples.

Suppose that we have the following dataset:

ID1	ID2	Sales
1	Jason	17
1	Adam	8
2	Jason	30
2	Adam	12
3	Jason	20

We may view these data as representing repeated measures on subjects with identifiers given in ID1, or as repeated measures for subjects with names provided in ID2. There are, for example, two repeated values for subjects with "ID1 = 1", and three repeated values for SALES for Jason. Note that in either case, the repeated values for the single series SALES are represented in multiple observations.

We can rearrange the layout of the data into an equivalent form where the values of ID2 are used to break SALES into multiple series (one for each distinct value of ID2):

ID1	SalesJason	SalesAdam
1	17	8
2	30	12
3	20	NA

The series ID2 no longer exists as a distinct series in the new format, but instead appears implicitly in the names associated with the new series (SALESJASON and SALESADAM). The repeated values for SALES are no longer represented by multiple observations, but are instead represented in the multiple series values associated with each value of ID1.

Note also that this representation of the data requires that we add an additional observation corresponding to the case ID1 = 3, ID2 = "Adam". Since the observation did not exist in the original representation, the corresponding value of SALESADAM is set to NA.

Alternatively, we may rearrange the data using the values in ID1 to break SALES into multiple series:

ID2	Sales1	Sales2	Sales3
Jason	17	30	20
Adam	8	12	NA

In this format, the series ID1 no longer exists as a distinct series, but appears implicitly in the series names for SALES1, SALES2, and SALES3. Once again, the repeated responses for SALES are not represented by multiple observations, but are instead held in multiple series.

The original data format is often referred to as *repeated observations* format, since multiple observations are used to represent the SALES data for an individual ID1 or ID2 value. The latter two representations are said to be in repeated variable or *multivariate* form since they employ multiple series to represent the SALES data.

When data are rearranged so that a single series in the original workfile is broken into multiple series in a new workfile, we term the operation *unstacking the workfile*. Unstacking a workfile converts data from repeated observations to multivariate format.

When data are rearranged so that sets of two or more series in the original workfile are combined to form a single series in a new workfile, we call the operation *stacking the workfile.* Stacking a workfile converts data from multivariate to repeated observations format.

In a time series context, we may have the data in the standard stacked format:

Date	Year	Quarter	Z
2000Q1	2000	1	2.1
2000Q2	2000	2	3.2
2000Q3	2000	3	5.7
2000Q4	2000	4	6.3
2001Q1	2001	1	7.4
2001Q2	2001	2	8.1
2001Q3	2001	3	8.8
2001Q4	2001	4	9.2

where we have added the columns labeled YEAR and QUARTER so that you may more readily see the repeated measures interpretation of the data.

We may rearrange the time series data so that it is unstacked by QUARTER,

Year	Z1	Z2	Z3	Z4
2000	2.1	3.2	5.7	6.3
2001	7.4	8.1	8.8	9.2

or in the alternative form where it is unstacked by YEAR:

Quarter	Z2000	Z2001
1	2.1	7.4
2	3.2	8.1
3	5.7	8.8
4	6.3	9.2

EViews provides you with convenient tools for reshaping workfiles between these different formats. These tools make it easy to prepare a workfile page that is set up for use with built-in pool or panel data features, or to convert data held in one time series representation into an alternative format.

Unstacking a Workfile

Unstacking a workfile involves taking series objects in a workfile page, and in a new workfile, breaking the original series into multiple series.

We employ an *unstacking ID series* in the original workfile to determine the destination series, and an *observation ID series* to determine the destination observation, for every observation in the original workfile. Accordingly, we say that a workfile is "unstacked by" the values of the unstacking ID series.

To ensure that each series observation in the new workfile contains no more than one observation from the existing workfile, we require that the unstacking ID and the observation ID are chosen such that no two observations in the original workfile have the same set of values for the identifier series. In other words, the identifier series must together uniquely identify observations in the original workfile.

While you may use any series in the workfile as your unstacking and observation identifier series, an obvious choice for the identifiers will come from the set of series used to structure the workfile (if available). In a dated panel, for example, the cross-section ID and date ID series uniquely identify the rows of the workfile. We may then choose either of these series as the unstacking ID, and the other as the observation ID.

If we unstack the data by the cross-section ID, we end up with a simple dated workfile with each existing series split into separate series, each corresponding to a distinct cross-section ID value. This is the workfile structure used by the EViews pool object, and is commonly used when the number of cross-sectional units is small. Accordingly, one important application of unstacking a workfile involves taking a page with a panel structure and creating a new page suitable for use with EViews pool objects.

On the other hand, if we unstack the panel workfile by date (using the date ID series or @DATE), we end up with a workfile where each row represents a cross-sectional unit, and each original series is split into separate series, one for each observed time period. This format is frequently used in the traditional repeated measures setting where a small number of variables in a cross-sectional dataset have been observed at different times.

To this point, we have described the unstacking of panel data. Even if your workfile is structured using a single identifier series, however, it may be possible to unstack the workfile by first splitting the single identifier into two parts, and using the two parts as the identifier series. For example, consider the simple quarterly data given by:

Date	X	Y
2000Q1	NA	-2.3
2000Q2	5.6	-2.3

2000Q3	8.7	-2.3
2000Q4	9.6	-2.3
2001Q1	12.1	1.6
2001Q2	8.6	1.6
2001Q3	14.1	1.6
2001Q4	15.2	1.6

Suppose we wish to unstack the X series. We may split the date identifier into a year compo-nent and a quarter component (using, say, the EViews @YEAR and @QUARTER functions). If we extract the QUARTER and YEAR from the date and use the QUARTER as the unstack-ing identifier, and the YEAR as the observation identifier, we obtain the unstacked data:

Year	X1	X2	X3	X4
2000	NA	5.6	8.7	9.6
2001	12.1	8.6	14.1	15.2

Note that we have chosen to form the series names by concatenating the name of the X series, and the values of the QUARTER series.

Alternatively, if we use YEAR as the unstacking ID, and QUARTER as the observation ID, we have:

Quarter	X2000	X2001
1	NA	12.1
2	5.6	8.6
3	8.7	14.1
4	9.6	15.2

In some cases, a series in the original workfile will not vary by the unstacking ID. In our example, we have a series Y that is only updated once a year. Stacking by QUARTER yields:

Year	Y1	Y2	Y3	Y4
2000	-2.3	-2.3	-2.3	-2.3
2001	1.6	1.6	1.6	1.6

Since there is no change in the observations across quarters, these data may be written as:

Year	Y
2000	-2.3
2001	1.6

without loss of information. When unstacking, EViews will automatically avoid splitting any series which does not vary across different values of the unstacking ID. Thus, if you ask EViews to unstack the original Y by QUARTER, only the compacted (single series) form will be saved. Note that unstacking by YEAR will *not* produce a compacted format since Y is not constant across values of YEAR for a given value of QUARTER.

Unstacking a Workfile in EViews

To unstack the active workfile page, you should select **Proc/Reshape Current Page/ Unstack in New Page...** from the main workfile menu. EViews will respond by opening the tabbed **Workfile Unstack** dialog.

When unstacking data, there are four key pieces of information that should be provided: a series object that contains the unstacking IDs, a series object that contains the observation IDs, the series in the source workfile that you wish to unstack, and a rule for defining names for the unstacked series.

Unstacking Identifiers

To unstack data contained in a workfile page, your source page must contain a series object containing the unstacking identifiers associated with each observation. For example, you may have an alpha series containing country abbreviations ("US," "JPN," "UK"), or individual names ("Joe Smith," "Jane Doe"), or a numeric series with integer identifiers ("1," "2," "3," "50," "100," ...). Typically, there will be repeated observations for each of the unique unstacking ID values.

You should provide the name of your unstacking ID series object in the top edit field of the dialog. When unstacking, EViews will create a separate series for each distinct value of the ID series, with each of these series containing the multiple observations associated with that value. The series used as the unstacking ID is always dropped from the destination workfile since its values are redundant since they are built into the multiple series names.

If you wish to unstack using values in more than one series, you must create a new series that combines the two identifiers by identifying the subgroups, or you may simply repeat the unstacking operation.

Observation Identifiers

Next, you must specify a series object containing an observation ID series in the second edit field. The values of this series are used to identify both the individual observations in the unstacked series and the structure of the destination page.

Once again, if your workfile is structured, an obvious choice for the unstacking identifier series comes from the series used to structure the workfile, either directly (the date or cross-section ID in a panel page), or indirectly (the YEAR or QUARTER extracted from a quarterly date).

EViews will, if necessary, create a new observation ID series in the unstacked page with the same name as, and containing the unique values of, the original observation ID series. This series will be used to structure the workfile.

If the original observation ID is an ordinary series or alpha, the new page will be structured as a cross-section page using the new identifier series. Alternatively, if the observation ID is a date series or the "@DATE" keyword, EViews will analyze the observed date values and will create a dated page with the appropriate frequency.

Series to be Unstacked

You may enter the names of the series, alphas, and links that you wish to unstack in the edit field **Series to be unstacked into new workfile page.** You may enter the names directly, or use expressions containing wildcards. For example, the expression "SALES A*" instructs EViews to unstack both the SALES series as well as all series objects beginning with the letter "A".

Note that the RESID series and the unstacking ID series may not be unstacked.

Naming Unstacked Series

EViews will use the pattern in the **Name pattern for unstacked series** field to construct the names for the new unstacked series or alphas associated with each stacked series object.

By default, the wildcard pattern "*?" will be used, meaning that unstacked series names will be constructed by concatenating the name of the series object to be unstacked and a string containing one of the unique values found in the unstacking ID series.

In our example above, when unstacking the SALES series using NAME as the unstacking ID series and the wildcard name pattern "*?", EViews will create the series JASONSALES and ADAMSALES. If instead, we enter the pattern "?_*", EViews will put the unstacked values in the series SALES_JASON and SALES_ADAM.

Unstacking Destination

By default, EViews will unstack the data in a new UNTITLED page in the existing workfile. You may provide an alternative destination by clicking on the **Page Destination** tab in the dialog, and entering the desired destination.

An Example

Consider a workfile that contains the series GDP and CONS, which contain the values of Gross Domestic Product and consumption for three countries stacked on top of each other.

Suppose further that there is an alpha object called COUNTRY containing the values "US," "UK," and "JPN", which identify the country associated with each observation on GDP and CONS. Finally, suppose there is a date series DATEID which identifies the date associated with each observation in the page. COUNTRY and DATEID uniquely determine the observation identifiers.

In our example, we assume that the source page contains annual data from 1991 to 2000 for the three countries in our panel. We can better see this structure by opening a group window showing the values of COUNTRY, DATEID (displayed in year-date format), and GDP.

We wish to unstack the data in GDP and CONS using the unstacking ID values in COUNTRY, and the observation IDs in DATEID. Click on **Proc/Reshape Current Page/Unstack in New Page...** in the workfile window to bring up the unstacking dialog.

Enter "COUNTRY" as the unstacking ID series, and "DATEID" for the observation identifier. We leave the remainder of the dialog settings at the default values, so that EViews will use "*?" as the name pattern, will copy all series objects in the page (with the exception of RESID and COUNTRY), and will place the results in a new page in the same workfile.

If you click on **OK** to accept the settings, EViews will first examine the DATEID series to determine the number of unique observation identifiers. Note that the number of unique observation identifier values determines the number of observations in the unstacked workfile. Next, EViews will determine the number of unique values in COUNTRY, which is equal to the number of unstacked series created for each stacked series.
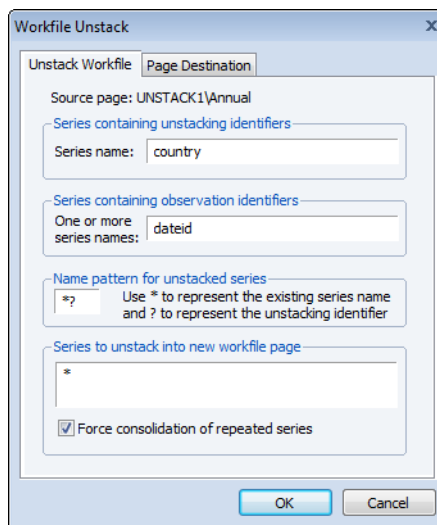
In this example, we start with a balanced panel with 10 distinct values for DATEID, and three distinct values in COUNTRY. The resulting UNTITLED workfile page will follow an annual frequency from the 10 observations from 1991 to 2000, and will have three unstacked series corresponding to each of the source series. The names of these series will be formed by taking the original series name and appending the distinct values in COUNTRY ("US," "UK," and "JPN").
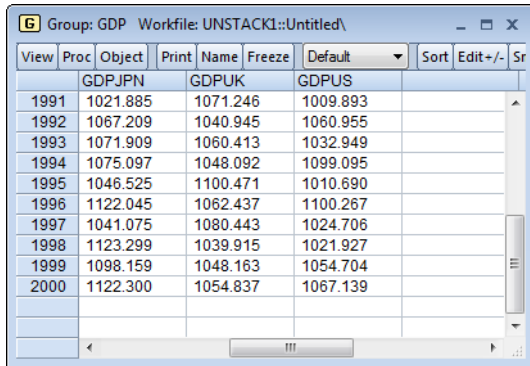
Note that in addition to the six unstacked series CONSJPN, CONSUK, CONSUS, GDPJPN, GDPUK, GDPUS, EViews has created four additional objects. First, the unstacked page contains two group objects taking the name of, and corresponding to, the original series CONS and GDP.

Each group contains all of the unstacked series, providing you with easy access to all of the series associated with the original stacked series. For example, the group GDP contains the three series, GDPJPN, GDPUK, and GDPUS, while CONS contains CONSJPN, CONSUK, and CONSUS.
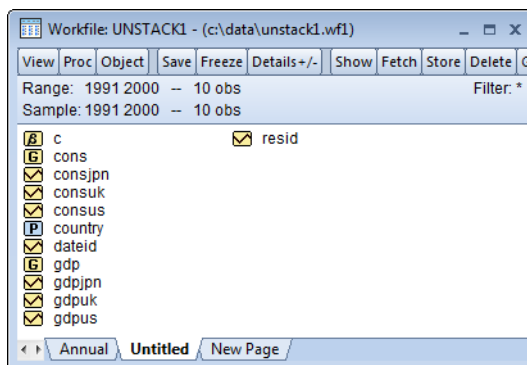
Opening the GDP group spreadsheet, we see the result of unstacking the original GDP series into three series: GDPJPN, GDPUK, and GDPUS. In particular, the values of the GDPJPN and GDPUK series should be compared with the values of GDP depicted in the group spreadsheet view of the stacked data.

Second, EViews has created a (date) series DATEID containing the distinct

values of the observation ID series. If necessary, this series will be used to structure the unstacked workfile.

Lastly, EViews has created a pool object named COUNTRY, corresponding to the specified unstack ID series, containing all of the unstacking identifiers. Since the unstacked series have names that were created using the specified name pattern, this pool object is perfectly set up for working with the unstacked data.

Stacking a Workfile

Stacking a workfile involves combining sets of series with related names into single series, or repeatedly stacking individual series into single series, and placing the results in a new workfile. The series in a given set to be stacked may be thought of as containing repeated measures data on a given variable. The individual series may be viewed as ordinary, non-repeated measures data.

The stacking operation depends crucially on the set of stacking identifiers. These identifiers are used to determine the sets of series, and the number of times to repeat the values of individual series.

In order for all of the series in a given set to be stacked, they must have names that contain a common component, or *base name*, and the names must differ systematically in containing an identifier. The identifiers can appear as a suffix, prefix, or even in the middle of the base name, but they must be used consistently across all series in each set.

Suppose, for example, we have a workfile containing the individual series Z, and the two groups of series: XUS, XUK and XJPN, and US_Y, UK_Y, and JPN_Y. Note that within each set of series, the identifiers "US," "UK," and "JPN" are used, and that they are used consistently within each set of series.

If we employ the set of three identifier values "US," "UK," and "JPN" to stack our workfile, EViews will stack the three series XUS, XUK, and XJPN on top of each other, and the series

US_Y, UK_Y, and JPN_Y on top of each other. Furthermore, the individual series Z will be stacked on top of itself three times so that there are three copies of the original data in the new series.

Stacking a Workfile in EViews

To stack the data in an existing workfile page, you should select **Proc/Reshape Current Page/Stack in New Page...** from the main workfile menu. EViews will respond by opening the tabbed **Workfile Stack** dialog.

There are two key pieces of information that you must provide in order to create a stacked workfile: the set of stack ID values, and the series that you wish to stack. This information should be provided in the two large edit fields. The remaining dialog settings involve options that allow you to modify the method used to stack the series and the destination of the stacked series.

Stacking Identifiers

There are three distinct methods that you may use to specify your stack ID values:

First, you may enter a space separated list containing the individual ID values (*e.g.*, "1 2 3", or "US UK JPN"). This is the most straightforward method, but can be cumbersome if you have a large list of values.

Second, you may enter the name of an existing pool object that contains the identifier values.

Lastly, you may instruct EViews to extract the ID values from a set of series representing repeated measures on some variable. To use this method, you should enter a series name pattern containing the base name and the "?" character in place of the IDs. EViews will use this expression to identify a set of series, and will extract the ID values from the series names. For example, if you enter "SALES?", EViews will identify all series in the workfile with names beginning with the string "SALES", and will form a list of identifiers from the remainder of the observed series names. In our example, we have the series SALES1, SALES2, and SALES3 in the workfile, so that the list of IDs will be "1", "2", and "3".
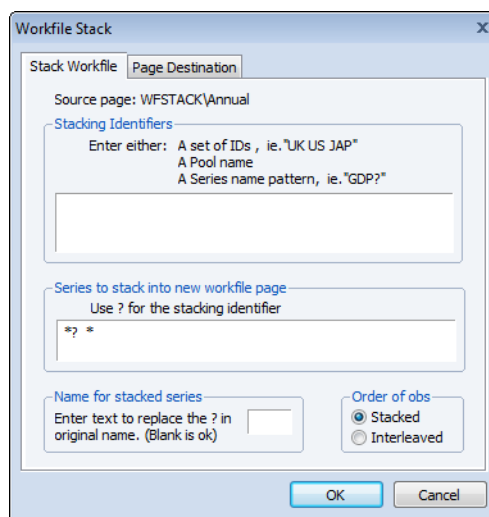
Series to be Stacked

Next, you should enter the list of series, alphas, and links that you wish to stack. Sets of series objects that are to be treated as repeated measures (stacked on top of each other) should be entered using "?" series name patterns, while individual series (those that should be repeatedly stacked on top of themselves), should be entered using simple names or wildcards.

You may specify the repeated measures series by listing individual stacked series with "?" patterns ("CONS? EARN?"), or you may use expressions containing the wildcard character "*" ("*?" and "?C*") to specify multiple sets of series. For example, entering the expression "?C* ?E*" will tell EViews to find all repeated measures series that begin with the letters "C" or "E" (*e.g.*, "CONS? CAP? EARN? EXPER?"), and then to stack (or interleave) the series using the list of stack ID values. If one of the series associated with a particular stack ID does not exist, the corresponding stacked values will be assigned the value NA.

Individual series may also be stacked. You may list the names of individual simple series (*e.g.*, "POP INC"), or you can specify your series using expressions containing the wildcard character "*" ("*", "*C", "F*"). The individual series will repeatedly be stacked (or interleaved), once for each ID value. If the target workfile page is in the same workfile, EViews will create a link in the new page; otherwise, the stacked series will contain repeated copies of the original values.

It should be noted that the wildcard values for individual series are processed *after* the repeated measures series are evaluated, so that a given series will only be used once. If a series is used as part of a repeated measures series, it will not be used when matching wildcards in the list of individual series to be stacked.

The default value "*? *" is suitable for settings where the repeated series have names formed by taking the base name and appending the stack ID values. The default will stack all repeated measures series, and all remaining individual series (except for RESID). Entering "*" alone will copy or link all series, but does not identify any repeated measures series.

Naming Stacked Series

Stacked individual series will be named in the destination page using the name of the series in the original workfile; stacked repeated measures series will, by default, be named using the base name. For example, if you stack the repeated measures series "SALES?" and the individual series GENDER, the corresponding stacked series will, by default, be named "SALES" and "GENDER", respectively.

This default rule will create naming problems when the base name of a repeated measures series is also the name of an individual series. Accordingly, EViews allows you to specify an alternative rule for naming your stacked repeated measures series in the **Name for stacked series** section of the dialog.

The default naming rule may be viewed as one in which we form names by replacing the "?" in the original specification with a blank space. To replace the "?" with a different string, you should enter the desired string in the edit field. For example, if you enter the string "_STK", then EViews will name the stacked series "CONS?" and "EARN?" as "CONS_STK" and "EARN_STK" in the destination workfile.

Stacking Order

EViews will, by default, create series in the new page by stacking series on top of one another. If we have identifiers "1", "2", and "3", and the series SALES1, SALES2, and SALES3, EViews will stack the entire series SALES1 followed by the entire series SALES2, followed by SALES3.

You may instruct EViews to interleave the data, by selecting the **Interleaved** radio button in the **Order of Obs** section of the dialog. If selected, EViews will stack the first observations for SALES1, SALES2, and SALES3, on top of the second observations, and so forth.

It is worth pointing out that stacking by series means that the observations contained in a given series will be kept together in the stacked form, while interleaving the data implies that the multiple values for a given original observation will be kept together. In some contexts, one form may be more natural than another.

In the case where we have time series data with different series representing different countries, stacking the data by series means that we have the complete time series for the "US" (USGDP), followed by the time series for the "UK" (UKGDP), and then "JPN" (JPNGDP). This representation is more natural for time series analysis than interleaving so that the observations for the first year are followed by the observations for the second year, and so forth.

Alternatively, where the series represent repeated measures for a given subject, stacking the data by series arranges the data so that all of the first measures are followed by all of the second measures, and so on. In this case, it may be more natural to interleave the data, so that all of the observations for the first individual are followed by all of the observations for the second individual, and so forth.

One interesting case where interleaving may be desirable is when we have data which has been split by period, within the year. For example, we may have four quarters of data for each year:

Year	XQ1	XQ2	XQ3	XQ4
2000	NA	5.6	8.7	9.6
2001	12.1	8.6	14.1	15.2

If we stack the series using the identifier list "Q1 Q2 Q3 Q4", we get the data:

Year	ID01	X
2000	Q1	NA
2001	Q1	12.1
2000	Q2	5.6
2001	Q2	8.6
2000	Q3	8.7
2001	Q3	14.1
2000	Q4	9.6
2001	Q4	15.2

which is not ordered in the traditional time series format from earliest to latest. If instead, we stack by "Q1 Q2 Q3 Q4" but interleave, we obtain the standard format:

Year	ID01	X
2000	Q1	NA
2000	Q2	5.6
2000	Q3	8.7
2000	Q4	9.6
2001	Q1	12.1
2001	Q2	8.6
2001	Q3	14.1
2001	Q4	15.2

Note that since interleaving changes only the order of the observations in the workfile and not the structure, we can always sort or restructure the workfile at a later date to achieve the same effect.

Stacking Destination

By default, EViews will stack the data in a new page in the existing workfile named "UNTITLED" (or the next available name, "UNTITLED1," "UNTITLED2," *etc.*, if there are existing pages in the workfile with the same name).

You may provide an alternative destination for the stacked data by clicking on the **Page Destination** tab in the dialog, and entering the desired destination.

Here, we instruct EViews to put the stacked series in the workfile named STACKWF in the named page ANNUAL-PANEL. If a page with that name already exists in the workfile, EViews will create a new page using the next available name.

We note that if you are stacking individual series, there is an important consequence of specifying a different workfile as the destination for your stacked series. If the target page is in the same workfile as the original page, EViews will stack individual series by creating link objects in the new page. These link objects have the standard advantages of being memory efficient and dynamically updating. If, however, the target page is in a different workfile, it is not possible to use links, so the stacked series will contain repeated copies of the original individual series values.

An Example

Consider an annual (1971 to 2000) workfile, "Wfstack.WF1", that contains the six series: CONSUS, CONSUK, CONSJPN, and GDPUS, GDPUK, GDPJPN, along with the ordinary series CONSTVAL and WORLDGDP.

We wish to stack series in a new page using the stack IDs: "US," "UK," and "JPN".

Click on the **Proc** button and select **Reshape Current Page/Stack in new Page...**.

We may specify the stacked series list explicitly by entering "US UK JPN" in the first edit box, or we can instruct EViews to extract the identifiers from series names by entering "GDP?". Note that we cannot use "CONS?" in place of "GDP?" since the presence of the ordinary series CONSTVAL means that "CONS?" is associated with the wrong number of identifiers.
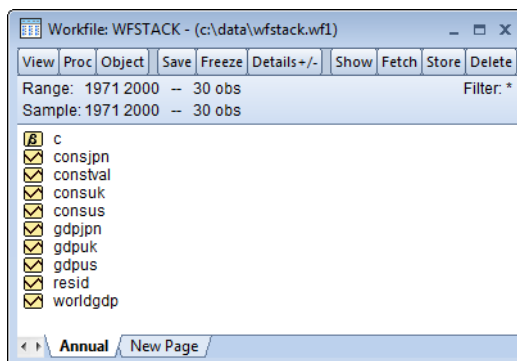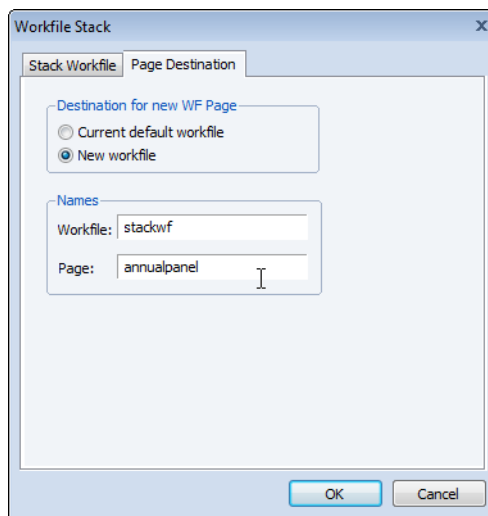
Assuming that we have entered "US UK JPN" or "GDP?" in the **Stacking Identifiers** edit box, we may then enter the expression

```
gdp? cons?
```

as our **Series to stack**. We leave the remainder of the dialog settings at their defaults, and click on **OK**.

EViews will first create a new page in the existing workfile and then will stack the GDPUS, GDPUK, and GDPJPN series and the CONSUS, CONSUK, and CONSJPN series. Since the dialog settings were retained at the default values, EViews will stack the data by series, with all of the values of GDPUS followed by the values of GDPUK and then the values GDPJPN, and will name the stacked series GDP and CONS.
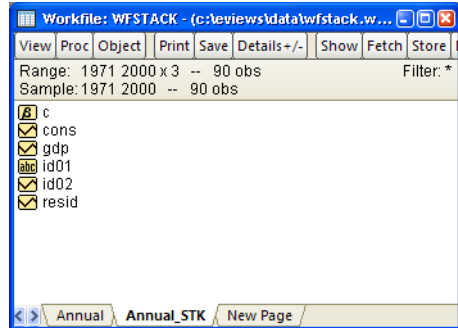
Here we see the resulting workfile page ANNUAL_STK, containing the stacked series GDP and CONS, as well as two EViews created series objects, ID01 and VAR01, that contain identifiers that may be used to structure the workfile.

VAR01 is an alpha series that contains the stack ID values "US," "UK," and "JPN" which are used as group identifiers, and ID01 is a data series containing the year observation identifiers (more generally, VAR01 will contain the values of the observation identifiers from the original page).

You may notice that EViews has already applied a panel structure to the page, so that there are three cross-sections of annual data from 1971 to 2000, for a total of 90 observations.

Note that EViews will only apply a panel structure to the new page if we stack the data by series, but not if we interleave observations. Had we chosen to interleave, we would obtain a new 90 observation unstructured page containing the series GDP and CONS and the alpha VAR01 and series ID01, with the observations for 1971 followed by observations for 1972, and so forth.

We may add our individual series to the stacked series list, either directly by entering their names, or using wildcard expressions. We may use either of the stack series expressions:

```
gdp? cons? worldgdp constval
```

or

```
gdp? cons? *
```

to stack the various "GDP?" and "CONS?" series on top of each other, and the individual series WORLDGDP and CONSTVAL will be linked to the new page so that the original series values are repeatedly stacked on top of themselves.

It is worth reminding you that the wildcard values for individual series are processed *after* the repeated measures series "GDP?" and "CONS?" are evaluated, so that a given series will only be used once. Thus, in the example above, the series CONSUS is used in forming the stacked CONS series, so that it is ignored when matching the individual series wildcard.

If we had instead entered the stacking list

```
gdp? *
```

EViews would stack the various "GDP?" series on top of each other, and would treat the remaining series as individual series so that the values of CONSUS, CONSUK, CONSJPN, CONSTVAL, and WORLDGDP are repeated in the stacked page. In this latter case, the wildcard implies that since the series CONSUS, CONSUK, and CONSJPN are not used in forming a stacked repeated measures series, they are used to create stacked individual series.
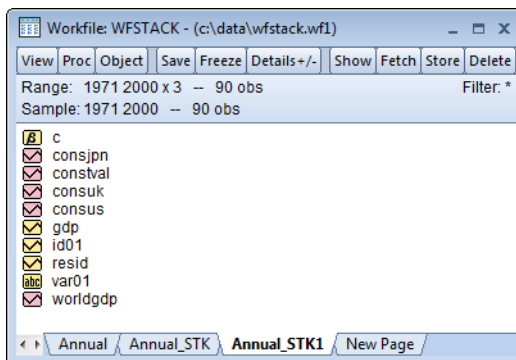
Lastly, we note that since EViews will, by default, create a new page in the existing workfile, all individual series will be stacked or interleaved by creating link objects. If, for example, you enter the stack series list

```
gdp? cons? worldgdp constval
```

the series WORLDGDP and CONSTVAL will be linked to the destination page using the VAR01 values. Alternately, if we were to save the stacked data to a new workfile, by clicking on the **Page Destination** tab and entering appropriate values, EViews will copy the original WORLDGDP and CONSTVAL series to the new page, repeating the values of the original series in the stacked series.

Sorting a Workfile

Basic data in workfiles are held in objects called series. If you click on **Proc/Sort Current Page …** in the workfile toolbar, you can sort all of the series in an unstructured workfile on the basis of the values of one or more of the series. A dialog box will open where you can provide the details about the sort.

If you list two or more series, EViews uses the values of the second series to resolve ties in the first series, and values of the third series to resolve ties in the first and second, and so forth. If you wish to sort in descending order, select the appropriate option in the dialog.

EViews will only sort unstructured workfiles since sorting a dated or structured workfile will break the link between an observation and the corresponding date identifier.

If you attempt to sort a dated or structured workfile, EViews will display a warning informing you that it will first unstructure your data, and then sort the data. Click on **OK** to continue with the operation.
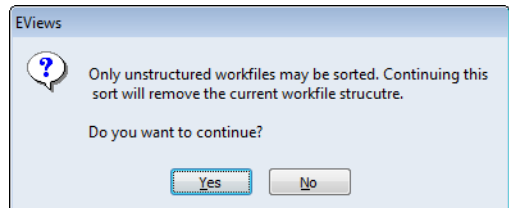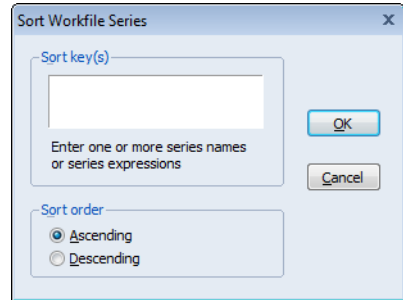
Exporting from a Workfile

MicroTSP Files

You can read or write your workfile in a format that is compatible with MicroTSP. The **Files of type** and **Save as type** dropdown menus in the Open and SaveAs dialogs allow you to handle DOS and Macintosh MicroTSP files. Simply click on the dropdown menu and select either **Old Dos Workfile** or **Old Mac Workfile**, as appropriate. You should be aware, however, that if you choose to save a workfile in MicroTSP format, only basic series data will be saved—the remainder of the workfile contents will be discarded.

Foreign Formats

To save your series (and possibly value map data) into a foreign data source, first select **File/Save As...**, from the workfile menu to bring up the standard file **Save** dialog. Clicking on the **Files of type** dropdown menu brings up a list of the output file types that EViews currently supports.

The data export interface is available for Microsoft Access, Aremos TSD, Gauss Dataset, GiveWin/Pc-Give, Rats 4.x, Rats Portable, SAS program files, SAS Transport, native SPSS (using the SPSS Input/output .DLL installed on your system), SPSS Portable, Stata, TSP Portable, Excel, raw ASCII or binary files, or ODBC Databases (using the ODBC driver already present on your system).

References

Baltagi, Badi H. (2005). *Econometric Analysis of Panel Data, Third Edition*, West Sussex, England: John Wiley & Sons.

Gilley, O.W., and R. Kelley Pace (1996). "On the Harrison and Rubinfeld Data," *Journal of Environmental Economics and Management*, 31, 403–405.

Harrison, D. and D. L. Rubinfeld (1978). "Hedonic Housing Prices and the Demand for Clean Air," *Journal of Environmental Economics and Management*, 5, 81-102.

Chapter 10. EViews Databases

An EViews database resembles a workfile in that it is used to contain a collection of EViews objects. It differs from a workfile in two major ways. First, unlike a workfile, the entire database need not be loaded into memory in order to access an object inside it; an object can be fetched or stored directly to or from the database on disk. Second, unlike a workfile page, the objects in a database are not restricted to being of a single frequency or range. A database could contain a collection of annual, monthly, and daily series, all with different numbers of observations.

EViews databases also differ from workfiles in that they support powerful query features which can be used to search through the database to find a particular series or a set of series with a common property. This makes databases ideal for managing large quantities of data.

While EViews has its own native storage format for databases, EViews also allows direct access to data stored in a variety of other formats through the same database interface. You can perform queries, copy objects to and from workfiles and other databases, and rename and delete objects within a database, all without worrying about in what format the data are actually stored.

Database Overview

An EViews database is a set of files containing a collection of EViews objects. In this chapter we describe how to:

- Create a new database or open an existing database.

- Work with objects in the database, including how to store and fetch objects into workfiles, and how to copy, rename and delete objects in the database.

- Use auto-series to work with data directly from the database without creating a copy of the data in the workfile.

- Use the database registry to create shortcuts for long database names and to set up a search path for series names not found in the workfile.

- Perform a query on the database to get a list of objects with particular properties.

- Use object aliases to work with objects whose names are illegal or awkward.

- Maintain a database with operations such as packing, copying, and repairing.

- Work with remote database links to access data from remote sites.

Database Basics

What is an EViews Database?

An EViews native format database consists of a set of files on disk. There is a main file with the extension .EDB which contains the actual object data, and a number of index files with extensions such as .E0, .E1A and .E1B which are used to speed up searching operations on the database. In normal use, EViews manages these files for the user, so there is no need to be aware of this structure. However, if you are copying, moving, renaming, or deleting an EViews database from outside of EViews (using Windows Explorer for example), you should perform the operation on both the main database file and all the index files associated with the database. If you accidentally delete or damage an index file, EViews can regenerate it for you from the main data file using the repair command (see "Maintaining the Database" on page 343).

The fact that EViews databases are kept on disk rather than in memory has some important consequences. Any changes made to a database cause immediate changes to be made to the disk files associated with the database. Therefore, unlike workfiles, once a change is made to a database, there is no possibility of discarding the change and going back to the previously saved version. Because of this, you should take care when modifying a database, and should consider keeping regular backup copies of databases which you modify frequently.

EViews also allows you to deal with a variety of foreign format databases through the same interface provided to EViews' native format databases. Foreign databases can have many different forms, including files on disk, or data made available through some sort of network server. See "Foreign Format Databases" on page 345 for a discussion of the different types of foreign databases that EViews can access.

Creating a Database

To create a database, simply select **File/New/Database...** from the main menu.

For a native EViews database, simply enter a name for the database in the field labeled **DB File name/path**, then click on the button marked **OK**. This will create a new EViews database in the current path.

To create a database in a different directory, you can enter the full path and database name in the **DB File name/path** edit field. Alternatively, you can browse to the desired directory. Simply click on the **Browse Files** button to call up the common file dialog, and then navigate to the target directory. Enter the name of the new data-

base in the **File name** edit field, then click on the **OK** button to accept the information and close the file dialog. EViews will put the new path and filename in the **DB File name/path** edit field.
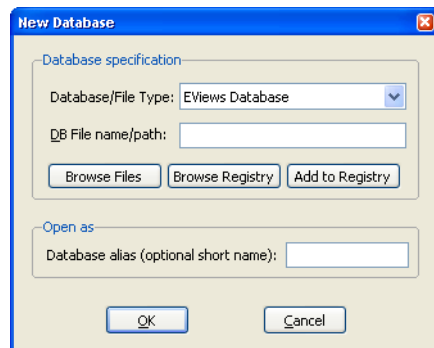
The **Database/File Type** field allows you to create different types of databases. See "Foreign Format Databases" on page 345 for a discussion of working with different database types.

The **Open As** field allows you to specify the shorthand that will be associated with this database. A shorthand is a short text label which is used to refer to the database in commands and programs. If you leave this field blank, a default shorthand will be assigned automatically (see "Database Shorthands" on page 321).

The **Browse Registry** and **Add to Registry** buttons provide a convenient way to recall information associated with a previously registered database or to include the new database in the database registry (see "The Database Registry" on page 331).

A database can also be created from the command line or in a program using the command:

```
dbcreate db_name
```

where db_name is the name of the database using the same rules given above.

The Database Window

When you create a new database, a database window will open on the screen.

The database window provides a graphical interface which allows you to query the database, copy-and-paste objects to and from your workfile, and perform basic maintenance on the database. Note that some database operations can also be carried out directly without first opening the database window.
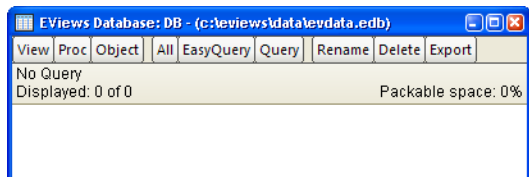
To open a database window for an existing database, select **File/Open/Database…** from the main menu. The same dialog will appear as was used during database creation. To open an EViews database, use the **Browse Files** button to select a file using the common file dialog, then click on **OK** to open the file. A new window should appear representing the open database.

From the command line or in a program, you can open a database window by typing:

```
dbopen db_name
```

Unlike a workfile window, a database window does not display the contents of the database when it is first opened, although it does tell you how many objects are in the database. The second line of the window text shows the number of objects currently displayed (zero when the window is first opened) followed by the total number of objects stored in the database.

You can bring up an alphabetical listing of every object in the database by clicking on the **All** button:

As for a workfile, each object is preceded by a small icon that identifies the type of the object. When performing an **All** query, no other information about the object is visible. However, by double clicking on an object you can bring up a full description of the object including its name, type, modification date, frequency, start and end date (for series), and label.
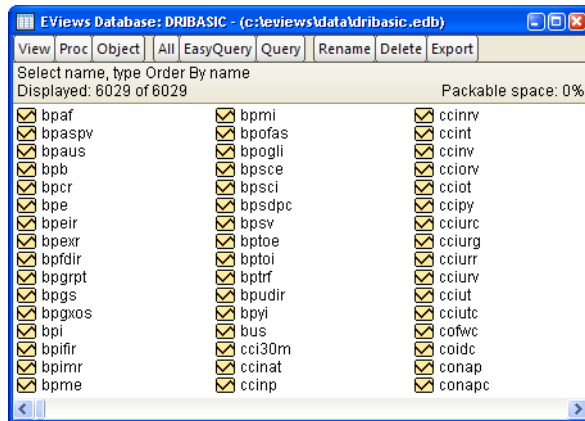
For large databases, the **All** button generally displays too many objects and not enough information about each object. The database query features ("Querying the Database" on page 333) allow you to control precisely which objects should be displayed, and what information about each object should be visible. The text form of the query currently being displayed is always visible in the top line of the database window.

When working with foreign databases, the object names may appear in color to indicate that they are illegal names or that an alias has been attached to an object name (see "Object Aliases and Illegal Names" on page 341).

The "Packable space" field in the database window displays the percentage of unused space in the database that can be recovered by a database pack operation (see "Packing the Database" on page 344).

A brief technical note: having a database window open in EViews generally does not keep a file open at the operating system level. EViews will normally open files only when it is performing operations on those files. Consequently, multiple users may have a database open at the same time and can perform operations simultaneously. There are some limits imposed by the fact that one user cannot read from a database that another user is writing to at the same time. However, EViews will detect this situation and continue to retry the operation until the database becomes available. If the database does not become available within a specified time, EViews will generate an error stating that a "sharing violation" on the database has occurred.

For some foreign formats, even minor operations on a database may require full rewriting of the underlying file. In these cases, EViews will hold the file open as long as the database window is open in order to improve efficiency. The formats that currently behave this way

are Aremos TSD files, RATS Portable files and TSP portable files. When using these formats, only one user at a time may have an open database window for the file.

Database Shorthands

In many situations, EViews allows you to prefix an object name with a database identifier to indicate where the series is located. These database identifiers are referred to as "shorthands". For example, the command:

```
fetch db1::x db2::y
```

indicates to EViews that the object named X is located in the database with the shorthand db1 and the object named y is located in the database with the shorthand db2.

Whenever a database is opened or created, it is assigned a shorthand. The shorthand can be specified by the user in the **Open as** field when opening a database, or using the "As" clause in the dbopen command (see dbopen (p. 349) in the *Command and Programming Reference*). If a shorthand is explicitly specified when opening a database, an error will occur if the shorthand is already in use.

If no shorthand is provided by the user, a shorthand is assigned automatically. The default value will be the name of the database after any path or extension information has been removed. If this shorthand is already in use, either because a database is already open with the same name, or because an entry in the database registry already uses the name, then a numerical suffix is appended to the shorthand, counting upwards until an unused shorthand is found.

For example, if we open two databases with the same name in a program:

```
dbopen test.edb
dbopen test.dat
```

then the first database will receive the shorthand "TEST" and the second database will receive the shorthand "TEST1". If we then issue the command:

```
fetch test::x
```

the object X will be fetched from the EViews database TEST.EDB. To fetch X from the Haver database TEST.DAT we would use:

```
fetch test1::x
```

To minimize confusion, you should assign explicit shorthands to databases whenever ambiguity could arise. For example, we could explicitly assign the shorthand TEST_HAVER to the second database by replacing the second dbopen command with:

```
dbopen test.dat as test_haver
```

The shorthand attached to a database remains in effect until the database is closed. The shorthand assigned to an open database is displayed in the title bar of the database window.

The Default Database

In order to simplify common operations, EViews uses the concept of a *default database*. The default database is used in several places, the most important of which is as the default source or destination for store or fetch operations when an alternative database is not explicitly specified.

The default database is set by opening a new database window, or by clicking on an already open database window if there are multiple databases open on the screen. The name of the default database is listed in the status line at the bottom of the main EViews window (see Chapter 4. "Object Basics," on page 101, for details). The concept is similar to that of the current workfile with one exception: when there are no currently open databases there is still a default database; when there are no currently open workfiles, the current workfile is listed as "none."

EViews .DB? files

Early versions of EViews and MicroTSP supported a much more limited set of database operations. Objects could be stored on disk in individual files, with one object per file. Essentially, the disk directory system was used as a database and each database entry had its own file. These files had the extension ".DB" for series, and .DB followed by an additional character for other types of objects. EViews refers to these collectively as .DB? files.

While the new database features added to EViews provide a superior method of archiving and managing your data, .DB? files provide backward compatibility and a convenient method of distributing data to other programs. Series .DB files are now supported by a large number of programs including TSP, RATS, and SHAZAM. Additionally, some organizations such as the National Bureau of Economic Research (NBER), distribute data in .DB format.

Working with Objects in Databases

Since databases are simply containers of other EViews objects, most of your work with databases will involve moving objects into and out of them. The sections on storing, fetching and exporting objects discuss different ways of doing this.

You will also need to manage the objects inside a database. You can examine previews of the objects to see their contents. You can create duplicate copies of objects, change their names, or remove them from the database entirely. The sections on previewing, copying, renaming and deleting objects discuss how these operations can be carried out.

Storing Objects in the Database

An object may be stored in a database in a number of ways. If you have a workfile open on the screen and would like to store objects contained inside it into a database, just select the objects from the workfile window with the mouse, then click on the **Store** button in the

workfile toolbar. A sequence of dialogs will come up, one for each object selected, which provide a number of options for renaming the object and determining where the object should be stored.

By default, the object will be stored in the default database with the name used as the workfile. Click **Yes** to store the specified object. If you are storing more than one object, EViews will allow you to select **Yes-to-All** to store all of the objects using the current settings.
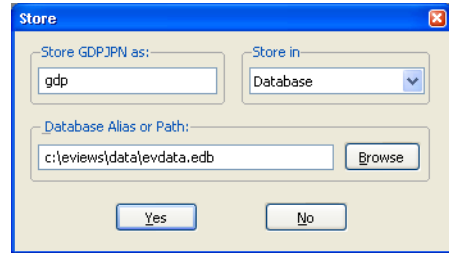
If you would like to store the object with a different name, simply type the new name over the old name in the **Store object_name as** edit box. If you would like to store the object in a different database, either enter the name of the new database in the text box marked **Database Alias or Path** (see "The Database Registry" on page 331 for an explanation of database aliases), or click on the button marked **Browse** to select the database name interactively. To store the object to disk as an EViews .DB? file, click on the arrow to the right of the field labeled **Store in** and select **Individual .DB? files**. You may then specify a path in which to place the file using the field labeled **Path for DB files**.

If there is already an existing object in the database with the same name, EViews will display a dialog. The first and last of the three options should be self explanatory. The second option may only be used if the object you are storing from the workfile and the object already in the database are both series of the same frequency. In this case, EViews will merge the data from the two series so that the new series in the database has all the observations from the series being stored, as well as any observations from the existing series which have not been overwritten. For example, if the existing series in the database is an annual series from 1950 to 1990, and the series being stored is an annual series from 1980 to 1995, the new series will run from 1950 to 1995, with data from the existing series for 1950 to 1979, and data from the new series for 1980 to 1995.
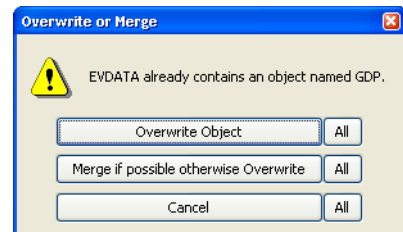
Fetching Objects from the Database

There are a number of ways to fetch objects from a database, most of which are similar to the methods for storing.

The first method is to click on the button marked **Fetch** on the toolbar of the workfile into which you would like to fetch the object. A dialog will come up which is similar to the dialog for store:

The dialog allows you to specify the names of the objects to fetch, and the database or directory from which to retrieve them.

Enter the names of the objects you would like to fetch in the field **Objects to Fetch**. Alternatively, you can use the mouse to select objects from the workfile window before clicking on the **Fetch** button, in which case the names of these objects will appear automatically.
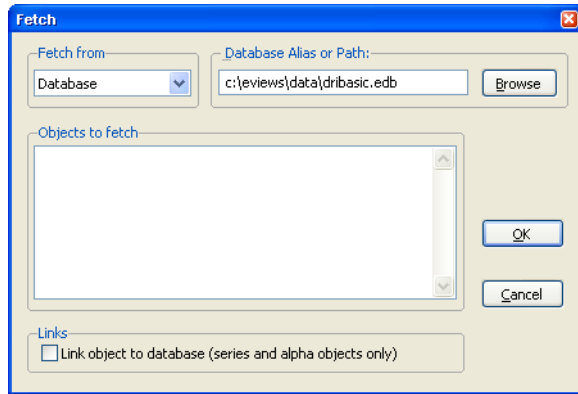
The fields labeled **Database Alias or Path** and **Fetch from** are the same as for the store dialog with one exception. In addition to **EViews Database** and **Individual .DB? files, Fetch from** has an option titled **Search Databases.** This option tells EViews to search multiple databases for objects which match the specified names. To use this option, you must first define a search order in the database registry (see "The Database Registry" on page 331).

The checkbox labeled **Link objects to database** on the bottom of the dialog instructs EViews to bring any listed series or alpha objects into the workfile as links to the data in the database. When you open an existing workfile containing database links, EViews will prompt you for whether you wish to refresh the data series. If you click on **No**, EViews will retain the existing data in the link, otherwise the data will be re-imported from the database when you load the workfile. You may also update existing links manually by selecting **Object/Manage Links & Formulae...** in the workfile window, specifying the objects to update, and clicking on the associated **Refresh** button.

When you click on **OK**, EViews will fetch all the objects. If an object which is being fetched is already contained in the workfile, a dialog will appear asking whether to replace the object or not. Click on **Yes** to replace the object in the workfile or **No** to leave the object in the workfile unchanged.

Because a workfile has a fixed frequency and range, fetching a series into a workfile may cause the data in the series to be modified to match the frequency and range of the workfile (see "Frequency Conversion" on page 170). Be aware that loading a series into a workfile then saving it back into the database can cause truncation and frequency conversion of the series stored in the database.

Object/Update selected from DB... from the workfile toolbar is the same as **Fetch** except that there is no overwrite warning message. If the object in the database is the same type as the one in the workfile, it is automatically overwritten. If it is of a different type, the fetch

does not proceed. Update is also available from the **Object** button in individual object windows.

Database Export

You can also move data into a workfile from the database window. From an open database window, select the objects you would like to copy using the mouse, then click on the button marked **Export** in the toolbar at the top of the database window. The **Database Export** dialog will appear on the screen:

When you click on the down arrow on the right of the field labeled **Workfile**, a list of all workfiles that are currently open will appear from which you may choose the workfile into which you would like to copy the objects. In addition, you may use the **Page** drop down menu to select an existing page in the selected workfile, or to create a new page. Clicking on the button marked **OK** will copy the selected objects to specified page of the selected workfile.
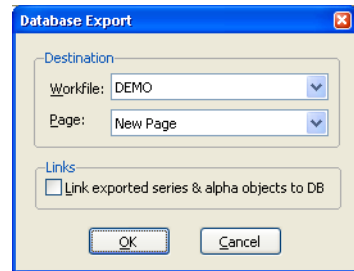
There is an extra option in the list of open workfiles for specifying a new workfile as your copy destination. If you select **New Workfile**, EViews will create a new workfile containing the objects you have selected. After you click on **OK**, a second dialog will appear in which you can set the frequency and range of the workfile to be created. The default frequency is set to the lowest frequency of any of the objects selected, and the default range is set to cover all the data points contained in the objects. Clicking on **OK** will open a new workfile window and copy the selected objects into it, performing frequency conversion where necessary.

Lastly, you may export your series or alpha objects to the workfile as database links. When you reopen your workfile containing database links, EViews will prompt you for whether you wish to refresh the data series from the database.

Previewing Objects in the Database

To preview one or more objects in the database, select the object or objects and, if necessary, open the preview window. The preview window will display a preview of the contents of the first selected object.

See "Previewing Objects," on page 107 for details.

Copying Objects

In addition to the above methods for moving objects, EViews provides general support for the copying of objects between any two EViews container objects (workfiles or databases). You may use these features to move objects between two databases or between two work-

files, to create duplicate copies of objects within a workfile or database, or as an alternative method for store and fetch.

Copy-and-Paste

For copying objects between containers, the procedure is very similar no matter what types of container objects are involved. Before you start, make sure that the windows for both containers are open on the screen. In the container *from which* you would like to copy the objects, select the objects then click on **Edit/Copy** in the EViews program menu. Click on the container object *into which* you would like to paste the objects, then select **Edit/Paste** or **Edit/Paste Special...** from the EViews program menu.

Depending on the types of the two containers, you may be presented with one or more dialogs. If, for example, you are performing a copy to or from a database, and click on **Edit/Paste**, the standard **Store** or **Fetch** dialogs will appear as if you had carried out the operations using the toolbar buttons on the workfile window. If you click on **Edit/Paste Special...**, an alternate dialog will be displayed, allowing you to override the default frequency conversion methods.

If, instead, you are copying between two workfiles, selecting **Edit/Paste** will simply copy the series using the default frequency conversion if necessary. You will only be prompted with a dialog if there is name collision. Selecting **Edit/Paste Special...** will display a dialog allowing you to override the default conversion methods.

Drag-and-Drop

An alternative to copy-and-paste is to use drag-and-drop to copy files between containers. If you use the right-mouse button to select the series, dropping them on the new tab is equivalent to copying and then selecting **Edit/Paste Special...**

Copy Procedure

You may perform similar operations using the object copy procedure. From the main menu select **Object/Copy** (this may appear as **Object/Copy selected...**). The **Object Copy** dialog will be displayed.

The **Source** field specifies the object or objects you would like to copy, the **Destination** field specifies where you would like to copy them and what names they should be given.

The **Source** field should be filled in with an expression of the form:

```
source_db::source_pattern
```
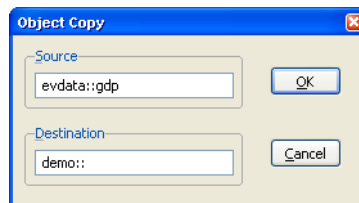
where `source_db::` is optional, and indicates which database the objects should be copied from (if no database name is supplied, the source is taken to be the default workfile), and `source_pattern` is either a simple object name or a name pattern. A name pattern may include the wildcard characters "?" which matches any single character, and "*" which matches zero or more characters.

The **Destination** field should be filled in with an expression of the form:

    dest_db::dest_name

where `dest_db::` is again optional, and indicates which database the objects should be copied to (if no database name is supplied, the destination is taken to be the default workfile), and `dest_name`, which is also optional, is the name to be given to the new copy of the object. If no name is given, the object will be copied with its existing name. If a pattern was used when specifying the source, a pattern must also be used when specifying the destination (see of the *Command and Programming Reference*).

For example, to copy an object from the database DB1 to the database DB2, keeping the existing name, you would fill in the dialog:

    source: db1::object_name
    destination: db2::

where OBJECT_NAME is the original name as displayed by EViews.

To copy all the objects in the database DB1 beginning with the letter X into the current workfile, changing the names so that they begin with Y, you would fill in the dialog

    source: db1::x*
    destination: y*

To make a duplicate copy of the object named ABC in the database DB1, giving it the new name XYZ, you would fill in the dialog:

    source: db1::abc
    destination: db1::xyz

Renaming Objects in the Database

You may rename an object in the database by selecting the object in an open database window, then clicking on the button marked **Rename** in the database window toolbar. A dialog will come up in which you can modify the existing name or type in a new name. You can rename several objects at the same time using wildcard patterns and the `rename` command.

Deleting Objects From the Database

To delete objects from the database, select the objects in an open database window, then click on the button marked **Delete** on the database window toolbar. You may delete several

objects at the same time using wildcard patterns. There is also a `delete` command. See `delete` (p. 353) in the *Command and Programming Reference* for details.

Store, Fetch, and Copy of Group Objects

A group object in EViews is essentially a list of series names that form the group. The data of each series are contained in the series object, not in the group object. When you do a store, fetch, or copy operation on a group object, an issue arises as to whether you want to do the operation on each of the series or to the group definition list.

Storing a Group Object

When you store a group object to a database, there are four available options:

- Store the group definition and the series as separate objects: stores the group object (only its definition information) and each of its series as separate objects in the database. If any of the series already exist in the database, EViews will ask whether to overwrite the existing series if in interactive mode, and will error if in batch mode.

- Store the group definition and the series as one object: stores each series within the group object. A group object that contains series data will have an icon G+ in the database directory window. A group object with only its definition information will have the usual icon G. If you use this option, you can store two different series with the same name (with one of the series as member of a group).

- Store only the series (as separate objects): only stores each series as separate objects in the database. If you want to store a long list of series into a database, you can create a temporary group object that contains those series and issue the store command only once.

- Store only the group definition: stores only the group definition information; none of the series data are stored in the database. This option is useful if you want to update the member data from the database but want to keep the group information (e.g. the dated data table settings) in the group.
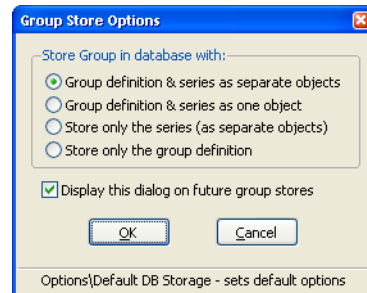
By default, EViews will display a dialog asking you to select a group store option every time you store a group object. You can, however, instruct EViews to suppress the dialog and use the global option setting. Simply click on **Options/Database Storage Defaults…** in the main EViews menu to bring up a dialog that allows you both to set the global storage options, and to suppress the group store option dialog.

Fetching a Group Object

When you fetch a group object to a database, there are three options available:

- Fetch both group definition and the actual series: fetches both group definition and its series as separate objects. If any of the series defined in the group is not found in the database, the corresponding series will be created in the workfile filled with NAs. If any of the series already exist in the workfile, EViews will ask whether to overwrite the existing series if in interactive mode, and will error if in batch mode.

- Fetch only the series in the group: only fetches each series defined in the group. If the series exists both within the group object (with a G+ icon) and as a separate series object in the database, the series within the group object will be fetched.

- Fetch only the group definition: fetches only the group definition (but not the series data). If any of the series defined in the group does not exist in the workfile, EViews will create the corresponding series filled with NAs.

You can click on **Options/Database Default Storage Options...** in the main menu to bring up a dialog that allows you both to set the global fetch options, and to suppress the fetch option dialog.

Copying Group Objects between Workfiles and Databases

You can also copy groups between different containers. The options that are available will differ depending on the type of source and destination container:

- Copy from workfile to database: same options as the store operation.

- Copy from database to workfile: same options as the fetch operation.

- Copy from workfile to workfile: both the group definition and series will be copied.

- Copy from database to database. If the group object contains only the group definition (with a G icon), only the group definition will be copied. If the group object also contains its series data (with a G+ icon), then the group will be copied containing the series data and the copied group will also appear with a G+ icon.

Database Auto-Series

We have described how to fetch series into a workfile. There is an alternative way of working with databases which allows you to make direct use of the series contained in a database without first copying the series. The advantage of this approach is that you need not go through the process of importing the data every time the database is revised. This approach

follows the model of auto-series in EViews as described in "Auto-series," beginning on page 193.

There are many places in EViews where you can use a series expression, such as log(X), instead of a simple series name, and EViews will automatically create a temporary auto-series for use in the procedure. This functionality has been extended so that you can now directly refer to a series in a database using the syntax:

```
db_name::object_name
```

where db_name is the shorthand associated with the database. If you omit the database name and simply prefix the object name with a double colon like this:

```
::object_name
```

EViews will look for the object in the default database.

A simple example is to generate a new series:

```
series lgdp = log(macro_db::gdp)
```

EViews will fetch the series named GDP from the database with the shorthand MACRO_DB, and put the log of GDP in a new series named LGDP in the workfile. It then deletes the series GDP from memory, unless it is in use by another object. Note that the generated series LGDP only contains data for observations within the current workfile sample.

You can also use auto-series in a regression. For example:

```
equation eq1.ls log(db1::y) c log(db2::x)
```

This will fetch the series named Y and X from the databases named DB1 and DB2, perform any necessary frequency conversions and end point truncation so that they are suitable for use in the current workfile, take the log of each of the series, then run the requested regression. Y and X are then deleted from memory unless they are otherwise in use.

The auto-series feature can be further extended to include automatic searching of databases according to rules set in the database registry (see "The Database Registry" on page 331). Using the database registry you can specify a list of databases to search whenever a series you request cannot be found in the workfile. With this feature enabled, the series command:

```
series lgdp = log(gdp)
```

looks in the workfile for a series named GDP. If it is not found, EViews will search through the list of databases one by one until a series called GDP is found. When found, the series will be fetched into EViews so that the expression can be evaluated. Similarly, the regression:

```
equation logyeq.ls log(y) c log(x)
```

will fetch Y and X from the list of databases in the registry if they are not found in the work-file. Note that the regression output will label all variables with the database name from which they were imported.

In general, using auto-series directly from the database has the advantage that the data will be completely up to date. If the series in the database are revised, you do not need to repeat the step of importing the data into the workfile. You can simply reestimate the equation or model, and EViews will automatically retrieve new copies of any data which are required.

There is one complication to this discussion which results from the rules which regulate the updating and deletion of auto-series in general. If there is an existing copy of an auto-series already in use in EViews, a second use of the same expression will not cause the expression to be reevaluated (in this case reloaded from the database); it will simply make use of the existing copy. If the data in the database have changed since the last time the auto-series was loaded, the new expression will use the old data.

One implication of this behavior is that a copy of a series from a database can persist for any length of time if it is stored as a member in a group. For example, if you type:

```
show db1::y db2::x
```

this will create an untitled group in the workfile containing the expressions `db1::y` and `db2::x`. If the group window is left open and the data in the database are modified (for example by a `store` or a `copy` command), the group and its window will not update automatically. Furthermore, if the regression:

```
equation logyeq.ls log(db1::y) c log(db2::x)
```

is run again, this will use the copies of the series contained in the untitled group; it will not re-fetch the series from the database.

The Database Registry

The database registry is a file on disk that manages a variety of options which control database operations. It gives you the ability to assign short alias names that can be used in place of complete database paths, and also allows you to configure the automatic searching features of EViews.

Selecting **Options/Database Registry…** from the main menu brings up the **Database Registry** dialog allowing you to view and edit the database registry:

The box labeled **Registry Entries** lists the databases that have been registered with EViews. The first time you bring up the dialog, the box will usually be empty. If you click on the **Add new entry** button, a **Database Registry Entry** dialog appears.

There are three things you must specify in the dialog: the full name (including path) of the database, the alias which you would like to associate with the database, and the option for whether you wish to include the database in automatic searches.

The full name and path of the database should be entered in the top edit field. Alternatively, click the **Browse** button to select your database interactively.

The next piece of information you must provide is a *database alias*: a short name that you can use in place of the full database path in EViews commands. The database alias will also be used by EViews to label database auto-series. For example, suppose you have a database named DRIBASIC located in the subdirectory C:\EVIEWS\DATA. The following regression command is legal but awkward:

```
equation eq1.ls c:\eviews\data\dribasic::gdp c
    c:\eviews\data\dribasic::gdp(-1)
```

Long database names such as these also cause output labels to truncate, making it difficult to see which series were used in a procedure.

By assigning full database path and name the alias DRI, we may employ the more readable command:
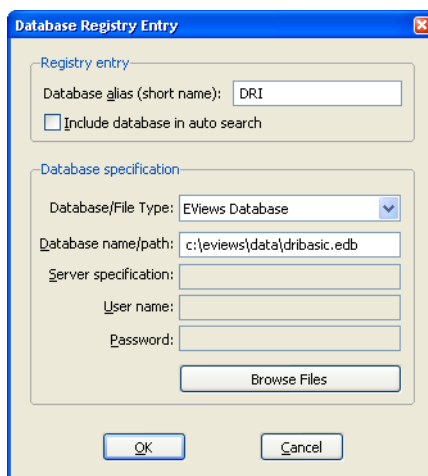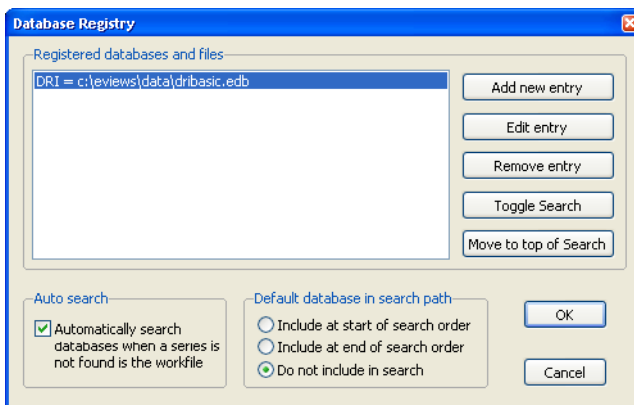
```
equation eq1.ls dri::gdp c dri::gdp(-1)
```

and the regression output will be labeled with the shorter names. To minimize the possibility of truncation, we recommend the use of short alias names if you intend to make use of database auto-series.

Finally, you should tell EViews if you want to include the database in automatic database searches by checking the **Include in auto search** checkbox. Click on **OK** to add your entry to the list

Any registry entry may be edited, deleted, switched on or off for searching, or moved to the top of the search order by highlighting the entry in the list and clicking the appropriate button to the right of the list box.

The remainder of the **Database Registry** dialog allows you to set options for automatic database searching. The **Auto-search** checkbox is used to control EViews behavior when you enter a command involving a series name which cannot be found in the current workfile. If this checkbox is selected, EViews will automatically search all databases that are registered for searching, before returning an error. If a series with the unrecognized name is found in any of the databases, EViews will create a database auto-series and continue with the procedure.

The last section of the dialog, **Default Database in Search Order**, lets you specify how the default database is treated in automatic database searches. Normally, when performing an automatic search, EViews will search through the databases contained in the **Registry Entries** window in the order that they are listed (provided that the **Include in auto search** box for that entry has been checked). These options allow you to assign a special role to the default database when performing a search.

- **Include at start of search order—**means that the current default database will be searched first, before searching the listed databases.

- **Include at end of search order—**means that the current default database will be searched last, after searching the listed databases.

- **Do not include in search—**means that the current default database will not be searched unless it is already one of the listed databases.

Querying the Database

A great deal of the power of the database comes from its extensive query capabilities. These capabilities make it easy to locate a particular object, and to perform operations on a set of objects which share similar properties.

The query capabilities of the database can only be used interactively from the database window. There are two ways of performing a query on the database: the easy mode and the

advanced mode. Both methods are really just different ways of building up a text query to the database. The easy mode provides a simpler interface for performing the most common types of queries. The advanced mode offers more flexibility at the cost of increased complexity.

Easy Queries

To perform an easy query, first open the database, then click on the **EasyQuery** button in the toolbar at the top of the database window. The Easy Query dialog will appear containing two text fields and a number of check boxes:

There are two main sections to this dialog: **Select** and **Where**. The **Select** section determines which fields to display for each object that meets the query condition. The **Where** section allows you to specify conditions that must be met for an object to be returned from the query. An Easy Query allows you to set conditions on the object name, object description, and/or object type.

The two edit fields (**name** and **description**) and the set of check boxes (object **type**) in the **Where** section provide three filters of objects that are returned from the query to the database. The filters are applied in sequence (using a logical 'and' operation) so that objects in the database must meet all of the criteria selected in order to appear in the results window of the query.
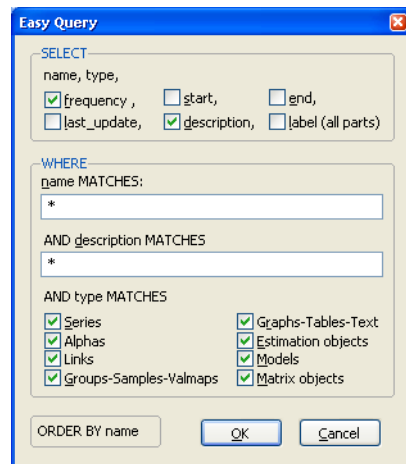
The **name** and **description** fields are each used to specify a *pattern expression* that the object must meet in order to satisfy the query. The simplest possible pattern expression consists of a single pattern. A pattern can either be a simple word consisting of alphanumeric characters, or a pattern made up of a combination of alphanumeric characters and the wildcard symbols "?" and "*", where "?" means to match any one character and "*" means to match zero or more characters. For example:

```
pr?d*ction
```

would successfully match the words production, prediction, and predilection. Frequently used patterns include "s*" for words beginning in "S," "*s" for words ending in "S," and "*s*" for words containing "S." Upper or lower case is not significant when searching for matches.

Matching is done on a word-by-word basis, where at least one word in the text must match the pattern for it to match overall. Since object names in a database consist of only a single word, pattern matching for names consists of simply matching this word.

For descriptions, words are constructed as follows: each word consists of a set of consecutive alphanumeric characters, underlines, dollar signs, or apostrophes. However, the following list words are explicitly ignored: "a," "an," "and," "any," "are," "as," "be," "between," "by," "for," "from," "if," "in," "is," "it," "not," "must," "of," "on," "or," "should," "that," "the," "then," "this," "to," "with," "when," "where," "while." (This is done for reasons of efficiency, and to minimize false matches to patterns from uninteresting words.) The three words "and," "or," and "not" are used for logical expressions.

For example:

```
bal. of p'ment: seas.adj. by X11
```

is broken into the following words: "bal," "p'ment," "seas," "adj," and "x11." The words "of" and "by" are ignored.

A pattern expression can also consist of one or more patterns joined together with the logical operators "and," "or," and "not" in a manner similar to that used in evaluating logical expressions in EViews. That is, the keyword `and` requires that both the surrounding conditions be met, the keyword `or` requires that either of the surrounding conditions be met, and the keyword `not` requires that the condition to the right of the operator is not met. For example:

```
s* and not *s
```

matches all objects which contain words which begin with, but do not end with, the letter "S".

More than one operator can be used in an expression, in which case parentheses can be added to determine precedence (the order in which the operators are evaluated). Operators inside parentheses are always evaluated logically prior to operators outside parentheses. Nesting of parentheses is allowed. If there are no parentheses, the precedence of the operators is determined by the following rules: `not` is always applied first; `and` is applied second; and `or` is applied last. For example:

```
p* or s* and not *s
```

matches all objects which contain words beginning with P, or all objects which contain words which begin with, but do not end with, the letter S.

The third filter provided in the **Easy Query** dialog is the ability to filter by object type. Simply select the object types which you would like displayed, using the set of check boxes near the bottom of the dialog.

Advanced Queries

Advanced queries allow considerably more control over both the filtering and the results which are displayed from a query. Because of this flexibility, advanced queries require some understanding of the structure of an EViews database to be used effectively.

Each object in an EViews database is described by a set of fields. Each field is identified by a name. The current list of fields includes:

name	The name of the object.
type	The type of the object.
last_write	The time this object was last written to the database.
last_update	The time this object was last modified by EViews.
freq	The frequency of the data contained in the object.
start	The date of the first observation contained in the object.
end	The date of the last observation contained in the object.
obs	The number of data points stored in the series (including missing values).
description	A brief description of the object.
source	The source of the object.
units	The units of the object.
remarks	Additional remarks associated with the object.
history	Recent modifications of the object by EViews.
display_name	The EViews display name.

An advanced query allows you to examine the contents of any of these fields, and to select objects from the database by placing conditions on these fields. An advanced query can be performed by opening the database window, then clicking on the button marked **Query** in the toolbar at the top of the window. The **Advanced Query** dialog is displayed.

The first edit field labeled **Select:** is used to specify a list of all the fields that you would like displayed in the query results. Input into this text box consists of a series of field names separated by commas. Note that the name and type fields are always fetched automatically.

The ordering of display of the results of a query is determined by the **Order By** edit field. Any field name can be entered into this box, though some fields are likely to be more useful than others. The description field, for example, does not provide a useful ordering of the objects. The **Order By** field can be useful for grouping together objects with the same value of a particular field. For example, ordering by type is an effective way to group together the results so that objects of the same type are placed
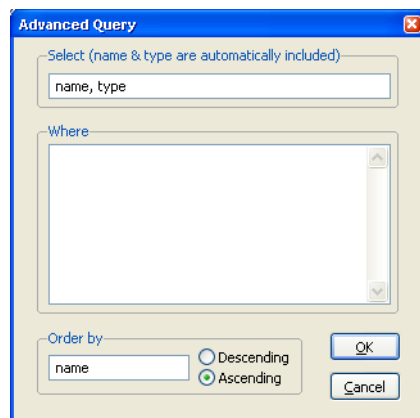
together in the database window. The **Ascending** and **Descending** buttons can be used to reverse the ordering of the objects. For example, to see objects listed from those most recently written in the database to those least recently written, one could simply sort by the field `last_write` in **Descending** order.

The **Where** edit field is the most complicated part of the query. Input consists of a logical expression built up from conditions on the fields of the database. The simplest expression is an operator applied to a single field of the database. For example, to search for all series which are of monthly or higher frequencies (where higher frequency means containing more observations per time interval), the appropriate expression is:

```
freq >= monthly
```

Field expressions can also be combined with the logical operators `and`, `or` and `not` with precedence following the same rules as those described above in the section on easy queries. For example, to query for all series of monthly or higher frequencies which begin before 1950, we could enter the expression:

```
freq >= monthly and start < 1950
```

Each field has its own rules as to the operators and constants which can be used with the field.

Name

The name field supports the operators " < ", " < = ", " > ", " > = ", " = ", and " < > " to perform typical comparisons on the name string using alphabetical ordering. For example,

```
name >= c and name < m
```

will match all objects with names beginning with letters from C to L. The name field also supports the operator "matches". This is the operator which is used for filtering the name field in the easy query and is documented extensively in the previous section. Note that if `matches` is used with an expression involving more than one word, the expression must be contained in quotation marks. For example,

```
name matches "x* or y*" and freq = quarterly
```

is a valid query, while

```
name matches x* or y* and freq = quarterly
```

is a syntax error because the part of the expression that is related to the `matches` operator is ambiguous.

Type

The type field can be compared to the standard EViews object types using the " = " operator and the keywords: `alpha`, `coef`, `equation`, `factor`, `graph`, `group`, `logl`, `matrix`, `model`, `pool`, `sample`, `scalar`, `series`, `spool`, `sspace`, `string`, `svector`, `sym`, `system`, `table`,

`text`, `var`, `vector`. Relational operators are defined for the type field, although there is no particular logic to the ordering. The ordering can be used, however, to group together objects of similar types in the **Order By** field.

Freq

The frequency field has one of the following values:

u	Undated
2y, 3y, 4y, 5y, 6y, 7y, 8y, 9y, 10y, 20y	Multi-year in intervals of: 2, 3, 4, 5, 6, 7, 8, 9, 10, or 20 years, respectively.
a	Annual
s	Semiannual
q	Quarterly
m	Monthly
bm	Bimonthly
f	Fortnight
t	Ten-day (daily in intervals of ten)
w	Weekly
5	5 day daily, Monday through Friday
7	7 day daily
h	Hourly
min	Minute
sec	Second

Any word beginning with the letter above is taken to denote that particular frequency, so that monthly can either be written as "m" or "monthly". Ordering over frequencies is defined so that a frequency with more observations per time interval is considered "greater" than a series with fewer observations per time interval. The operators " < ", " > ", " < = ", " > = ", " = ", " < > " are all defined according to these rules. For example,

```
freq <= quarterly
```

will match objects whose frequencies are quarterly, semiannual, annual or undated.

Start and End

Start and end dates use the following representation. A date from an annual series is written as an unadorned year number such as "1980". A date from a semiannual series is written as a year number followed by an "S" followed by the six month period, for example "1980S2". The same pattern is followed for quarterly and monthly data using the letters "Q" and "M"

between the year and period number. Weekly, 5-day daily, and 7-day daily data are denoted by a date in the format:

```
mm/dd/yyyy
```

where `m` denotes a month digit, `d` denotes a day digit, and `y` denotes a year digit.

Operators on dates are defined in accordance with calendar ordering where an earlier date is less than a later date. Where a number of days are contained in a period, such as for monthly or quarterly data, an observation is ordered according to the first day of the period. For example:

```
start <= 1950
```

will include dates whose attributed day is the first of January 1950, but will not include dates which are associated with other days in 1950, such as the second, third, or fourth quarter of 1950. However, the expression:

```
start < 1951
```

would include all intermediate quarters of 1950.

Last_write and Last_update

As stated above, `last_write` refers to the time the object was written to disk, while `last_update` refers to the time the object was last modified inside EViews. For example, if a new series was generated in a workfile, then stored in a database at some later time, `last_write` would contain the time that the store command was executed, while `last_update` would contain the time the new series was generated. Both of these fields contain date and time information which is displayed in the format:

```
mm/dd/yyyy hh:mm
```

where `m` represents a month digit, `d` represents a day digit, `y` represents a year digit, `h` represents an hour digit and `m` represents a minute digit.

The comparison operators are defined on the time fields so that earlier dates and times are considered less than later dates and times. A typical comparison has the form:

```
last_write >= mm/dd/yyyy
```

A day constant always refers to twelve o'clock midnight at the beginning of that day. There is no way to specify a particular time during the day.

Description, Source, Units, Remarks, History, Display_name

These fields contain the label information associated with each object (which can be edited using the Label view of the object in the workfile). Only one operator is available on these fields, the `matches` operator, which behaves exactly the same as the description field in the section on easy queries.

Query Examples

Suppose you are looking for data related to gasoline consumption and gasoline prices in the database named DRIBASIC. First open the database: click **File/Open**, select **Files of type: Database.edb** and locate the database. From the database window, click **Query** and fill in the **Advanced Query** dialog as follows:

> **Select:** name, type, freq, description
> **Where:** description matches gasoline

If there are any matches, the results are displayed in the database window similar to the following:
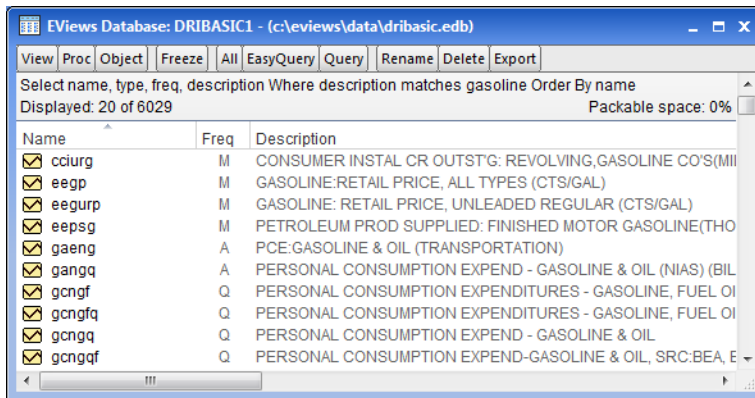
To view the contents of all fields of an item, double click on its name. EViews will open an **Object Description** window that looks as follows:
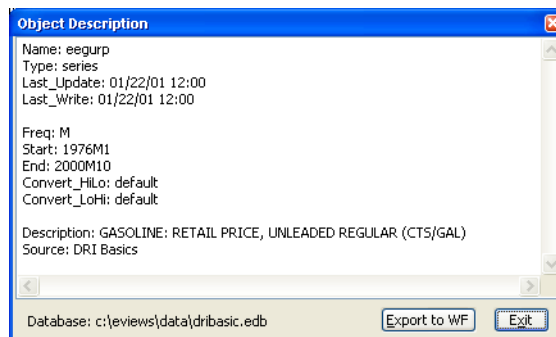
To further restrict your search to series with at least quarterly frequency and to display the start and end dates of the results, click **Query** and again and modify the fields as follows:
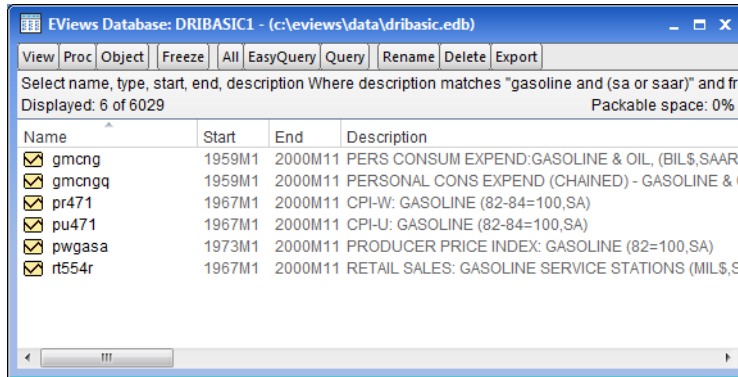
> **Select**: name, type, start, end, description
> **Where**: description matches gasoline and freq>=q

If you are interested in seasonally adjusted series, which happen to contain `sa` or `saar` in their description in this database, further modify the fields to

> **Select**: `name, type, start, end, description`
>
> **Where**: `description matches "gasoline and (sa or saar)" and freq>=q`

The display of the query results now looks as follows:

The columns of the database display can be reordered by dragging each to its desired location, or resized by dragging the column header's right edge. You can also sort by any of the columns by clicking the column header to alternate between ascending and descending sort.

By adding columns, the database display can be customized to provide additional information and grouping capabilities. See "Workfile Details Display" on page 62 for a thorough discussion of customized attributes in this view. Most importantly, it describes how to view all available attributes in the database, by right-clicking in a column header and selecting **Edit Columns...**. The methods presented in the discussion detail how you can view specific characteristics across all items in your database.

Object Aliases and Illegal Names

When working with a database, EViews allows you to create a list of aliases for each object in the database so that you may refer to each object by a different name. The most important use of this is when working with a database in a foreign format where some of the names used in the database are not legal EViews object names. However, the aliasing features of EViews can also be used in other contexts, such as to assign a shorter name to a series with an inconveniently long name.

The basic idea is as follows: each database can have one or more *object aliases* associated with it where each alias entry consists of the name of the object in the database and the name by which you would like it to be known in EViews.

The easiest way to create an object alias for an illegal name is to attempt to fetch the object with the illegal name into EViews. If you are working with query results, you can tell which object names are illegal because they will be displayed in the database window in red. When you try to fetch an object with an illegal name, a dialog will appear.

The field labeled **EViews Name** initially contains the illegal name of the database object. You should edit this to form a legal EViews object name. In this example, we could change the name C to CONSUMP. The checkbox labeled **Add this name to the database alias list** (which is not checked by default), determines whether you want to create a permanent association between the name you have just typed and the illegal name. If you check the box, then whenever you use the edited object name in the future, EViews will take it to refer to the underlying illegal name. The edited name acts as an *alias* for the underlying name. It is as though you had renamed the object in the database to the new legal name, except that you have not actually modified the database itself, and your changes will not affect other users of the database.
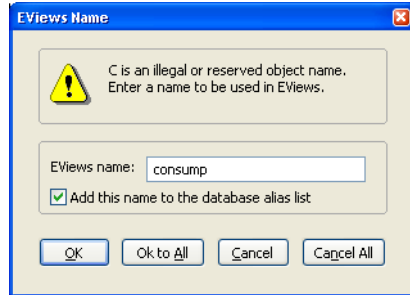
When EViews displays an object in the database window for which an alias has been set, EViews will show the alias, rather than the underlying name of the object. In order to indicate that this substitution has been done, EViews displays the name of the aliased object in blue.

Creating an alias can cause *shadowing* of object names. Shadowing occurs when you create an alias for an object in the database, but the name you use as an alias is the name of another object in the database. Because the existence of the alias will stop you from accessing the other object, that object is said to be shadowed. To indicate that an object name being displayed has been shadowed, EViews displays the name of shadowed objects in green. You will not be able to fetch an object which has been shadowed without modifying either its name or the alias which is causing it to be shadowed. Even if the shadowed series is explicitly selected with the mouse, operations performed on the series will use the series with the conflicting alias, not the shadowed series.

You can view a list of the aliases currently defined for any database by clicking on the **View** button at the top of the database window, then selecting **Object Aliases**. A list of all the aliases will be displayed in the window.
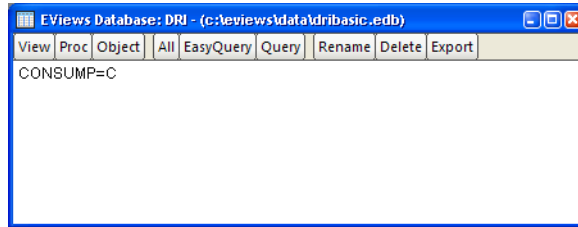
Each line represents one alias attached to the database and follows the format:

```
alias = database_object_name
```

You can edit the list of aliases to delete unwanted entries, or you can type in, or cut-and-paste, new entries into the file. You must follow the rule that both the set of aliases and the set of database names do not contain any repeated entries. (If you do not follow this rule, EViews will refuse to save your changes). To save any modifications you have made, simply switch back to the **Object Display** view of the database. EViews will prompt you for whether you want to save or discard your edits.

The list of currently defined database aliases for all databases is kept in the file OBALIAS.INI in the EViews installation directory. If you would like to replicate a particular set of aliases onto a different machine, you should copy this file to the other machine, or use a text editor to combine a portion of this file with the file already in use on the other machine. You must exit and restart EViews to be sure that EViews will reread the aliases from the file.

Maintaining the Database

In many cases an EViews database should function adequately without any explicit mainte-nance. Where maintenance is necessary, EViews provides a number of procedures to help you perform common tasks.

Database File Operations

Because EViews databases are spread across multiple files, all of which have the same name but different extensions, simple file operations like copy, rename and delete require multiple actions if performed outside of EViews. The **Proc** button in the database window toolbar contains the procedures **Copy the database**, **Rename the database**, and **Delete the data-base** that carry out the chosen operation on all of the files that make up the database.

Note that file operations do not automatically update the database registry. If you delete or rename a database that is registered, you should either create a new database with the same name and location, or edit the registry.

Packing the Database

If many objects are deleted from an EViews database without new objects being inserted, a large amount of unused space will be left in the database. In addition, if objects are frequently overwritten in the database, there will be a tendency for the database to grow gradually in size. The extent of growth will depend on the circumstances, but a typical database is likely to stabilize at a size around 60% larger than what it would be if it were written in a single pass.

A database can be compacted down to its minimum size by using the pack procedure. Simply click on the button marked **Proc** in the toolbar at the top of the database window, then select the menu item **Pack the Database**. Depending on the size of the database and the speed of the computer which you are using, performing this operation may take a significant amount of time.

You can get some idea of the amount of space that will be reclaimed during a pack by looking at the Packable Space percentage displayed in the top right corner of the database window. A figure of 30%, for example, indicates that roughly a third of the database file consists of unused space. A more precise figure can be obtained from the **Database Statistics** view of a database. The number following the label "unused space" gives the number of unused bytes contained in the main database file.

Dealing with Errors

EViews databases are quite robust, so you should not experience problems working with them on a regular basis. However, as with all computer files, hardware or operating system problems may produce conditions under which your database is damaged.

The best way to protect against damage to a database is to make regular backup copies of the database. This can be performed easily using the **Copy the Database** procedure documented above. EViews provides a number of other features to help you deal with damaged databases.

Damaged databases can be divided into two basic categories depending on how severely the database has been damaged. A database which can still be opened in a database window but generates an error when performing some operations may not be severely damaged and may be reparable. A database which can no longer be opened in a database window is severely damaged and will need to be rebuilt as a new database.

EViews has two procedures designed for working with databases which can be opened: **Test Database Integrity** and **Repair Database**. Both procedures are accessed by clicking on the button marked **Proc** in the database window toolbar, then selecting the appropriate menu item.

Test Database Integrity conducts a series of validity checks on the main database and index files. If an error is detected, a message box will be displayed, providing some information as to the type of error found and a suggestion as to how it might be dealt with. Because testing performs a large number of consistency checks on the database files, it may take considerable time to complete. You can monitor its progress by watching the messages displayed in the status line at the bottom of the EViews window. Testing a database does not modify the database in any way, and will never create additional damage to a database.

Repair Database will attempt to automatically detect and correct simple problems in the database. Although care has been taken to make this command as safe as possible, it will attempt to modify a damaged database, so it is probably best to make a back up copy of a damaged database before running this procedure.

Rebuilding the Database

If the database is badly corrupted, it may not be possible for it to be repaired. In this case, EViews gives you the option of building a new database from the old one using the `dbrebuild` command. This operation can only be performed from the command line (since it may be impossible to open the database). The command is:

```
dbrebuild old_dbname new_dbname
```

The `dbrebuild` command does a low level scan through the main data file of the database `old_dbname` looking for any objects which can be recovered. Any such objects are copied into the new database `new_dbname`. This is a very time consuming process, but it will recover as much data as possible from even heavily damaged files.

Foreign Format Databases

While most of your work with databases will probably involve using EViews native format databases, EViews also gives you the ability to access data stored in a variety of other formats using the same database interface. You can perform queries, copy objects to and from workfiles and other databases, rename and delete objects within the database, add databases to your search path, and use EViews' name aliasing features, all without worrying about how the data are stored.

When copying objects, EViews preserves not only the data itself, but as much as possible of any date information and documentation associated with the object. Missing values are translated automatically.

To Convert Or Not To Convert?

Although EViews allows you to work with foreign files in their native format, in some cases you may be better off translating the entire foreign file into EViews format. If necessary, you can then translate the entire file back again when your work is complete. EViews native databases have been designed to support a certain set of operations efficiently, and while

access to foreign formats has been kept as fast as possible, in some cases there will be substantial differences in performance depending on the format in use.

One significant difference is the time taken to search for objects using keywords in the description field. If the data are in EViews format, EViews can typically query databases containing tens of thousands of series in a couple of seconds. When working with other formats, you may find that this same operation takes much longer, with the time increasing substantially as the database grows.

On the other hand, keeping the data in the foreign format may allow you to move between a number of applications without having to retranslate the file. This minimizes the number of copies of the data you have available, which may make the data easier to update and maintain.

Using EViews, you can either translate your data or work with your data directly in the foreign format. You should choose between the two based on your particular needs.

Opening a Foreign Database

Working with foreign formats requires very little additional knowledge. To open a foreign database, simply select **File/Open/Database…** from the main menu to open the dialog. In the field **Database/File Type:** select the type of the foreign database or file you wish to open. If the database is a local file, you can then use the **Browse Files** button to locate the database in exactly the same way as for a native EViews database. You can create a new foreign format database by a similar procedure way using **File/New/Database…** from the main EViews menu.

If the database is accessed through a client-server model, selecting the dialog will change to show extra fields necessary for making the connection to the server. For example, when accessing a database located on a FAME server, the dialog will include fields for the FAME server, username and password.

Since access to a server requires many fields to be entered, you may wish to save this information as an entry in the database registry (see for details).

There are special issues relating to working with DRIPro links. See for details.

You can also create and open foreign format files using the dbopen or dbcreate commands. You may either use an option to specify the foreign type explicitly, or let EViews determine the type using the file extension. See and in the *Command and Programming Reference* for details.
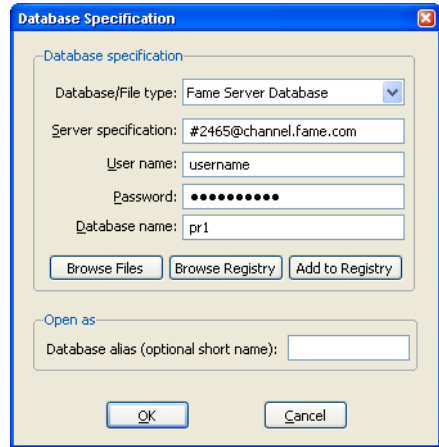
Copying a Foreign Database

Once you have opened a window to a foreign database, you can copy the entire database into a new format using **Proc/Copy the Database** from the database menus. A dialog will appear which allows you to specify the type and other attributes of the new database you would like to create.

When performing a database copy to a new format, objects which cannot be copied due to incompatibility between formats will result in error messages in the EViews command window but will not halt the copying process. Upon completion, a message in the status line reports how many objects could not be copied.

Notes on Particular Formats

AREMOS

EViews allows you to open and access data in AREMOS data banks using the standard EViews database interface. Bear in mind that there are a few limitations when working with AREMOS data banks:

- EViews can only translate series, matrix and list objects. AREMOS models and equations will be displayed in a database query, but cannot be imported into EViews. AREMOS list objects are translated into string objects inside EViews, with AREMOS comma delimited lists translated into EViews space delimited lists so that they can easily be used in EViews for loops and with the EViews "word" manipulation functions that begin with the "@w").

- EViews does not support the following AREMOS frequencies: two-monthly, lunar, periodic and daily with non-consecutive days of the week. All other frequencies should be supported. Series with unsupported frequencies will be read into EViews as "undated" series. Note that series containing weekly data should be translated correctly, but will be labeled slightly differently in AREMOS and EViews since EViews labels each observation with the first day of the week and AREMOS labels each observation with the last day of the week. (You can use the `@enddate` series in EViews to display the same dates that AREMOS would use for each observation).

- EViews cannot handle databanks that contain multiple objects with the same name and version (but different object types). Queries that return more than one object with the same name and version cause EViews to error, and fetching objects with one of these shared names is unreliable.

- EViews does not add/hide default versions to the end of names based on the working frequency in the same way that AREMOS does. AREMOS versions are simply treated as part of the object name. Since the dot character is illegal in EViews object names, this means that most objects in an AREMOS databank have illegal names. There are a variety of mechanisms in EViews to help users to work with objects with illegal names which will automatically come into play when working with AREMOS databanks. (The default outcome is that the dot that separates the name and version in Aremos will become an underscore when the object is moved into an EViews workfile or database).

Note also that there are special rules for handling AREMOS frequencies when fetching AREMOS data into EViews. See "Special Handling of Frequencies" on page 380 for discussion.

AREMOS TSD

The TSD format is a portable ASCII file format written by the AREMOS package. The EViews database tools provide an intuitive graphical interface to the data, and allows you to move data directly in and out of an EViews database without having to move the data through a workfile (which may force the data to be converted to a single frequency).

Bloomberg

EViews Enterprise Edition users may use the standard EViews database interface to search, browse, and retrieve Bloomberg Reference Data Services extensive library market and economic data.

You can search through Bloomberg securities and fields from within EViews to quickly locate the data you need, then fetch the data directly into an EViews workfile without the need for intermediate files or formats.

To access Bloomberg data from EViews, you must have:

1. The Bloomberg Professional application software installed on your machine

2. A currently active subscription to Bloomberg and be able to log into the Bloomberg Professional terminal on your machine.

In addition, Bloomberg connectivity requires the EViews Enterprise Edition.

Note that the highest resolution of data that can be fetched by EViews is one minute interval data. EViews does not currently support reading tick data for a security. Further, the Bloomberg database extension for EViews that powers the connection only supports Bloomberg Desktop API configurations (Bloomberg Server API and B-Pipe configurations are not supported).

For detail on Bloomberg Reference Data Services please see the Bloomberg website http://www.bloomberg.com/enterprise/data/reference-data-services.

The following discussion offers a quick introduction to use EViews to connect with Bloomberg data. See our whitepaper: *Bloomberg Database Extension for EViews* (available from our website www.eviews.com/download/download.html) for additional detail and discussion.

To start, you must open a database window to Bloomberg by selecting **File/Open Database…** from the main EViews menus, then selecting **Bloomberg Database** from the **Database/File type** dropdown menu. The following dialog should appear:

In most cases, you should simply be able to click on **OK** to get started. Clicking on **OK** should bring up a Bloomberg database window.
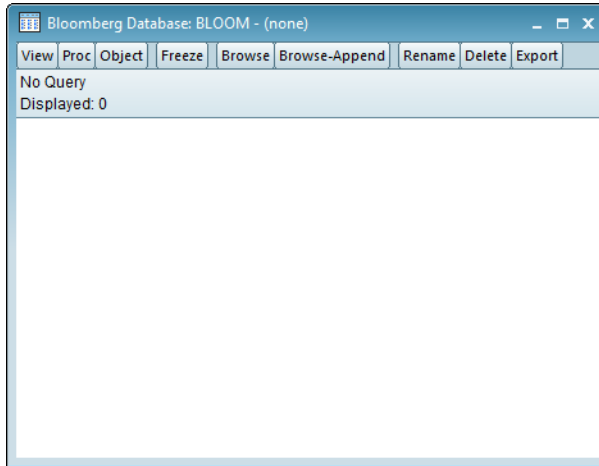
The database window represents a connection to Bloomberg data.

The first step in retrieving data is to find a series that we would like to fetch. Click on **Browse** to bring up the Bloomberg database custom search dialog:
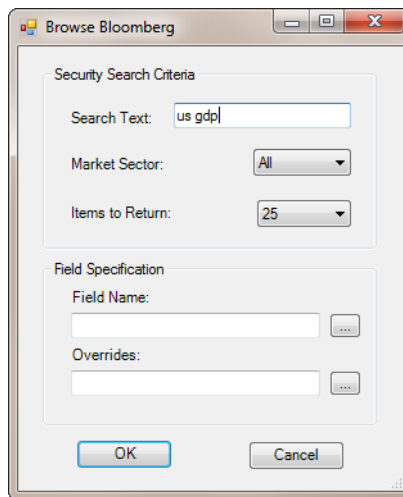
- The **Search Text** field allows you to enter one or more keywords to search for. The 'Market Sector' field lets you restrict the search to a particular Bloomberg market sector. (If you are using a Bloomberg keyboard you can change the market sector field by pressing a Yellow Key).

- The **Items to Return** field specifies how many items to return from the search. Bloomberg is very effective at selecting relevant results so only a small number of results is generally needed.

- The **Field Name** and **Overrides** boxes allow you to specify what particular information about a security you would like to fetch. By default EViews will fetch the field "PX_LAST" without any overrides.

For example, we can simply type "us gdp" into the **Search Text** box and press **OK**. You should return to the database window with the following search results displayed:
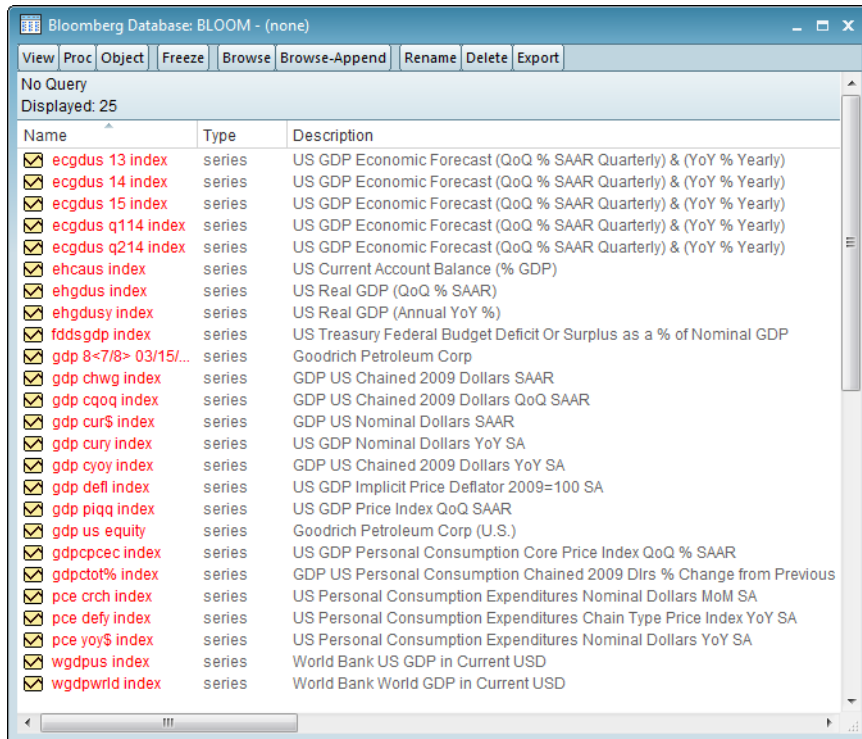
Any series displayed in the database may be manipulated following the usual EViews interface conventions:

- Double click on a series to bring up detailed information on the series

- Drag and drop series or copy and paste them into an EViews workfile or database to fetch the current data from the server (drag with the right mouse button or use Paste Special to create links)

Limitations

Note that the highest resolution of data that can be fetched by EViews is one minute interval data. EViews does not currently support reading tick data for a security.

Further, the Bloomberg database extension for EViews that powers the connection only supports Bloomberg Desktop API configurations (Bloomberg Server API and B-Pipe configurations are not supported).

Bureau of Labor Statistics (BLS)

The BLS database is a large collection of free and publicly available economic data provided by the Bureau of Labor Statistics of the U.S. Department of Labor.

EViews 10 offers a custom interface to BLS data. To open the BLS database, select **File/Open Database…** from the main EViews menu, then select **Bureau Of Labor Statistics** from the **Database/File type** dropdown menu:
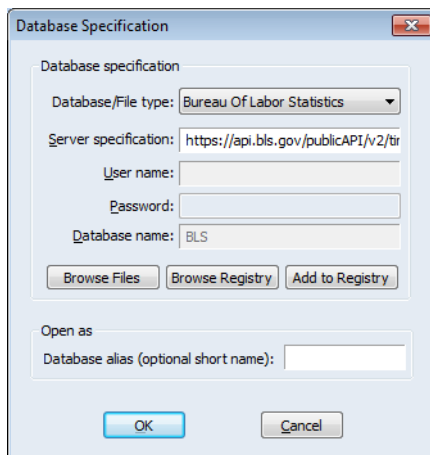
The server specification will be automatically filled in with the BLS server address https://api.bls.gov/publicAPI/v2/timeseries/data/, which you should not need to change. Click on **OK** to proceed.

Use of the BLS service is free but requires registration. The first time you attempt to open the database you will be asked to enter an API Key obtained from the Bureau of Labor Statistics:
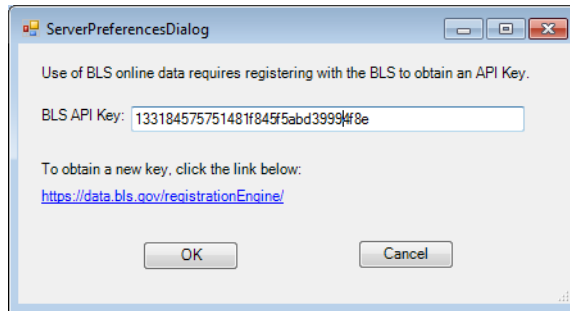
Please enter your API key and then click on OK. The key will be saved as a user specific set-ting in your EViews ".ini" file. If you need to change the key at a later time, select **View/ Preferences…** from the EViews database menu to modify your settings.

When you click on **OK**, EViews will open a standard database window:
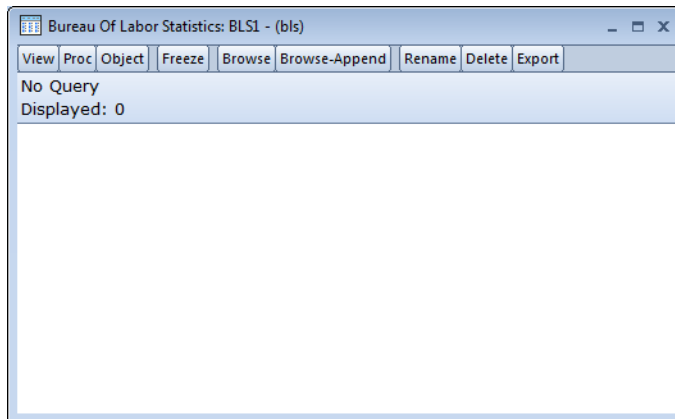
Click on **Browse** to open the custom BLS window. The standard database window will be minimized and the BLS window will open:
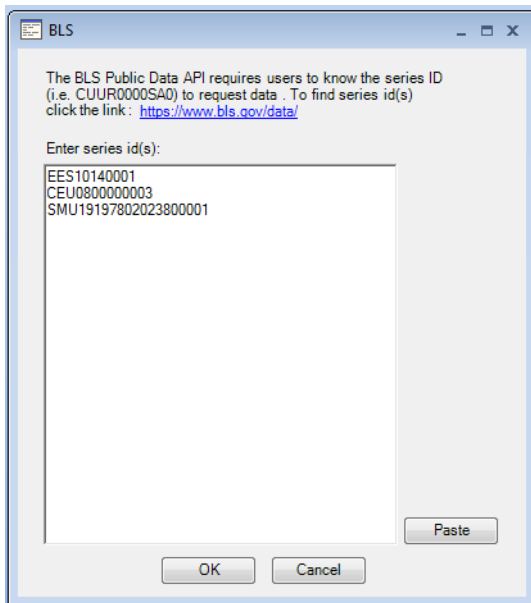
The browser window displays an edit box where the user can type or **Paste** one or more series IDs to be retrieved from BLS. In the example above the edit box contains three typed series. After clicking **OK**, the regular EViews database window will be restored and will contain any requested series.

Please note that the BLS Data API requires users to know the series ID to request data.
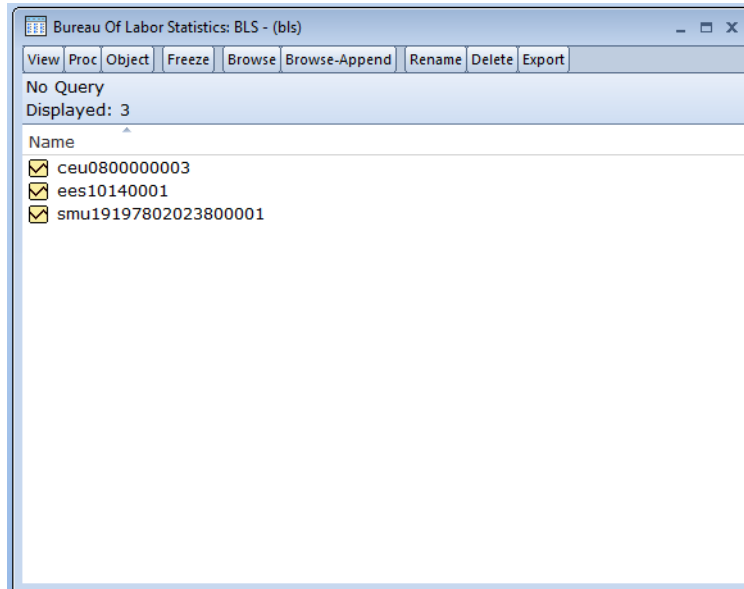
Once you have found the series of interest, you may drag-and-drop (or copy-and-paste) the series directly into an EViews workfile.

CEIC

The CEIC database includes economic, sector and financial data. Data are fetched from the internet by Internet Securities, Inc. (also known as IS Emerging Markets) XML based data servers.

CEIC database access is only available in the Enterprise Edition of EViews.

To access CEIC data, the CEIC-EViews Interface Software must already be installed on the client machine. To obtain the software and a valid CEIC account, please contact Internet Securities, Inc. directly. For additional information, please see http://www.securities.com.

To access a CEIC database, select **File/Open/Database...** and select **CEIC Database** in the drop-down menu and fill out your **User name** and **Password** as prompted. The Sever specification is pre-filled with the CEIC default, but you may enter the specification information for a different server if desired:
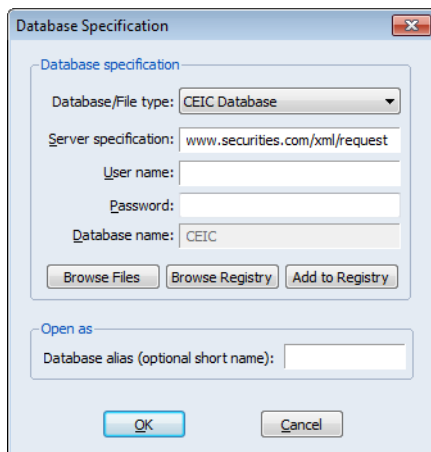
You may optionally provide a **Database alias**. Click on **OK** to open the database in the standard EViews interface.

Datastream

The Datastream database allows you to fetch data remotely over the internet from Datastream's extensive collection of financial and economic data. Data are retrieved from the Thomson Financial Datastream historical XML API. The location of the XML API must be entered in the server specification of the open database dialog window.

Datastream database access requires the Enterprise Edition of EViews.

In addition, you must be able to access to the Thomson Datafeeds service using Datastream Advance through an internet connection, and you must have a valid XML API account from Thomson Financial. (Note that the XML API user name is not the Datastream user name as used with thick client products such as Datastream Advance.)

Please contact Thomson Reuters for further information (http://www.thomsonreuters.com).

DRIPro Link

A DRIPro link is a special type of database which allows you to fetch data remotely over the internet from DRI's extensive collection of economic data. To access these data, you must have a valid DRIPro account with DRI, and you must have already installed and configured the DRIPro server software.

(Note: the DRIPro Link interface has been replaced by the newer IHS Global Insight Database Browser; see "IHS Global Insight" on page 378.)

There are special issues involved with using DRIPro links, which are discussed in detail in "Working with DRIPro Links" on page 392.

DRIBase Database

The DRIBase system is a client server system used by DRI to provide databases at the client site which can be kept current by remote updates. Customers can also use DRIBase as a means of storing their own databases in an Sybase or Microsoft SQL Server system.

(Note: the DRIBase interface has been replaced by the newer IHS Global Insight Database Browser; see "IHS Global Insight" on page 378.)

DRIBase access is only available in the Enterprise Edition of EViews.

To access DRIBase databases, the TSRT library from DRI must already be installed on the client machine (this will normally be done by DRI as part of the DRIBase installation procedure).

When working with DRIBase databases, the **Server specification** field should be set to contain the DRIBase database prefix, while the **Database name** field should contain the DRIBase bank name, including the leading "@" where appropriate. Note that these fields, as well as the **Username** and **Password** fields may be case sensitive, so make sure to preserve the case of any information given to you.

A DRIBase database has slightly different handling of frequencies than most other databases supported by EViews. See "Issues with DRI Frequencies" on page 395 for details. You should also read "Dealing with Illegal Names" on page 395 for a discussion of how DRI names are automatically remapped by EViews.

For further information on DRIBase, please contact IHS Global Insight (http://www.globalinsight.com).

EcoWin

EcoWin database support provides online access to economic and financial market data from EcoWin. The EcoWin Economic and Financial databases contain global international macroeconomic and financial data from more than 100 countries and multinational aggregates. Additional databases provide access to equities information and detailed country-specific information on earnings estimates, equities, funds, fixed income, and macroeconomics. For further information on EcoWin data and software, please contact EcoWin directly (http://www.thomsonreuters.com/products_services/financial/financial_products/investment_management/research_analysis/ecowin_pro).

EcoWin database access is only available in the Enterprise Edition of EViews.

With EViews Enterprise Edition, you can open an EViews window into an online EcoWin database. This window allows browsing and text search of the series in the database, select-

ing series, and copying/exporting series into an EViews workfile or another EViews database. In addition, EViews provides a set of commands that may be used to perform tasks such as fetching a particular series from a EcoWin database.

Access to EcoWin databases within EViews Enterprise Edition requires that the EcoWin Pro software has already been installed on the local machine, and that configuration of EcoWin database access using the EcoWin Database Configuration software has already been completed outside of EViews.

Interactive Graphical Interface

To open a graphical window to an EcoWin database, you should first open **Database Specification** dialog by selecting **File/Open/ Database…** from the main EViews menu. Next, choose **EcoWin Database** in the **Database/File Type** dropdown, and enter the name of the online database as specified in the EcoWin Database Configuration software, typically "DEFAULT".

Clicking on **OK** will open an empty EViews database window. To access the EcoWin data, click on the **Query–Select** button in the database window toolbar. EViews will open a window containing a EcoWin Pro control for browsing and searching the online data. Note that it may take a bit of time to initialize the EcoWin control. Once initialized, EViews will open the **EcoWin Query** window.
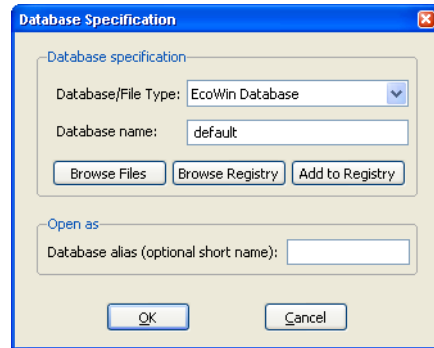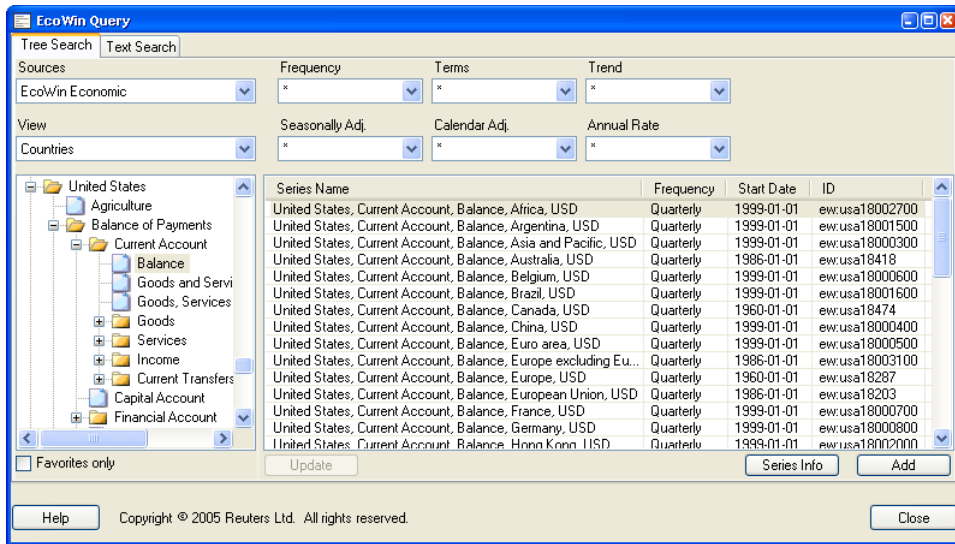
The **EcoWin Query** window provides you with two methods for selecting series to be brought into your EViews database.

First, you may use **Tree Search** to browse a directory structure of the online database. You should use the tree on the left to navigate to the directory of interest, then select series in the window on the right by clicking or control-clicking on the entry, or by clicking on the right-mouse button and choosing **Select All**. Once the desired series have been highlighted, click on **OK** to bring the selected data into your EViews database.

This procedure, first browsing to find a directory containing data of interest, selecting series, and then clicking on **OK** to bring in data, can be performed multiple times, until a list of all the series that you wish to use has been accumulated within the EViews database window. At this point the EcoWin browse control can be closed using the **Cancel** button.
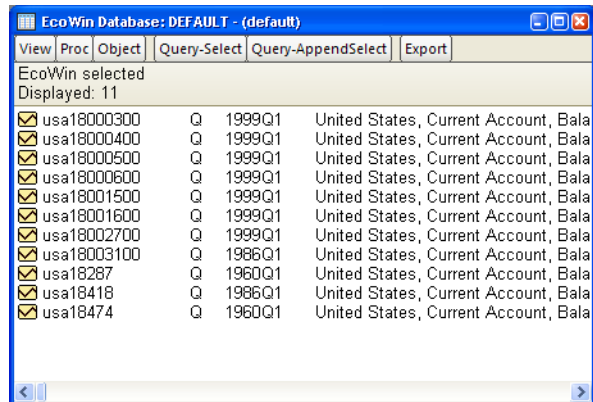
In place of browsing the tree structure of the database, you may elect to use text search to display a list of series in the database. Click on the **Text Search** selection at the top of the dialog to change the dialog to

the search display, and enter the information in the appropriate fields. For example, to search for all series in the database using the text "PETROLEUM" and "US", we have:
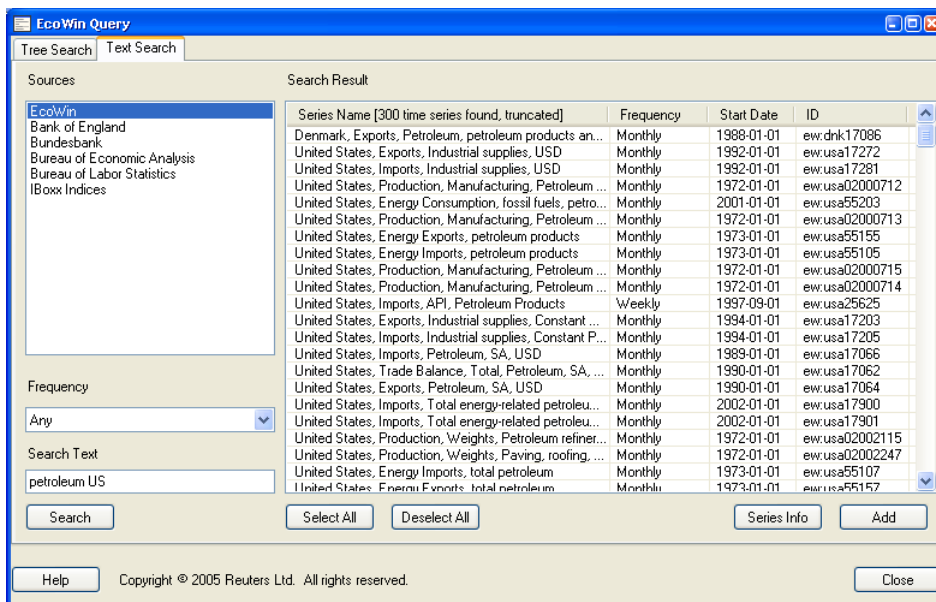
Highlight the series of interest and click on **OK** to bring them into the database. Repeat the tree browsing or search method of adding series until the list in the database is complete, then click on **Cancel** to close the query window.

Once series of interest have been included in the database window, all of the standard EViews database tools, such as copy and paste into an existing workfile or database using the right mouse menus, creating a new EViews workfile containing the data using the **Export** button, or importing data into an existing EViews workfile using the **Fetch** menu item from the workfile window, are available.

Note that after you have completed your initial query, you may reopen the EcoWin query window at any time. To add series to those already available in the database window, press the **Query Append Select** button in the database window, then browse or search for your series. To first clear the contents of the database window, you should press the **Query Select** button instead of the **Query Append Select** button.

Tips for Working with EcoWin Databases

If an EcoWin database is going to be used frequently or for direct access to individual series, you should find it useful to add an EcoWin entry in the database registry ("The Database Registry" on page 331).

The EViews database registry may be accessed by choosing **Options/Database Registry...** from the main EViews menu. Press **Add New Entry** to add a new database registry entry to the list. The procedure for adding an EcoWin database to the registry is identical to that for opening an EcoWin database. The **Database/File Type** field should be set to **EcoWin Database** and the **Database Name/Path** field should be filled with the name assigned to the database in the EcoWin Database Configuration software (generally "DEFAULT").

Once the EcoWin database has been put in the registry, it may be referred to by its alias (short hand) name. For example, if you have assigned the EcoWin database the alias "EW", you can open the database with the simple command:

        dbopen ew

or by using the **Browse Registry** button in the **Database Specification** dialog. The database name "EW" will be added to the most recently used file list, where it may be selected at a later time to reopen the database.

Assigning the EcoWin database a shorthand name also allows you to reference data without explicitly opening the database. For example, the command

        equation eq1.ls ew::usa09016 c ew:usa09016(-1) @trend

runs a regression of U.S. unemployment on an intercept, its own lagged value, and a time trend. The series USA09016 will be accessed directly from the EcoWin servers, and does not need to appear within a currently open database window for this command to be used. Other commands such as copy allow the name associated with the series to be changed during the procedure, as well as supporting the copying of series directly from an EcoWin database to another EViews database.

        show ew::usa09016

displays a table of U. S. unemployment.

Note that series in the EcoWin "Economic" or EcoWin "Financial" databases may be referenced merely by using the database shorthand and the series name. In the example above, EViews looks for USA09016 in the two base EcoWin databases.

Series located in add-on EcoWin databases such as "Bank of England," "Bundesbank," "Bureau of Economic Analysis," must also provide the name of the add-on database in which the series is located. You should provide the name of the EcoWin shortcut followed by a double colon, an EcoWin add-on database prefix, a slash, and then the series name. For example, you can fetch the mortgage rate (LUM5WTL) in the Bank of England database with

        fetch ew::boe\lum5wtl

where we follow the database name with the add-on name BOE. The series will be named "BOE\LUM5WTL" in EViews. Note that the add-on name BOE is taken from the EcoWin name prefix (for example, LUM5WTL appears as "BOE:LUM5WTL" within EcoWin.
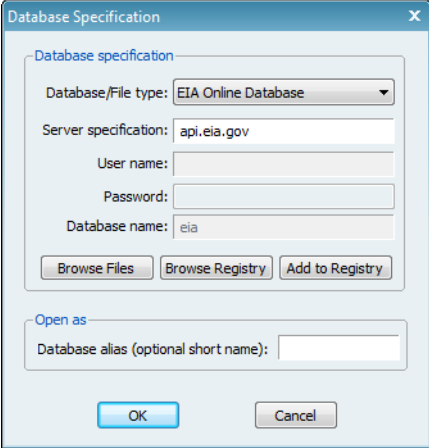
EIA (U.S. Energy Information Administration)

The EIA database is a large collection of free and publicly available energy and related economic data provided by the U.S. Energy Information Administration.

EViews Enterprise Edition supports both online access to EIA data via the EIA web server as well as offline access to data downloaded in the EIA Bulk File format.

Online EIA Database

To open the EIA online database, simply select **File/Open Database…** from the main EViews menu, then select **EIA Online Database** from the **Database/File type** dropdown menu:

The server specification will be automatically filled in with the EIA server address api.eia.gov which you should not need to change. The database will always be opened with the name "eia". Click on **OK** to proceed.

Use of the EIA online service is free but requires registration. The first time you attempt to open the EIA online database you will be asked to enter an API Key obtained from the EIA:
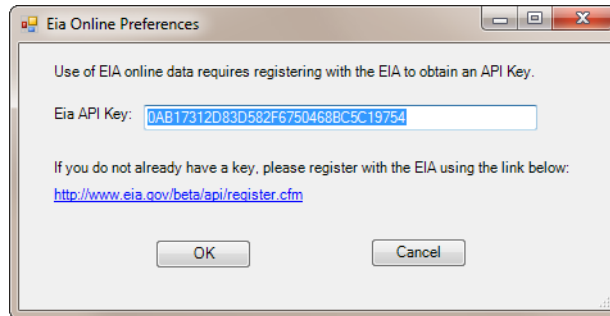
Please enter your API key then click on **OK**. The key will be saved as a user specific setting in your EViews ".ini" file. If you need to change the key at a later time, use V**iew/Preferences…** from the EViews database menus to bring up the dialog again so you can modify your settings.

When you click on **OK**, EViews will open a standard database window:
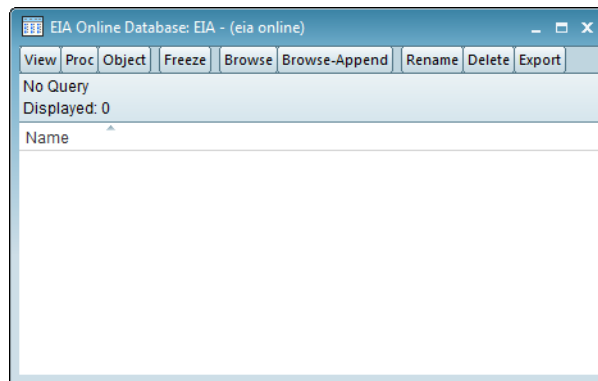
The EIA database connection supports a custom browser window for navigating through the EIA data. Click on **Browse** to open the custom browser window. The standard database window will be minimized and the EIA browser window will open:
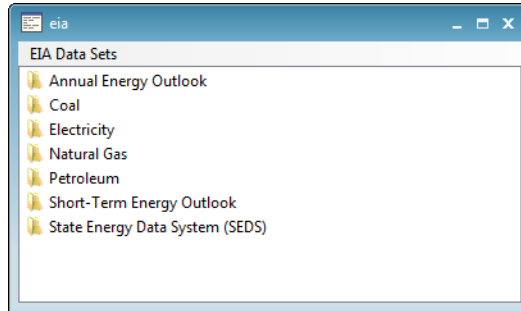
The browser window displays the data available within the database as a set of folders, where each folder may contain additional folders or a list of series available for downloading. Clicking on a folder moves you into that folder and adds the folder to the path displayed at the top of the window:
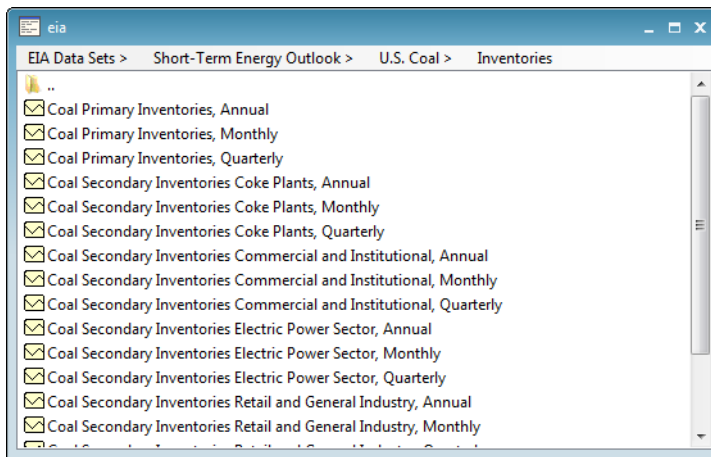
You can return to the previous folder by clicking on the ".." folder icon at the top of the list. You can also move directly to any folder within the current path by clicking on the path component at the top of the window.

Any series contained in the folder may be manipulated in the usual fashion. When you have finished working with the browser, simply close the browser window to restore the regular EViews database window.

Offline EIA Databases

EViews also allows you to work offline with data downloaded from the EIA using the EIA bulk download facility. The bulk download feature allows you to fetch all series and category information from an entire EIA data set into a single local file. Downloading an entire

bulk file is generally much more efficient than fetching a large number of series one at a time. Once the file has been downloaded, you can navigate within the dataset and bring data into EViews without reconnecting to the EIA servers.

To download a bulk file, open the browser window of the EIA online database, right click on a data set, then select **Download Bulk File…**:
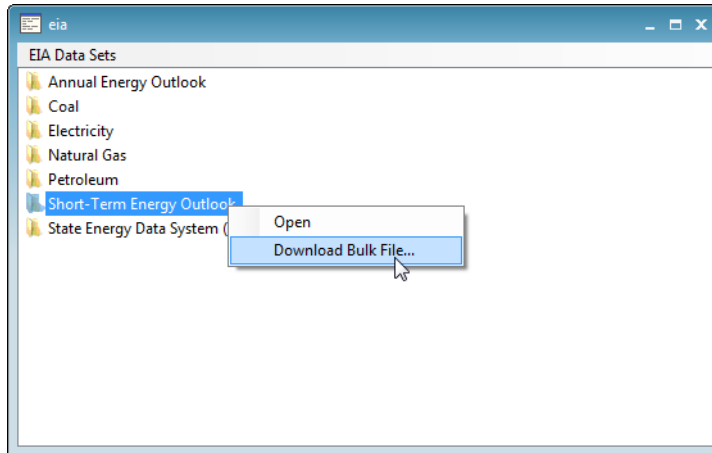
A dialog will appear giving instructions for downloading the file and preparing it for use by EViews:
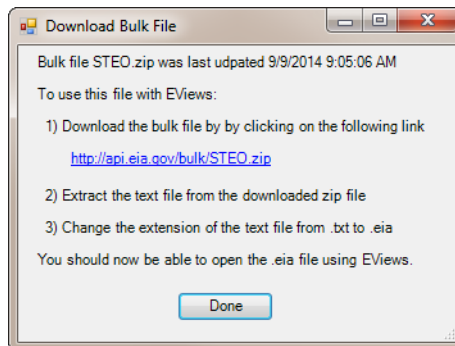
Clicking on the download link will open your default web browser and start a download of the file. The bulk file is compressed in ".zip" format. You will need to use an unzipping utility to extract the single ".txt" file contained within the zip file. Finally, rename the ".txt" file to ".eia" so EViews can tell that the file contains data in EIA bulk file format rather than a generic text file.

Once the bulk file has been prepared, use **File/Open Database…** and browse to the file or simply drag and drop the file onto EViews to open it. The caption on the window indicates that you are working with a local file. Otherwise, the interface behaves the same as when working with online data:
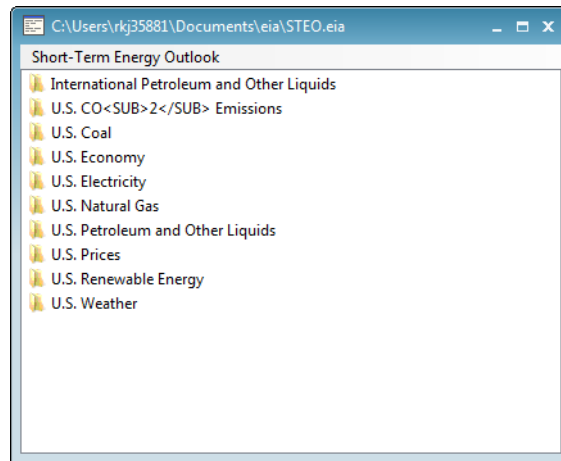
Note that some EIA data sets are large (hundreds of megabytes of uncompressed data). When working with large files you may see a delay as the file is opened since EViews must completely scan the entire bulk file to find all series and category information stored in the file. Subsequent operations should be fast since all data is stored locally on the machine.

Eurostat, ECB, UN, and IMF

SDMX Databases provide access to a list of datasets that offer a large range of publicly available data.

Both the Standard and Enterprise versions of EViews offer access to Eurostat, ECB (European Central Bank), UN (United Nations), and IMF (International Monetary Fund) data using SDMX Web Services.

Please note that an internet connection will be required to obtain SDMX online data. For more information on the datasets, please see

> http://ec.europa.eu/eurostat/data/database
> https://ec.sdw-wsrest.ecb.europa.eu
> http://data.un.org/WS/
> http://sdmxcentral.imf.org
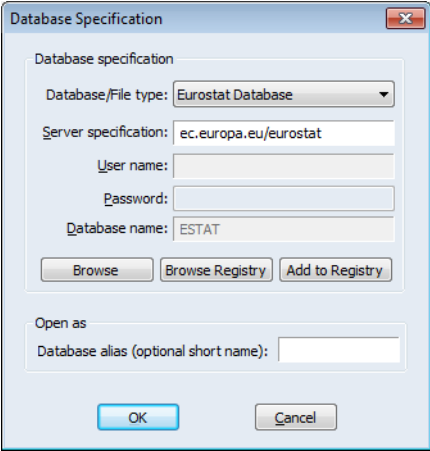
EViews offers a custom interface to the SDMX databases. The interface includes a custom browser for navigating and retrieval of available data. To start, you must open a database window to SDMX database by selecting **File/Open Database…** from the main EViews

menus, then selecting **Eurostat SDMX Database**, **ECB SDMX Database**, **UN SDMX Database,** or **IMF SDMX Database** from the **Database/File type** dropdown menu. A dialog similar to the one below will be displayed:

Click on **OK** to open the online database. You will be presented with a standard EViews database representing a connection to data. Click on **Browse** or **Browse-Append** to open a custom database.

Here we see the interface to the Eurostat data:
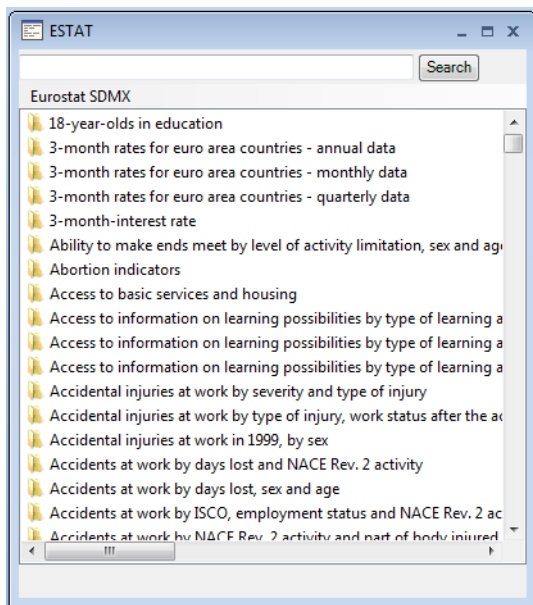
The browser allows you to find data available within datasets by navigating through a set of folders. Click on any folder to move inside it. A dialog with a list of filters will be shown, which will allow you to filter and customize the search results of the available series objects in the selected dataset.
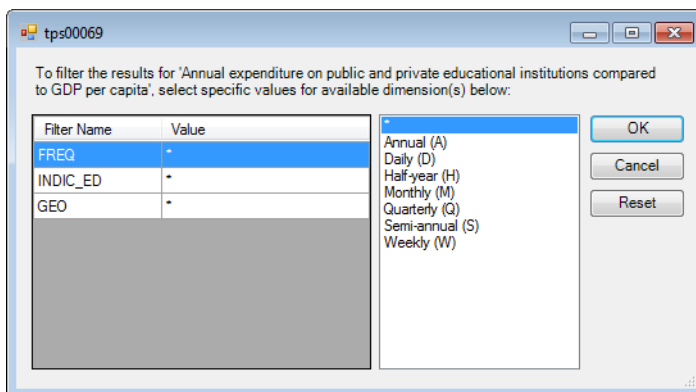
Click on **OK** to show the results.

The full path of the active folder will be shown in the header at the top of the window. You can click on any folder within the path to navigate back to that folder. Similarly, clicking on the ".." in the window listing will move up a single level.

The browser interface to the data offers a number of useful tools for working with your Eurostat database:

- The **Search** button allows searching the browse window by keyword. Simply enter a keyword into the edit box and click the **Search** button. EViews will display a list of data sets or series that match the keyword.
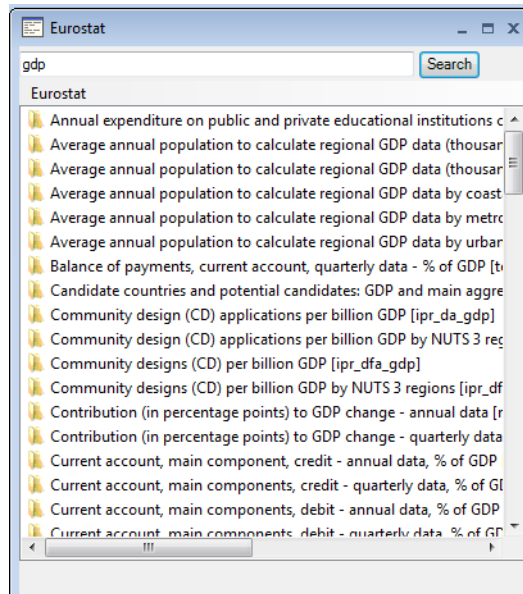
- Right-clicking on items in the browser will bring up a menu for performing useful tasks:
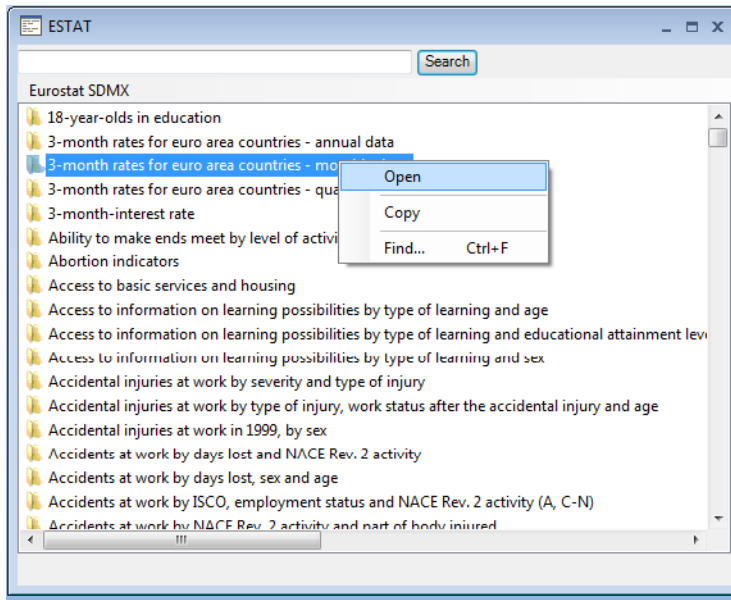
1. Open can be used to navigate inside a folder or bring a series object in the database window.

2. Copy can be used to select a series object to be copied to a workfile

3. Find allows you to do find a keyword on your window.
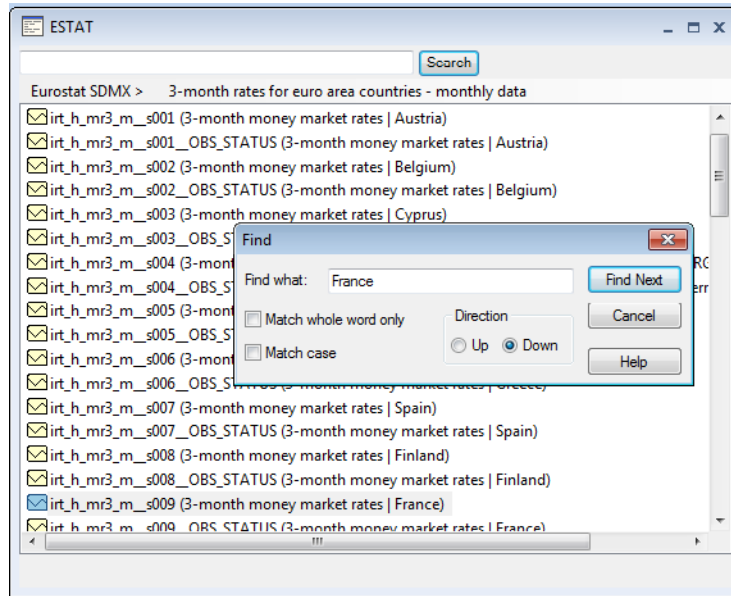
Once you have found the series of interest, you may drag-and-drop (or copy-and-paste) the series directly into an EViews workfile in the usual fashion.

FactSet

The FactSet database is a remote database provided by FactSet Research Systems. It is composed of more than 800 commercial data sets of financial data. Data include international macroeconomic, commodity, and financial time series from more than 150 countries.

Use of FactSet databases requires the Enterprise Edition of EViews and installation of FactSet client software. For further information on using Factset please contact Factset directly (http://www.factset.com).

FAME

The FAME format is a binary file format written by FAME (SunGard MarketMap Analytic Platform) database products. FAME provides a variety of products and services for working with time series data.

FAME access is only available in the Enterprise Edition of EViews.

In order to access FAME databases, a valid installation of FAME must already be available. EViews makes use of the FAME C Host Language Interface (C HLI) library, and will error unless the FAME .DLLs are correctly installed on the machine. EViews supports versions 8 and newer of the FAME libraries.

A local FAME database can have any file extension, and EViews supports access to a FAME database with any name. However, because many commands in EViews use the file extension to automatically detect the file type, you will generally find it easier to work with FAME databases which have the default ".DB" extension.

EViews also allows access to FAME databases located on a FAME Database Server. When working with a FAME server, the **Server specification** should be given in the form:

    #port_number@ip_address

For example, the server specification for access to a FAME/Channel database might appear as:

    #2552@channel.fame.com

Access to a server will require a valid username and password for that server.

Please contact SunGard directly (http://www.sungard.com/fame) for further information about the FAME database system and other FAME products.

FRED

FRED (Federal Reserve Economic Data) is a publicly accessible database of more than 20,000 U.S. and international time series of multiple frequencies, provided by the Economic Research Division of the Federal Reserve Bank of St. Louis. The FRED database offers a wide variety of *freely downloadable data*, including interest rates, exchange rates, national income accounts, and many others. Data found in FRED are drawn from several national and international sources and is updated frequently.

For more information on the FRED database, please see http://research.stlouisfed.org/fred2.

Both the Standard and Enterprise versions of EViews offer FRED access. Please note that an internet connection will be required to obtain FRED data.

EViews offers a custom interface to the FRED data service provided by the Federal Reserve Bank of St Louis. The interface includes a custom browser for navigating the available FREDX data and adds support for retrieval of historical releases. To access the FRED database, use **File/Open Database…** from the main EViews menus and choose "FRED Database":
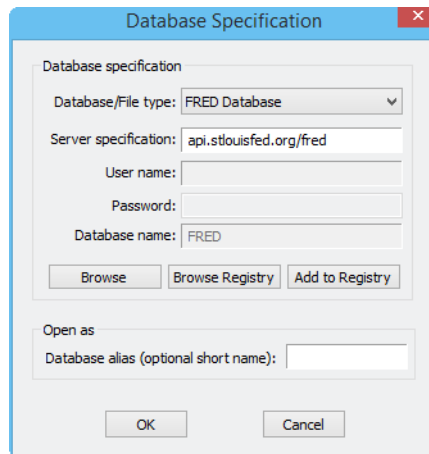
Click on **OK** to open the online database. You will be presented with a standard EViews database window with the **Query** and **Easy Query** search interface. Click on **Browse** or **Browse–Append**, which will open an all new custom FRED database. Click on the folder icons to drill down into subtopics:
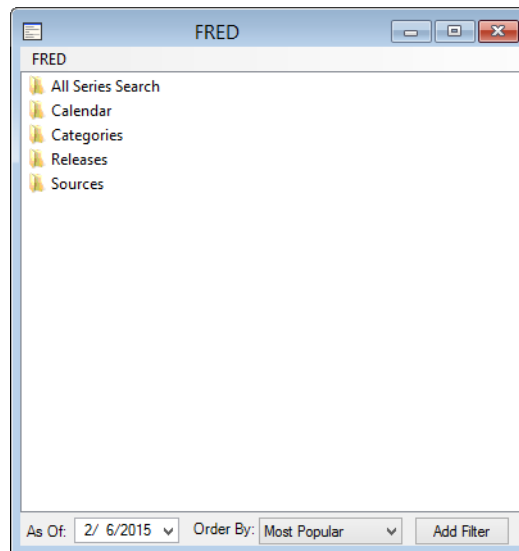
The browser allows you to find data available within FRED by navigating through a set of nested folders. Simply click on any folder to move inside it.
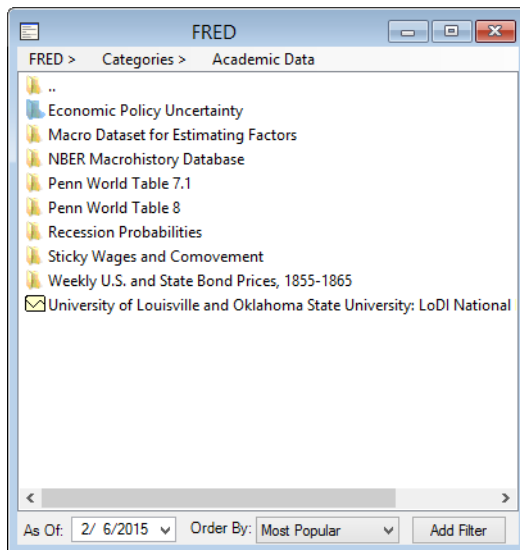
374—Chapter 10. EViews Databases

The full path of the active folder will be shown in the header at the top of the window. You can click on any folder within the path to navigate back to that folder. Here, clicking on **Categories** will move back up a single level to show the category choices. Similarly, clicking on the "." in the window listing will move up a single level.

Once you have found the series of interest, you may drag-and-drop (or copy-and-paste) the series directly into an EViews workfile in the usual fashion.

The browser interface to FRED offers a number of useful tools for working with your FREDX database:

- Clicking on the **All Series Search** folder in the FRED root directory changes the interface to allow searching by keyword. Simply enter one or more keywords into the Search For box and EViews will display a list of series that match those keywords.
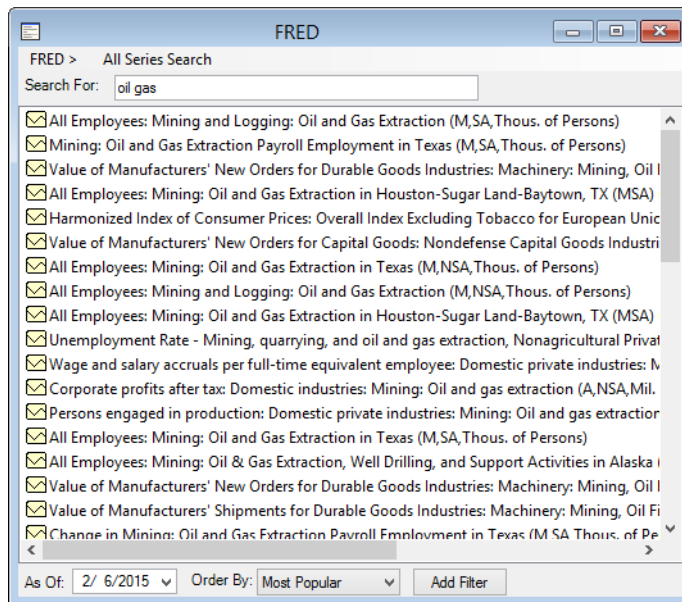
- You can use the **Order By** combo to change the order in which series are listed. By default, EViews will show the **Most Popular** items first, but you can order by alphabetical **Description** order, **Recently Updated**, **Earliest Values**, **Latest Values**, **Highest Frequency**, and **Lowest Frequency**.

- You may click on the **Add Filter** button to narrow the display list. Filters will be displayed at the top of the window and may be modified, disabled, or removed by clicking on the filter description in the header.

- You may use the **As Of** dropdown calendar to restrict the data to a particular vintage, which is handy when trying to match a particular data revision.

- Right-clicking on items in the browser will bring up a menu with items for performing useful tasks:
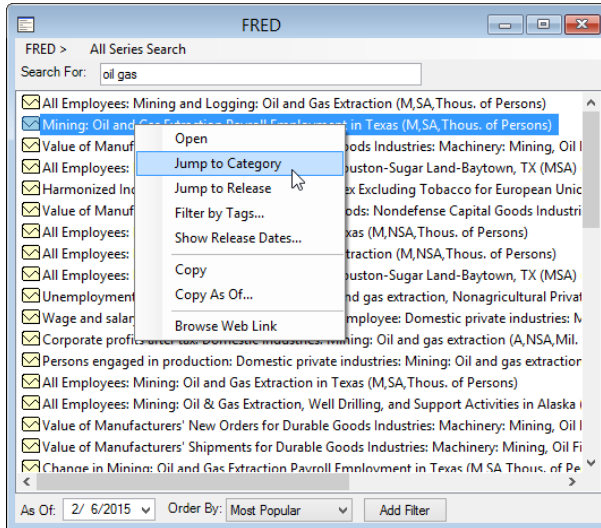
1. **Jump to Category** can be used to navigate directly to the folder within the category tree that contains the selected series. If the series appears in more than one category, a list will appear for you to choose which of the folders you would like to navigate to.

2. **Jump to Release** can be used to navigate directly to the statistical release that contains the selected series.

3. **Filter by tags** allows you to see the list of tags attached to a particular series. Tags are standardized concepts that may be used to filter which series are displayed within the browser window. You may check or uncheck the tags of a series to adjust the filtering currently in effect:

4. The **Show Release Dates** menu item displays a list of release dates for the selected series. You may use this information in conjunction with the **As Of** dropdown to retrieve the historical version of a series.

5. The **Copy As Of** right mouse button menu item allows you to retrieve multiple historical revisions at once. Simply select one or more series in the browser window, then click on **Copy As Of** to see a list of available releases. Select the releases you would like to copy by clicking on them then click on the **Copy Selected Releases** button at the bottom of the screen.
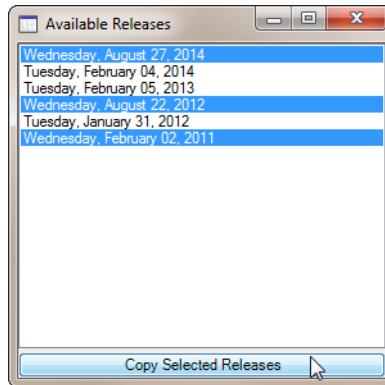
GiveWin/PcGive

The GiveWin/PcGive format is a binary file format used by GiveWin, PcGive versions 7 through 10, and PcFiml.

There are two issues when working with GiveWin/PcGive files. The first is that EViews is case insensitive when working with object names, while GiveWin and PcGive are case sensitive. Because of this, if you intend to work with a file in both packages, you should avoid having two objects with names distinguished only by case. If your files do not follow this rule, EViews will only be able to read the last of the objects with the same name. Any early objects will be invisible.

The second issue concerns files with mixed frequency. The GiveWin/PcGive file format does support series of mixed frequency, and EViews will write to these files accordingly. However, GiveWin itself appears to only allow you to read series from one frequency at a time, and will ignore (with error messages) any series which do not conform to the chosen frequency. Consequently, depending on your application, you may prefer to store series of only one frequency per GiveWin/PcGive file.

Haver

The Haver database format is a binary file format used by Haver Analytics when distributing data.

Direct access to Haver database files is only available in the Enterprise Edition of EViews.

The main difference between Haver databases and other file formats supported by EViews is that Haver databases are read-only. You cannot create your own database in Haver format, nor can you modify an existing database. EViews will error if you try to do so.

Please contact Haver Analytics (http://www.haver.com) directly for further information about Haver Analytics data products.

IHS Global Insight

The IHS Global Insight Database is a large and frequently updated international data offering for over 200 global markets. The Global Insight Database offers a modern interface for accessing the same data offered through the IHS Magellan interface.

To access the Global Insight Database you must have a valid IHS MyInsight username and password, which can be requested from the IHS webpage (http://www.ihs.com/products/global-insight/data-analytics/index.aspx).

Note that access is only available in the Enterprise Edition of EViews.

To open a connection to the IHS Global Insight Database, select **File/Open Database…** from the main EViews menu, then select **IHS Global Insight Database** from the **Database/File type** dropdown menu:
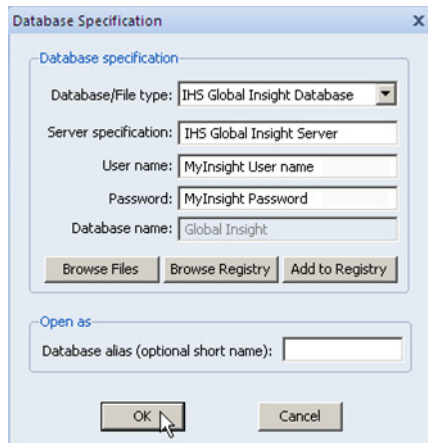
EViews will launch the Global Insight Database Browser, allowing you to search the database of series available from IHS:
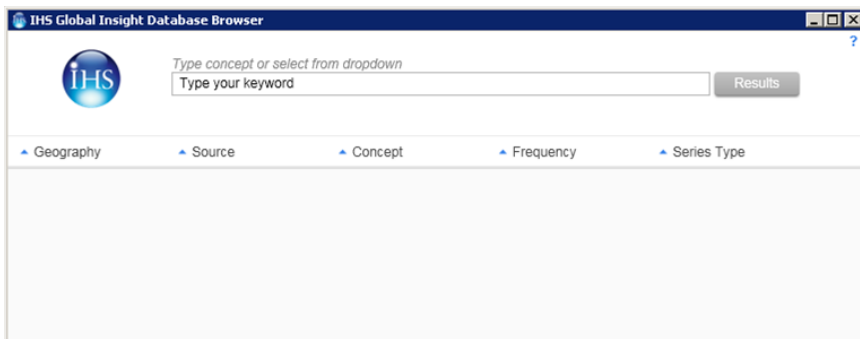
For more information see the PDF guide to using the IHS Global Insight Database Browser from within EViews, available from our website: www.eviews.com/download/download.html.

IHS Magellan

You may use EViews to access IHS Global Insight data stored in Magellan data banks Please note that an internet connection and a MyInsight account will be required to obtain IHS Magellan data.

(Note: the IHS Magellan interface has largely been replaced by the newer IHS Global Insight Database Browser; see "IHS Global Insight" on page 378.)

Bank Names

There are two ways of specifying a particular Magellan series in an EViews command:

- using series type (*e.g.* "gdp.q.hist")
- using bankname (*e.g.* "us\gdp.q")

Both of these identifiers will work *no matter what bank name was specified when opening the database.*

The bank name specified when opening a database only affects what happens when you use an incomplete series identifier (containing neither a series type nor a bank name).

When you specify a particular bankname when opening a database, this bankname is used as the *default* bankname for all operations in the database. This means that whenever you specify a series name without a leading bankname, this default bankname will be added to the front of the name. So, if you open a database with bank "US":

- "fetch gdp.q" will be interpreted as "fetch us\gdp.q".
- doing an easy query for "name matches gdp*" will be interpreted as an easy query for "name matches us\gdp*".

When you specify the special "all banks" keyword when opening a database ("IGI") then there is no default bank. In this case a bank name is never added to any series names, but the series type ".hist" will be added whenever a series type is required but has not been specified in a name. So

- "fetch gdp.q" will be interpreted as "fetch gdp.q.hist".

Note that in either case, you can always override the bankname specified during database open in Easy Query by adding an explicit "*bankname*\" to the beginning of the name. Thus, name matches "us*" and description matches "iron steel" will produce the same results *no matter the default bankname.*

Special Handling of Mnemonics

EViews recognizes mnemonics in IHS Magellan databases that have illegal EViews characters ("@," "&," *etc.*) and will, if possible, prompt to rename the series before transferring it to your active workfile. The default replacement for illegal characters is an underscore. If you are using a command to fetch data, EViews will automatically replace an illegal character with an underscore.

Note that EViews drops the frequency identifier when a series is pulled into the workfile if the frequency of the series fetched matches the workfile frequency. If the original series does not match the workfile frequency, EViews will note this fact and convert the series to the workfile page frequency using the series defaults and prompt you for a name change.

For example, if you fetch the Magellan series GDP.Q into an annual frequency page of a workfile, EViews will convert the data in GDP.Q to an annual frequency (*i.e.*, averaging the four quarters, if average is the default for high to low frequency) and it will prompt for a series name since "GDP.Q" is not a legal EViews name. The prompt will suggest the name "GDP_Q," but you will have the option to provide different name. If instead you fetch GDP.A into the annual-frequency workfile, EViews will automatically drop the ".A," leaving the series GDP in the workfile.

(Note that while this method of fetching data will provide you with a quarterly series from one that is reported at a monthly frequency, an better approach might be to bring the monthly series into a monthly page, then link the data into a quarterly page.)

Special Handling of Frequencies

There are two rules to make handling of Magellan (and AREMOS) frequencies easier in EViews:

1. When fetching into a workfile, if the Magellan series identifier does not contain an explicit *version/frequency*, EViews will add a *version/frequency* based on the frequency of the destination workfile.

2. When fetching into a workfile, if the Magellan *version/frequency* matches the version/frequency of the workfile being fetched into, then EViews will automatically trim the version/frequency from the name.

Examples:

- "fetch gdp" into a quarterly workfile will try to fetch "gdp.q" (using rule 1) and then discard the version ".q" from the name (using rule 2) so that the series appears in the workfile as "gdp."

- "fetch gdp.q" into a quarterly workfile will also discard the version ".q" from the name (using rule 2) so that the series appears as "gdp."

- "fetch gdp.m" into a quarterly workfile will not match either rule, so the series will keep the illegal name "gdp.m" which EViews will suggest renaming to "gdp_m."

- copying a series from a Magellan database into an EViews database will never use either rule (because EViews databases do not have a default frequency).

Running EViews from a Network Folder

When running EViews from a folder on a local network, the IHS Magellan database extension may fail to load properly due to a restrictive Runtime Security Policy for .NET assemblies. By default, the .NET Framework does not allow the use of .NET assemblies that are located on a network folder. In order for this extension to work properly with EViews, you must grant FullTrust permissions to the EViews network folder.

To do this, go to a computer that will run EViews:

1. If you haven't already done so, run EViews on that machine. If the **Register Components** dialog does not appear automatically upon startup, run the `regcomponents` command to bring it up manually. Verify that all required components have been registered properly on this machine. Click the **Yes** button to register (or re-register) and wait for confirmation of success.

2. Shutdown EViews (if it is already running).

3. In Control Panel, open Administrative Tools.

4. Run "Microsoft .NET Framework 2.0 Configuration."

5. In the left tree-view, expand the node ".NET Framework 2.0 Configuration/My Computer/Runtime Security Policy/Machine/Code Groups/All_Code."

6. Under "All_Code," right-click "LocalIntranet_Zone," then click on **New**.

7. Enter a name and description (*e.g.* "EViews Network Share"). Click **Next**.

8. In the "Choose the condition type for this code group" list, click "URL."

9. In the "URL" box, type in the full path to your EViews network file share followed by an asterisk to allow any extensions in that folder to run with the new permissions (*e.g.* "\\servername\folder\EViews 9*"). Click **Next**.

10. Select "Use existing permission set" and then select "FullTrust" from the dropdown list. Click **Next**.

11. Click **Finish**.

12. Restart EViews from the network folder to obtain the new permissions. Try opening a database using the IHS Magellan database extension to verify it loads properly.

You will have to repeat this procedure on each computer that will run EViews in order to get the IHS Magellan database extension working properly from the network.

Moody's Economy.com

The Moody's Economy.com format is a binary file format written by Moody's Economy.com tools. Moody s Economy.com features a database containing more than 190 million economic, financial, and demographic time series covering more than 180 countries and their sub-regions.

The Enterprise Edition of EViews is required to read Moody's Economy.com formatted data.

Additional information on Moody's Economy.com databases can be found at http://www.economy.com.

RATS 4.x

The RATS 4.x format is a binary file format used by RATS Version 4 through 7 on all platforms.

The main issue to be aware of when working with RATS 4.x format files is that the ".RAT" extension is also used by RATS version 3 files. EViews will neither read from nor write to RATS files in this earlier format. If you try to use EViews to open one of these files, EViews will error, giving you a message that the file has a version number which is not supported.

To work with a RATS Version 3 file in EViews, you will first have to use RATS to translate the file to the Version 4 format. To convert a Version 3 file to a Version 4 file, simply load the file into RATS and modify it in some way. When you save the file, RATS will ask you whether you would like to translate the file into the new format. One simple way to modify the file without actually changing the data is to rename a series in the file to the name which it already has. For example, if we have a Version 3 file called "OLDFILE.RAT", we can convert to a Version 4 by first opening the file for editing in RATS:

```
dedit oldfile.rat
```

then listing the series contained in the file:

```
catalog
```

then renaming one of the series (say "X") to its existing name

```
rename x x
```

and finally saving the file

```
save
```

At this point, you will be prompted whether you would like to translate the file into the Version 4 format.

See the RATS documentation for details.

RATS Portable

The RATS portable format is an ASCII file format which can be read and written by RATS. It is generally slower to work with than RATS native format, but the files are human readable and can be modified using a text editor.

You can read the contents of a RATS portable file into memory in RATS with the following commands:

```
open data filename.trl
data(format=portable) start end list_of_series
close data
```

To write what is currently in memory in RATS to a RATS portable file, use:

```
open copy filename.trl
copy(format=portable) start end list_of_series
close copy
```

See the RATS documentation for details.

SDMX-ML Files

SDMX (Statistical Data and Metadata eXchange) standardizes the processes for the exchange of statistical data among international organizations. EViews offers an interface to SDMX data where the user has the ability to read SDMX datasets and data structure definitions (DSD) in structured XML documents (SDMX-ML), using the standard EViews database interface. EViews allows you to work offline with data from organizations like EUROSTATA, ECB, IMF, UN, BIS, and OECD. Both the Standard and Enterprise versions of EViews offer reading of SDMX data. Please note that the EViews SDMX-ML interface requires that data were previously downloaded in SDMX XML format.

To start, you must open a database window to SDMX-ML by selecting **File/Open Data-base…** from the main EViews menus, then selecting **SDMX-ML files** from the **Database/File type** dropdown menu. The following dialog should appear:
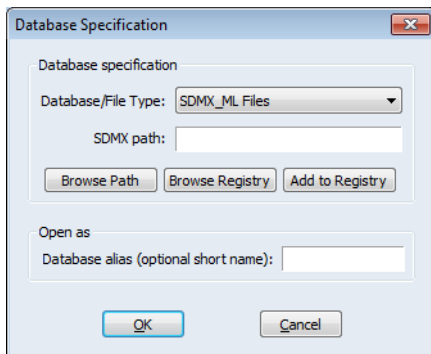
Click on **Browse Path** to select the folder/path on disk where the SDMX data and data defi-nitions files are located.
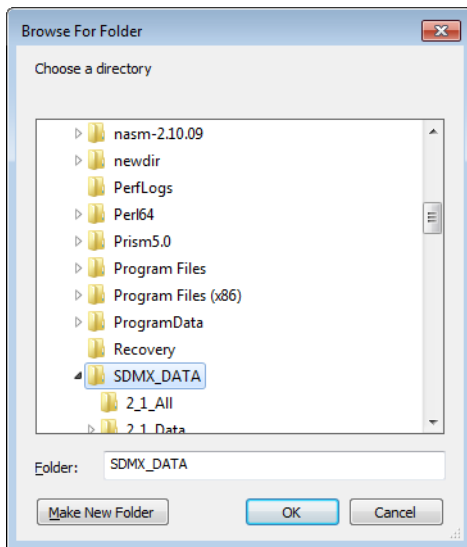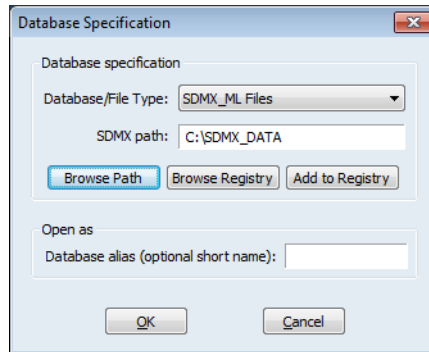
Click on **OK** to open the database. You will be presented with a standard EViews database with the **All**, **Query** and **Easy Query** search interface, to view all or a selection of the data series:
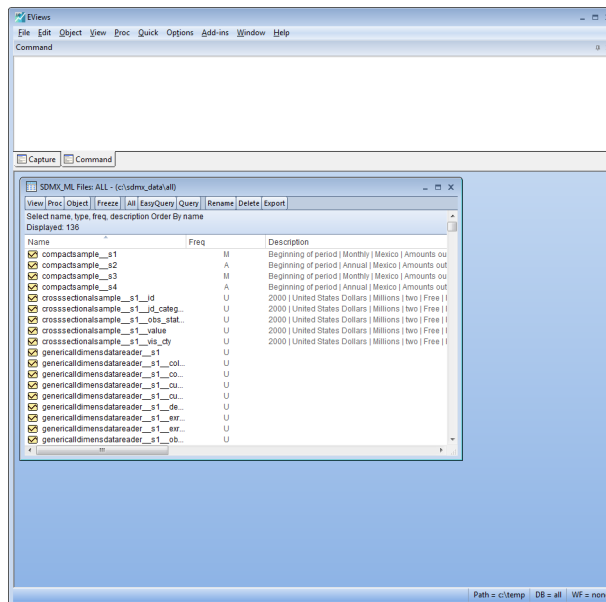
Once you have found the series of interest, you may drag-and-drop (or copy-and-paste) the series directly into an EViews workfile in the usual fashion.

TSP Portable

The TSP portable format is an ASCII file format which can be read and written by copies of TSP on all platforms. The file consists of a translation of a TSP native databank (which typi-

cally have the extension ".TLB") into a TSP program which, when executed, will regenerate the databank on the new machine.

To create a TSP portable file from a TSP databank file, use the DBCOPY command from within TSP:

```
dbcopy databank_name
```

To translate a TSP portable file back into a TSP databank file, simply execute the TSP file as a TSP program.

Once the data are in TSP databank format, you can use the TSP command,

```
in databank_name
```

to set the automatic search to use this databank and the TSP command,

```
out databank_name
```

to save any series which are created or modified back to the databank.

See the TSP documentation for details.

World Bank Data

EViews 10 offers a new custom interface to World Bank data. World Bank Open Data provides access to a list of datasets that offer access to global development data. These databases offer a wide variety of freely downloadable indicators about development in countries around the world.

For more information on the World Bank datasets, please see http://data.worldbank.org.

Both the Standard and Enterprise versions of EViews offer World Bank access. Please note that an internet connection will be required to obtain World Bank data.

EViews offers a custom interface to the World Bank. The interface includes a browser for navigating and retrieval of World Bank data. To start, you will open a database window by selecting **File/Open Database…** from the main EViews menus, then selecting **World Bank Database** from the **Database/File type** dropdown menu. The following dialog should appear:
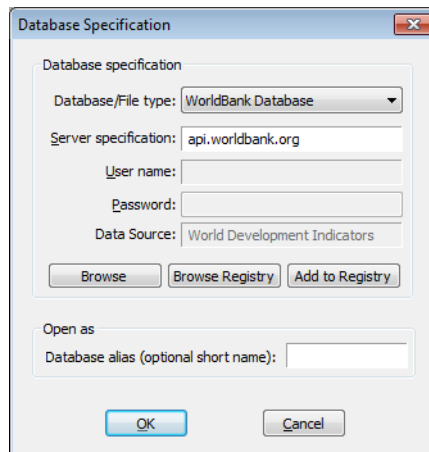
The default data source selected is World Development Indicators (a collection of development indicators that includes global development data). Click on **Browse** to view and select another data source.
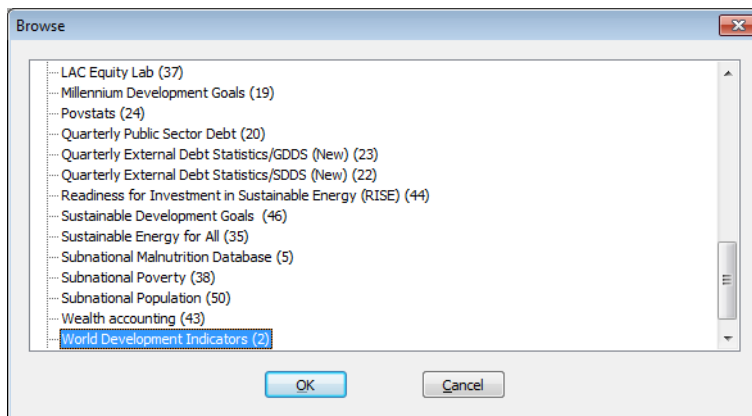
Click on **OK** to open the online database. You will be presented with a standard EViews database window.
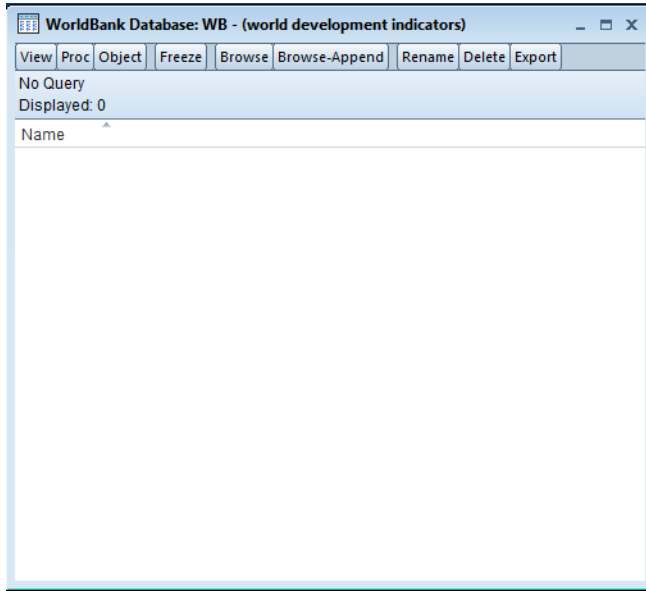
The database window represents a connection to the data. Click on **Browse** or **Browse-Append**, which opens a new custom World Bank database. Click on the folder icons to move down into subtopics:
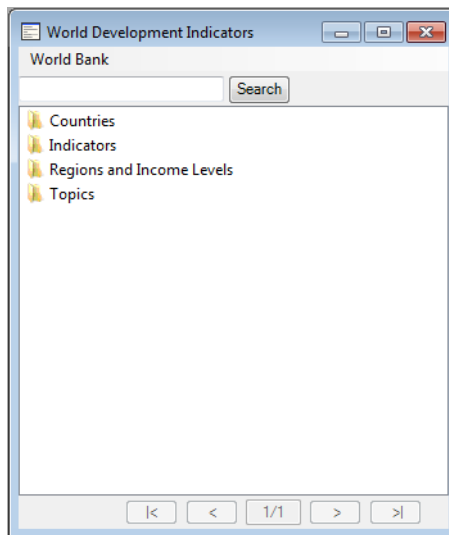
The browser allows you to find data available within the World Bank by navigating through a set of nested folders. Simply click on any folder to move inside it.
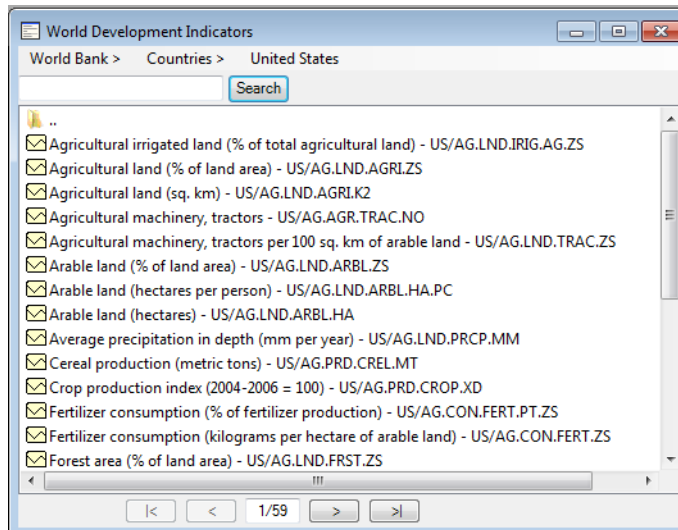
The full path of the active folder will be shown in the header at the top of the window. You can click on any folder within the path to navigate back to that folder. Here, clicking on Countries will move back up a single level to show the country choices. Similarly, clicking on the ".." in the window listing will move up a single level.

The browser interface to the World Bank offers a number of useful tools for working with your World Bank database:

- The Search button allows searching by keyword. Simply enter a keyword into the edit box and click the Search button. EViews will display a list of indicators that match the keyword.
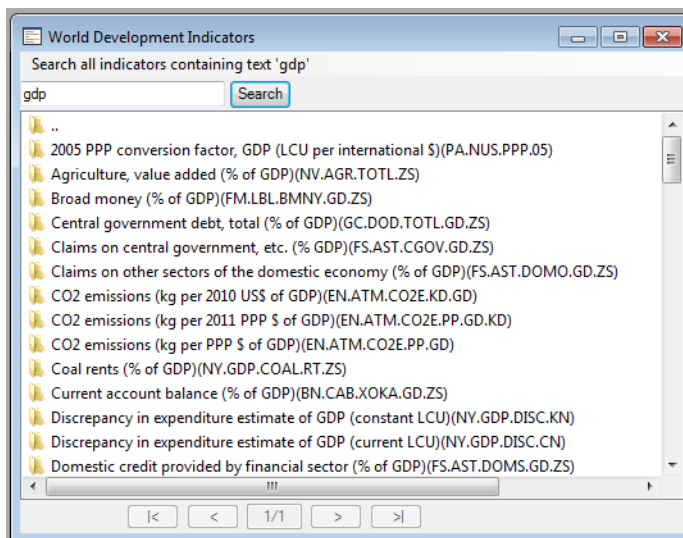
- You can use the < and > buttons to navigate to the next and the previous indicator pages.

- You can use the |< and >| buttons to navigate to the first or last indicator pages.

- Right-clicking on items in the browser will bring up a menu with items for performing useful tasks:
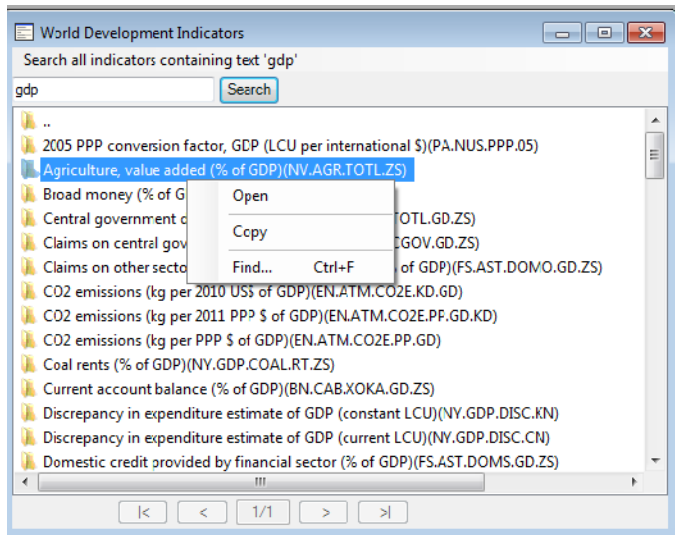
1. Open can be used to navigate inside a folder or bring an indicator in the database window.

2. Copy can be used to select an indicator to be copied to a workfile

3. Find allows you to do a keyword search.
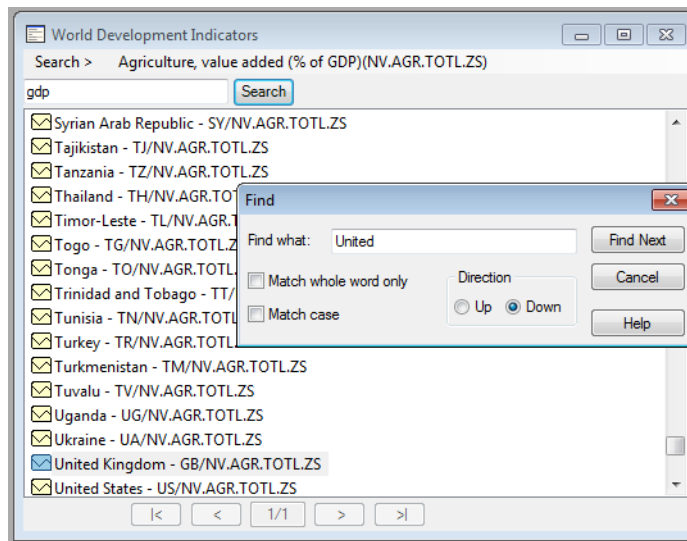
Once you have found the series of interest, you may drag-and-drop (or copy-and-paste) the series directly into an EViews workfile in the usual fashion.

Working with DRIPro Links

EViews has the ability to remotely access databases hosted by DRI. Subscribers to DRI DRI-Pro data services can use these features to access data directly from within EViews.

Although the interface to remote databases is very similar to that of local databases, there are some differences due to the nature of the connection. There are also some issues specifically related to accessing DRI data. The following sections docOnce you have found the series of interest, you may drag-and-drop (or copy-and-paste) the series directly into an EViews workfile in the usual fashion.

ument these differences.

Enabling DRI Access

In order to access DRI data services, you will need to have an active DRIPro account. If you are not an existing DRIPro customer but may be interested in becoming one, you should contact Global Insight for details (http://www.globalinsight.com).

Access to DRI data will not be possible unless you have already installed and configured the DRIPro server software. If you have difficulties with getting the software to work, you should contact Global Insight directly for technical support.

Creating a Database Link

A remote DRI database is represented in EViews by a *database link*. A database link resembles a local database, consisting of a set of files on disk, but instead of containing the data itself, a database link contains information as to how to access the remote data. A database link also contains a cache in which copies of recently retrieved objects are kept, which can substantially reduce the time taken to perform some database operations.

You can create a database link by following a similar procedure to that used to create a local database. Select **File/New/Database…** from the main menu, then select **DRIPro Link** in the field **Database/File Type**. The dialog should change appearance so that a number of extra fields are displayed. Enter the name you would like to give the new database link in **Cache name/path**. You may wish to name the database link after the DRI databank to which it links.

In the **Connection name** field you should enter the name of the DRIPro connection you would like to use, as it appears in the **Connection Settings** box in the DRIPro configuration program. If you have only configured a single connection, and have not modified the connection name, the connection name will be DEFAULT, and this will be filled in automatically by EViews if you leave the field blank.

In the **DRI Databank** field you should input the full name of the DRIPro bank to which you would like to connect, not including any leading "@" sign. For example, to connect to the DRI U.S. Central database, you should enter the name `uscen`. Each EViews database link may be associated with only one DRI databank, although you can create as many database links as you require.

The **Local Password** field may be used to set a password that must be entered whenever you wish to use the database link. This should not be confused with your DRIPro username and password, which you must already have provided in the DRIPro configuration program. Accessing a database link which contains a local password will cause a dialog to appear which prompts the user to input the password. Access to the remote database is only provided if the remote password is valid. Leave this field blank if you do not want a password to be attached to the database link.

When you have finished filling in the dialog fields, click on the **OK** button. A new database will be created and a database window should appear on the screen.

The database link window is very similar to a normal EViews database window. You should be able to perform basic query operations and simple fetching of series without any special instructions. Note, however, that it is not possible to modify a remote DRI database from within EViews, so operations which involve writing to the database have been removed. There are a number of other complications related to dealing with DRIPro databases that are described "Issues with DRI Frequencies" on page 395.

Understanding the Cache

A database link includes a cache of recently fetched objects which is used to speed up certain operations on the database. In some circumstances, fetching an object from the database will simply retrieve a copy from the local cache, rather than fetching a fresh copy of the data from the remote site. Even if a fresh copy is retrieved, having a previous copy of the series in the cache can substantially speed up retrieval.

You can regulate the caching behavior of the database link in a number of different ways. The basic option which determines under what circumstances a new copy of the data should be fetched is the *days before refresh*. If you attempt to fetch an object from the database link, and the copy of the object currently in the cache was fetched more recently than the days before refresh value, then the object currently in the cache will be returned instead of a fresh copy being fetched. For example, if days before refresh is set to one, any object which has already been fetched today will be retrieved from the cache, while any object which has not yet been fetched today will be retrieved from the remote site. Similarly, if days before refresh is set to seven, then an object in the cache must be more than a week old before a new copy of the object will be fetched. If days before refresh is set to zero, then a new copy of the data is fetched every time it is used.

You can change the days before refresh setting by clicking on the **Proc** button at the top of the database link window, then choosing **Link Options…** from the pop-up menu. A dialog will appear:

The dialog contains a number of fields, one of which is labeled **Days before refreshing objects**. Type a new number in the field to change the value.

The same dialog also contains a button marked **Reset cache now**. This button can be used to modify the behavior documented above. Clicking on the button causes the cache to mark all objects in the cache as out of date, so that the next time each object is fetched, it is guaranteed that a fresh copy will be retrieved. This provides a simple way for you to be certain that the database link will not return any data fetched before a particular time.
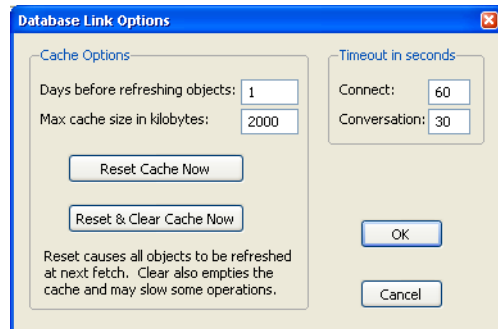
The dialog also contains some options for managing the size of the cache. The field marked **Maximum cache size in kilobytes** can be used to set the maximum size that the cache will be allowed to grow to on disk. If the cache grows above this size, a prompt will appear warning you that the cache has exceeded the limit and asking if you would like to compact the cache. Compacting is performed by deleting objects from oldest to newest until the cache size is reduced to less than three quarters of its maximum size. The cache is then packed to reclaim the empty space.

You can also completely clear the contents of the cache at any time by clicking on the button marked **Reset & Clear Cache Now**.

You can always examine the current contents of the database cache by clicking on the **Cache** button at the top of the database link window. This will display the names of all objects currently in the cache.

Configuring Link Options

The **Database Link Options** dialog also allows you to specify a number of timeout values. In most cases, the default values will behave acceptably. If you believe you are having problems with EViews aborting the connection too early, or you would like to shorten the times so as to receive a timeout message sooner, then enter new values in the appropriate fields.

- **Connection timeout—**is the length of time, in seconds, that EViews will wait for a response when first connecting to DRI. Depending on the type of connection you are making to DRI, this can take a significant amount of time.

- **Conversation timeout—**is the length of time, in seconds, that EViews will wait for a response from DRIPro when carrying out a transaction after a connection has already been made.

The values are attached to a particular database link, and can be reset at any time.

Dealing with Illegal Names

DRI databanks contain a number of series with names which are not legal names for EViews objects. In particular, DRI names frequently contain the symbols "@", "&" and "%", none of which are legal characters in EViews object names. We have provided a number of features to allow you to work with these series within EViews.

Because the "@" symbol is so common in DRI names, while the underline symbol (which is a legal character in EViews) is unused, we have hard-coded the rule that all underlines in EViews are mapped into "@" symbols in DRI names when performing operations on an DRI database link. For example, if there is a series with the name JQIMET@UK, you should refer to this series inside EViews as JQIMET_UK. Note that when performing queries, EViews will automatically replace the "@" symbol by an underline in the object name before displaying the query results on the screen. Consequently, if you are fetching data by copying-and-pasting objects from a query window, you do not need to be aware of this translation.

For other illegal names, you should use the object aliasing features (see "Object Aliases and Illegal Names" on page 341) to map the names into legal EViews object names.

Issues with DRI Frequencies

DRI databases have a different structure than EViews databases. An EViews database can contain series with mixed frequencies. A DRI database can contain data of only a single frequency. In order that similar data may be grouped together, each DRI databank is actually composed of a series of separate databases, one for each frequency. When working with DRI data from within DRIPro software, you will often have to specify at exactly which frequency

a particular series can be found. In some cases, a DRI databank may contain a series with the same name stored at several different frequencies.

Because this approach is inconsistent with the way that EViews works, we have tried to create a simpler interface to DRI data where you do not need to keep track of the frequency of each series that you would like to fetch. Instead, you can simply fetch a series by name or by selecting it from the query window, and EViews will do whatever is necessary to find out the frequency for you.

An ambiguity can arise in doing this, where a series with the same name appears at a variety of different frequencies in the DRI databank. By default, EViews resolves this ambiguity by always fetching the *highest* frequency data available. EViews will then perform necessary frequency conversions using the standard rules for frequency conversion in EViews (see "Frequency Conversion" on page 170).

In many cases, this procedure will exactly replicate the results that would be obtained if the lower frequency data was fetched directly from DRIPro. In some cases (typically when the series in question is some sort of ratio or other expression of one or more series), the figures may not match up exactly. In this case, if you know that the DRI data exists at multiple frequencies and you are familiar with DRI frequency naming conventions, you can explicitly fetch a series from a DRI database at a particular frequency by using a modified form of the command line form of fetch. Simply add the DRI frequency in parentheses after the name of the series. For example, the command:

```
fetch x(Q) y(A)
```

will fetch the series X and Y from the current default database, reading the quarterly frequency copy of X and the annual frequency copy of Y. If you request a frequency at which the data are not available, you will receive an error message. You should consult DRI documentation for details on DRI frequencies.

Limitations of DRI Queries

Queries to DRI database links are more limited than those available for EViews databases. The following section documents the restrictions.

First, queries on DRI databases allow only a subset of the fields available in EViews databases to be selected. The fields supported are: `name`, `type`, `freq`, `start`, `end`, `last_update` and `description`.

Second, the only fields which can be used in "where" conditions in a query on a DRIPro database link are name and description. (EViews does not support queries by frequency because of the ambiguities arising from DRI frequencies noted above).

Each of these fields has only one operator, the "matches" operator, and operations on the two fields can only be joined together using the "and" operator.

The "matches" operator is also limited for queries on DRI databases, matching only a subset of the expressions available for EViews databases. In particular, the pattern expression in a query on an DRI database must either have the form

    a or b or … c

or the form

    a and b and … c

Mixing of "and" and "or" is not allowed, and the "not" operator is not supported.

Patterns, however, are allowed and follow the normal EViews rules where "?" denotes any single character and "*" denotes zero or more characters.

Sorting of results by field is not supported.

Dealing with Common Problems

As stated in the introduction, you must install and configure the DRIPro software before EViews will be able to connect to DRI. If you cannot connect to DRI using the DRIPro software, you should contact DRI directly for assistance.

Assuming that you have correctly configured your DRIPro connection, in most cases EViews will be able to recover adequately from unexpected problems which arise during a DRIPro session without user intervention. Sometimes this will require EViews to automatically disconnect then reconnect to DRI.

There are some circumstances in which EViews may have problems making a connection. In order to connect to DRI, EViews uses a program written by DRI called DRIprosv. You can tell when this program is running by looking for the icon labeled "DRIpro server" in the Windows taskbar. Because of problems that can arise with multiple connections, EViews will not attempt to use the program if it is already running. Instead, EViews will report an error message "DRI server software already running". If there is another application which is using the connection to DRI, you can simply close down that program and the DRIPro server software should shut down automatically. If this is not the case, you may have to close down the DRIPro server software manually. Simply click on the icon in the Windows taskbar with the right mouse button, then select **Close** from the pop-up menu.

You may also use this as a procedure for forcing the DRIPro connection to terminate. Closing down the server software may cause EViews to report an error if it is currently carrying out a database transaction, but should otherwise be safe. EViews will restart the server software whenever it is needed.

Note that running other DRIPro software while EViews is using the DRIPro server software may cause EViews to behave unreliably.

Part II. Basic Data Analysis

The following chapters describe the EViews objects and tools that you will use to perform basic data analysis.

- Chapter 11. "Series," beginning on page 401 describes the series object. Series are the basic unit of numeric data in EViews and are the basis for most univariate analysis. This chapter documents the basic data analysis and display features associated with series.

- Chapter 12. "Groups," on page 547 documents the views and procedures for the group object. Groups are collections of series (and series equivalent objects) which form the basis for a variety of multivariate graphical display and data analyses.

- Chapter 13. "Graphing Data," beginning on page 617 describes the display of graph views of data in series and group objects.

- Chapter 14. "Categorical Graphs," on page 717 describes the construction of categorical graphs formed using subsets of the data in series or groups.

Chapter 11. Series

EViews provides various statistical graphs, descriptive statistics, and procedures as *views* and *procedures* of a numeric series. Once you have read or generated data into series objects using any of the methods described in Chapter 5. "Basic Data Handling," Chapter 6. "Working with Data," and Chapter 10. "EViews Databases," you are ready to perform statistical and graphical analysis using the data contained in the series.

Series views compute various statistics for a single series and display these statistics in various forms such as spreadsheets, tables, and graphs. The views range from a simple line graph, to kernel density estimators. Series procedures create new series from the data in existing series. These procedures include various seasonal adjustment methods, exponential smoothing methods, and the Hodrick-Prescott filter.

The group object is used when working with more than one series at the same time. Methods which involve groups are described in Chapter 12. "Groups," on page 547.

To access the views and procedures for series, open the series window by double clicking on the series name in the workfile, or by typing `show` followed by the name of the series in the command window.

Series Views Overview

The series view drop-down menu is divided into four blocks. The first block lists views that display the underlying data in the series. The second and third blocks provide access to general statistics (the views in the third block are mainly for time series analysis). The fourth block allows you to modify and display the series labels.
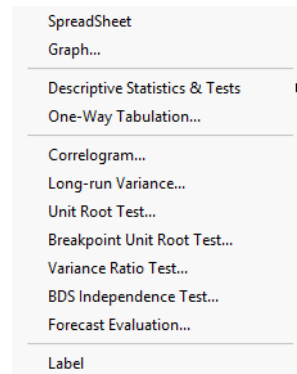
Spreadsheet

The spreadsheet view is the basic tabular view for the series data and displays the raw, mapped, or transformed data series data in spreadsheet format.

You may customize your spreadsheet view in a number of ways (see "Changing the Spreadsheet Display" in "Data Objects" on page 124).

The spreadsheet view may also be used to edit your series values (see "Editing a series" on page 131 and "Series Adjust" on page 433 for discussion).

In addition, the right-mouse button menu allows you to write the contents of the spreadsheet view to a CSV, tab-delimited ASCII text, RTF, HTML, Windows Metafile, or PDF file. Simply right-mouse button click, select the **Save table to disk...** menu item, and fill out the resulting dialog.

Graph

The **Graph...** menu item brings up the **Graph Options** dialog, which allows you to select various types of graphical display of the series. You can create graph objects by freezing these views. See Chapter 13. "Graphing Data," beginning on page 617 for a discussion of techniques for creating and customizing the graphical display.

Descriptive Statistics & Tests

This set of views displays various summary statistics for the series. The submenu contains entries for histograms, basic statistics, and statistics by classification.

Histogram and Stats

This view displays the frequency distribution of your series in a histogram. The histogram divides the series range (the

distance between the maximum and minimum values) into a number of equal length intervals or bins and displays a count of the number of observations that fall into each bin.

(If you would like to produce a histogram where you have greater control over the bin width and placement, or if you would like to construct related graphs such as a kernel density plot or histogram polynomial, you should use the graph view of the series.)

A complement of standard descriptive statistics are displayed along with the histogram. All of the statistics are calculated using the observations in the current sample.
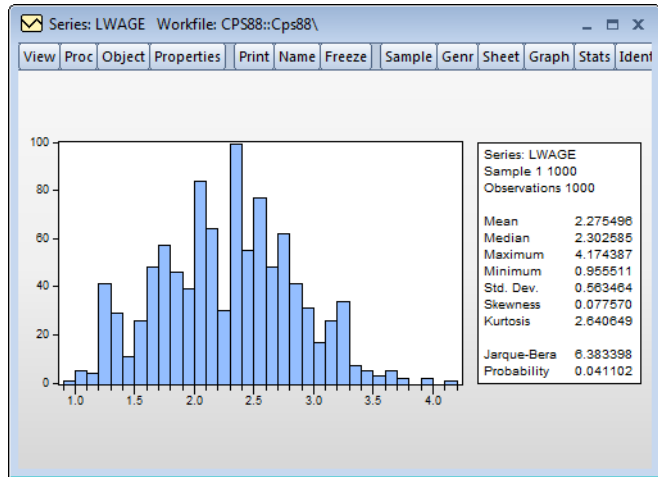
- **Mean** is the average value of the series, obtained by adding up the series and dividing by the number of observations.

- **Median** is the middle value (or average of the two middle values) of the series when the values are ordered from the smallest to the largest. The median is a robust measure of the center of the distribution that is less sensitive to outliers than the mean.

- **Max** and **Min** are the maximum and minimum values of the series in the current sample.

- **Std. Dev.** (standard deviation) is a measure of dispersion or spread in the series. The standard deviation is given by:

$$s = \sqrt{\left(\sum_{i=1}^{N} (y_i - \bar{y})^2\right) / (N-1)} \tag{11.1}$$

where N is the number of observations in the current sample and $\bar{y}$ is the mean of the series.

- **Skewness** is a measure of asymmetry of the distribution of the series around its mean. Skewness is computed as:

$$S = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - \bar{y}}{\hat{\sigma}}\right)^3 \tag{11.2}$$

where $\hat{\sigma}$ is an estimator for the standard deviation that is based on the biased estimator for the variance $(\hat{\sigma} = s\sqrt{(N-1)/N})$. The skewness of a symmetric distribution, such as the normal distribution, is zero. Positive skewness means that the distribution has a long right tail and negative skewness implies that the distribution has a long left tail.

- **Kurtosis** measures the peakedness or flatness of the distribution of the series. Kurtosis is computed as

$$K = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - \bar{y}}{\hat{\sigma}} \right)^4 \tag{11.3}$$

where $\hat{\sigma}$ is again based on the biased estimator for the variance. The kurtosis of the normal distribution is 3. If the kurtosis exceeds 3, the distribution is peaked (leptokurtic) relative to the normal; if the kurtosis is less than 3, the distribution is flat (platykurtic) relative to the normal.

- **Jarque-Bera** is a test statistic for testing whether the series is normally distributed. The test statistic measures the difference of the skewness and kurtosis of the series with those from the normal distribution. The statistic is computed as:

$$\text{Jarque-Bera} = \frac{N}{6} \left(S^2 + \frac{(K-3)^2}{4} \right) \tag{11.4}$$

where S is the skewness, and K is the kurtosis.

Under the null hypothesis of a normal distribution, the Jarque-Bera statistic is distributed as χ^2 with 2 degrees of freedom. The reported Probability is the probability that a Jarque-Bera statistic exceeds (in absolute value) the observed value under the null hypothesis—a small probability value leads to the rejection of the null hypothesis of a normal distribution. For the LWAGE series displayed above, we reject the hypothesis of normal distribution at the 5% level but not at the 1% significance level.

Stats Table

The **Stats Table** view displays descriptive statistics for the series in tabular form.

Note that this view provides slightly more information than the **Histogram and Stats** view.
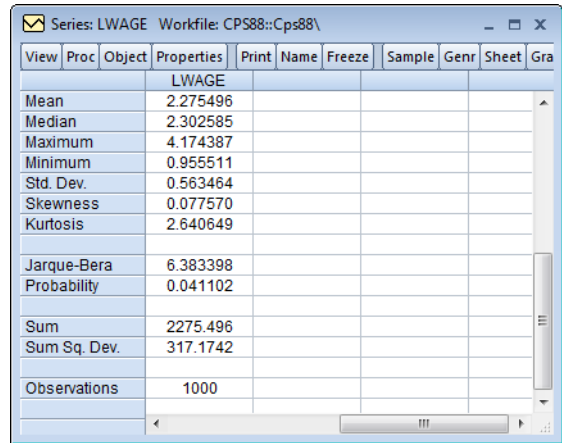
Stats by Classification

This view allows you to compute the descriptive statistics of a series for various subgroups of your sample. If you select **View/Descriptive Statistics & Tests/Stats by Classification…** a **Statistics by Classification** dialog box appears:
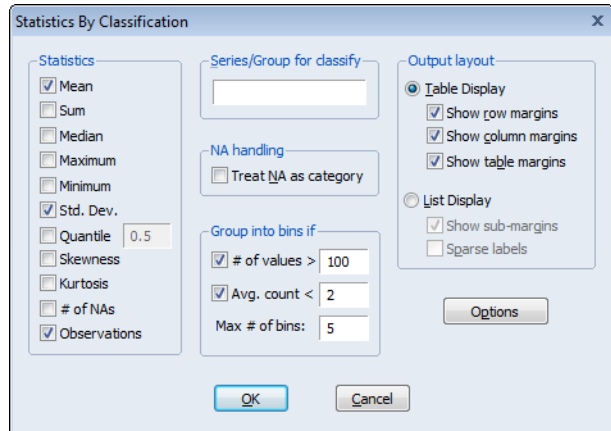
The **Statistics** option at the left allows you to choose the statistic(s) you wish to compute.

In the **Series/Group for Classify** field enter series or group names that define your subgroups. You must type at least one name. Descriptive statistics will be calculated for each unique value of the classification series (also referred to as a factor) unless binning is selected. You may type more than one series or group name; separate each name by a space. The quantile statistic requires an additional argument (a number between 0 and 1) corresponding to the desired quantile value. Click on the **Options** button to choose between various methods of computing the quantiles. See "Empirical CDF" on page 696 for details.

By default, EViews excludes observations which have missing values for any of the classification series. To treat NA values as a valid subgroup, select the **NA handling** option.

The **Layout** option allows you to control the display of the statistics. Table layout arrays the statistics in cells of two-way tables. The list form displays the statistics in a single line for each classification group.

The **Table** and **List** options are only relevant if you use more than one series as a classifier.

The **Sparse Labels** option suppresses repeating labels in list mode to make the display less cluttered.

The **Row Margins**, **Column Margins**, and **Table Margins** instruct EViews to compute statistics for aggregates of your subgroups. For example, if you classify your sample on the basis of gender and age, EViews will compute the statistics for each gender/age combination. If you elect to compute the marginal statistics, EViews will also compute statistics corresponding to each gender, and each age subgroup.

A classification may result in a large number of distinct values with very small cell sizes. By default, EViews automatically groups observations into categories to maintain moderate cell sizes and numbers of categories. **Group into Bins** provides you with control over this process.

Setting the **# of values** option tells EViews to group data if the classifier series takes more than the specified number of distinct values.

The **Avg. count** option is used to bin the series if the average count for each distinct value of the classifier series is less than the specified number.

The **Max # of bins** specifies the maximum number of subgroups to bin the series. Note that this number only provides you with approximate control over the number of bins.

The default setting is to bin the series into 5 subgroups if either the series takes more than 100 distinct values or if the average count is less than 2. If you do not want to bin the series, unmark both options.

For example, consider the following stats by classification table view of the series LWAGE, categorized by values of MARRIED and UNION (from the workfile "Cps88.WF1"):

```
Descriptive Statistics for LWAGE
Categorized by values of MARRIED and UNION
Date: 08/18/09   Time: 16:26
Sample: 1 1000
Included observations: 1000
```

Mean Median Std. Dev. Obs.		UNION		
		0	1	All
	0	1.993829 1.906575 0.574636 305	2.387019 2.409132 0.395838 54	2.052972 2.014903 0.568689 359
MARRIED	1	2.368924 2.327278 0.557405 479	2.492371 2.525729 0.380441 162	2.400123 2.397895 0.520910 641
	All	2.223001 2.197225 0.592757 784	2.466033 2.500525 0.386134 216	2.275496 2.302585 0.563464 1000

The header indicates that the table cells are categorized by two series MARRIED and UNION. These two series are dummy variables that take only two values. No binning is performed; if the series were binned, intervals rather than a number would be displayed in the margins.

The upper left cell of the table indicates the reported statistics in each cell; in this case, the median and the number of observations are reported in each cell. The row and column labeled **All** correspond to the **Row Margin** and **Column Margin** options described above.

Here is the same view in list form with sparse labels:

Descriptive Statistics for LWAGE
Categorized by values of MARRIED and UNION
Date: 08/18/09   Time: 16:27
Sample: 1 1000
Included observations: 1000

UNION	MARRIED	Mean	Median	Std. Dev.	Obs.
0	0	1.993829	1.906575	0.574636	305
	1	2.368924	2.327278	0.557405	479
	All	2.223001	2.197225	0.592757	784
1	0	2.387019	2.409132	0.395838	54
	1	2.492371	2.525729	0.380441	162
	All	2.466033	2.500525	0.386134	216
All	0	2.052972	2.014903	0.568689	359
	1	2.400123	2.397895	0.520910	641
	All	2.275496	2.302585	0.563464	1000

For series functions that compute by-group statistics and produce new series, see "By-Group Statistics" on page 580 in the *Command and Programming Reference*.

Simple Hypothesis Tests

This view carries out simple hypothesis tests regarding the mean, median, and the variance of the series. These are all single sample tests; see "Equality Tests by Classification" on page 411 for a description of two sample tests. If you select **View/Descriptive Statistics & Tests/Simple Hypothesis Tests**, the **Series Distribution Tests** dialog box will be displayed. Depending on which edit field on the left you enter a value, EViews will perform a different test.

Mean Test

Carries out the test of the null hypothesis that the mean μ of the series X is equal to a specified value m against the two-sided alternative that it is not equal to m:

$$H_0: \quad \mu = m$$
$$H_1: \quad \mu \neq m.$$

(11.5)

If you do not specify the standard deviation of X, EViews reports a t-statistic computed as:

$$t = \frac{\overline{X} - m}{s / \sqrt{N}}$$

(11.6)

where $\overline{X}$ is the sample mean of X, s is the unbiased sample standard deviation, and N is the number of observations of X. If X is normally distributed, under the null hypothesis the t-statistic follows a t-distribution with $N - 1$ degrees of freedom.

If you specify a value for the standard deviation of X, EViews also reports a z-statistic:

$$z = \frac{\overline{X} - m}{\sigma / \sqrt{N}} \tag{11.7}$$

where σ is the specified standard deviation of X. If X is normally distributed with standard deviation σ, under the null hypothesis, the z-statistic has a standard normal distribution.

To carry out the mean test, type in the value of the mean under the null hypothesis in the edit field next to **Mean**. If you want to compute the z-statistic conditional on a known standard deviation, also type in a value for the standard deviation in the right edit field. You can type in any number or standard EViews expression in the edit fields.

Hypothesis Testing for LWAGE
Date: 07/31/06   Time: 11:03
Sample: 1 1000
Included observations: 1000
Test of Hypothesis: Mean = 2.000000

Sample Mean = 2.275496
Sample Std. Dev. = 0.563464

Method	Value	Probability
t-statistic	15.46139	0.0000

The reported probability value is the p-value, or marginal significance level, against a two-sided alternative. If this probability value is less than the size of the test, say 0.05, we reject the null hypothesis. Here, we strongly reject the null hypothesis for the two-sided test of equality. The probability value for a one-sided alternative is one half the p-value of the two-sided test.

Variance Test

Carries out the test of the null hypothesis that the variance of a series X is equal to a specified value σ^2 against the two-sided alternative that it is not equal to σ^2:

$$H_0: \ \text{var}(x) = \sigma^2$$
$$H_1: \ \text{var}(x) \neq \sigma^2. \tag{11.8}$$

EViews reports a χ^2 statistic computed as:

$$\chi^2 = \frac{(N-1)s^2}{\sigma^2} \tag{11.9}$$

where N is the number of observations, s is the sample standard deviation, and $\overline{X}$ is the sample mean of X. Under the null hypothesis and the assumption that X is normally distributed, the statistic follows a χ^2 distribution with $N-1$ degrees of freedom. The probability

value is computed as $\min(p, 1 - p)$, where p is the probability of observing a χ^2-statistic as large as the one actually observed under the null hypothesis.

To carry out the variance test, type in the value of the variance under the null hypothesis in the field box next to **Variance**. You can type in any *positive* number or expression in the field.

Hypothesis Testing for LWAGE
Date: 071/31/06   Time: 01:22
Sample: 1 1000
Included observations: 1000
Test of Hypothesis: Variance = 0.300000

Sample Variance = 0.317492

Method	Value	Probability
Variance Ratio	1057.247	0.0979

Median Test

Carries out the test of the null hypothesis that the median of a series X is equal to a specified value m against the two-sided alternative that it is not equal to m:

$$H_0:\ \mathrm{med}(x) = m$$
$$H_1:\ \mathrm{med}(x) \neq m.$$

(11.10)

EViews reports three rank-based, nonparametric test statistics. The principal references for this material are Conover (1980) and Sheskin (1997).

- **Binomial sign test**. This test is based on the idea that if the sample is drawn randomly from a binomial distribution, the sample proportion above and below the true median should be one-half. Note that EViews reports two-sided p-values for both the sign test and the large sample normal approximation (with continuity correction).

- **Wilcoxon signed ranks test**. Suppose that we compute the absolute value of the difference between each observation and the mean, and then rank these observations from high to low. The Wilcoxon test is based on the idea that the sum of the ranks for the samples above and below the median should be similar. EViews reports a p-value for the asymptotic normal approximation to the Wilcoxon T-statistic (correcting for both continuity and ties). See Sheskin (1997, p. 82–94) and Conover (1980, p. 284).

- **Van der Waerden (normal scores) test**. This test is based on the same general idea as the Wilcoxon test, but is based on smoothed ranks. The signed ranks are smoothed by converting them to quantiles of the normal distribution (normal scores). EViews reports the two-sided p-value for the asymptotic normal test described by Conover (1980).

To carry out the median test, type in the value of the median under the null hypothesis in the edit box next to **Median**. You can type any numeric expression in the edit field.

```
Hypothesis Testing for LWAGE
Date: 07/31/06   Time: 11:06
Sample: 1 1000
Included observations: 1000
Test of Hypothesis:  Median =  2.250000
```

Sample Median = 2.302585

Method	Value	Probability
Sign (exact binomial)	532	0.0463
Sign (normal approximation)	1.992235	0.0463
Wilcoxon signed rank	1.134568	0.2566
van der Waerden (normal scores)	1.345613	0.1784

Median Test Summary

Category	Count	Mean Rank
Obs > 2.250000	532	489.877820
Obs < 2.250000	468	512.574786
Obs = 2.250000	0	
Total	1000	

Equality Tests by Classification

This view allows you to test equality of the means, medians, and variances across subsamples (or subgroups) of a single series. For example, you can test whether mean income is the same for males and females, or whether the variance of education is related to race. The tests assume that the subsamples are independent.

For single sample tests, see the discussion of "Simple Hypothesis Tests" on page 408. For tests of equality across different series, see "Tests of Equality" on page 589.

Select **View/Descriptive Statistics & Tests/ Equality Tests by Classification…** and the **Tests by Classification** dialog box appears.

First, select whether you wish to test the mean, the median or the variance. Specify the subgroups, the NA handling, and the grouping options as described in "Stats by Classification," beginning on page 405.
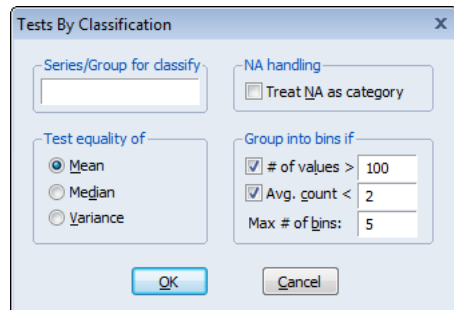
Mean Equality Test

This test is a single-factor, between-subjects, analysis of variance (ANOVA). The basic idea is that if the subgroups have the same mean, then the variability between the sample means (between groups) should be the same as the variability within any subgroup (within group).

Denote the i-th observation in subgroup g as $x_{g,\,i}$, where $i = 1, \ldots, n_g$ for groups $g = 1, 2, \ldots G$. The between and within sums of squares are defined as:

$$SS_B = \sum_{g=1}^{G} n_g(\bar{x}_g - \bar{x})^2 \tag{11.11}$$

$$SS_W = \sum_{g=1}^{G} \sum_{i=1}^{n_g} (x_{ig} - \bar{x}_g)^2 \tag{11.12}$$

where $\bar{x}_g$ is the sample mean within group g and $\bar{x}$ is the overall sample mean. The F-statistic for the equality of group means is computed as:

$$F = \frac{SS_B/(G-1)}{SS_W/(N-G)} \tag{11.13}$$

where N is the total number of observations. The F-statistic has an F-distribution with $G-1$ numerator degrees of freedom and $N-G$ denominator degrees of freedom under the null hypothesis of independent and identical normal distributed data, with equal means and variances in each subgroup.

When the subgroup variances are heterogeneous, we may use the Welch (1951) version of the test statistic. The basic idea is to form a modified F-statistic that accounts for the unequal variances. Using the Cochran (1937) weight function,

$$w_g = n_g / s_g^2 \tag{11.14}$$

where s_g^2 is the sample variance in subgroup g, we form the modified F-statistic

$$F^* = \frac{\displaystyle\sum_{g=1}^{G} w_g(\bar{x}_g - \bar{x}^*)^2 / (G-1)}{1 + \dfrac{2(G-2)}{G^2-1} \displaystyle\sum_{g=1}^{G} \dfrac{(1-h_g)^2}{n_g-1}} \tag{11.15}$$

where h_g is a normalized weight and $\bar{x}^*$ is the weighted grand mean,

$$h_g = w_g \Big/ \left(\sum_{g=1}^{G} w_k \right)$$

$$\bar{x}* = \sum_{g=1}^{G} h_k \bar{x}_g$$

(11.16)

The numerator of the adjusted statistic is the weighted between-group mean squares and the denominator is the weighted within-group mean squares. Under the null hypothesis of equal means but possibly unequal variances, $F*$ has an approximate F-distribution with $(G-1, DF*)$ degrees-of-freedom, where

$$DF* = \frac{(G^2 - 1)}{3 \sum\limits_{g=1}^{G} \dfrac{(1 - h_g)^2}{n_g - 1}}$$

(11.17)

For tests with only two subgroups ($G = 2$), EViews also reports the t-statistic, which is simply the square root of the F-statistic with one numerator degree of freedom. Note that for two groups, the Welch test reduces to the Satterthwaite (1946) test.

The top portion of the output contains the ANOVA results for a test of equality of means for LWAGE categorized by the four groups defined by the series MARRIED and UNION:

Test for Equality of Means of LWAGE
Categorized by values of MARRIED and UNION
Date: 08/18/09   Time: 16:31
Sample: 1 1000
Included observations: 1000

Method	df	Value	Probability
Anova F-test	(3, 996)	43.40185	0.0000
Welch F-test*	(3, 231.728)	45.31787	0.0000

*Test allows for unequal cell variances

Analysis of Variance

Source of Variation	df	Sum of Sq.	Mean Sq.
Between	3	36.66990	12.22330
Within	996	280.5043	0.281631
Total	999	317.1742	0.317492

The results show that there is strong evidence that LWAGE differs across groups defined by MARRIED and UNION; both the standard ANOVA and the Welch adjusted ANOVA statistics are in excess of 40, with probability values near zero.

The analysis of variance table shows the decomposition of the total sum of squares into the between and within sum of squares, where:

Mean Sq. = Sum of Sq./df

The F-statistic is the ratio:

F = Between Mean Sq./Within Mean Sq.

The bottom portion of the output provides the category statistics:

Category Statistics

UNION	MARRIED	Count	Mean	Std. Dev.	Std. Err. of Mean
0	0	305	1.993829	0.574636	0.032904
0	1	479	2.368924	0.557405	0.025468
1	0	54	2.387019	0.395838	0.053867
1	1	162	2.492371	0.380441	0.029890
	All	1000	2.275496	0.563464	0.017818

Median (Distribution) Equality Tests

EViews computes various rank-based nonparametric tests of the hypothesis that the sub-groups have the same general distribution, against the alternative that at least one subgroup has a different distribution.

We note that the "median" category in which we place these tests is somewhat misleading since the tests focus more generally on the equality of various statistics computed across subgroups. For example, the Wilcoxon test examines the comparability of mean ranks across subgroups. The categorization reflects common usage for these tests and various textbook definitions. The tests should, of course, have power against median differences.

In the two group setting, the null hypothesis is that the two subgroups are independent samples from the same general distribution. The alternative hypothesis may loosely be defined as "the values [of the first group] tend to differ from the values [of the second group]" (see Conover 1980, p. 281 for discussion). See also Bergmann, Ludbrook and Spooren (2000) for a more precise analysis of the issues involved.

- **Wilcoxon rank sum test**. This test is computed when there are two subgroups. The test is identical to the Wilcoxon signed rank median test ("Median Test" on page 410) but the division of the series into two groups is based upon the values of the classification variable instead of the value of the observation relative to the median.

- **Chi-square test** for the median. This is a rank-based ANOVA test based on the comparison of the number of observations above and below the overall median in each subgroup. This test is sometimes referred to as the *median test* (Conover, 1980).

Under the null hypothesis, the median chi-square statistic is asymptotically distributed as a χ^2 with $G - 1$ degrees of freedom. EViews also reports Yates' continuity corrected statistic. You should note that the use of this correction is controversial (Sheskin, 1997, p. 218).

- **Kruskal-Wallis one-way ANOVA by ranks**. This is a generalization of the Mann-Whitney test to more than two subgroups. The idea behind the Mann-Whitney test is to rank the series from smallest value (rank 1) to largest, and to compare the sum of the ranks from subgroup 1 to the sum of the ranks from subgroup 2. If the groups have the same median, the values should be similar.

  EViews reports the asymptotic normal approximation to the U-statistic (with continuity and tie correction) and the p-values for a two-sided test. For details, see Sheskin (1997). The test is based on a one-way analysis of variance using only ranks of the data. EViews reports the χ^2 chi-square approximation to the Kruskal-Wallis test statistic (with tie correction). Under the null hypothesis, this statistic is approximately distributed as a χ^2 with $G - 1$ degrees of freedom (see Sheskin, 1997).

- **van der Waerden (normal scores) test**. This test is analogous to the Kruskal-Wallis test, except that we smooth the ranks by converting them into normal quantiles (Conover, 1980). EViews reports a statistic which is approximately distributed as a χ^2 with $G - 1$ degrees of freedom under the null hypothesis.

The top portion of the output displays the test statistics:

Test for Equality of Medians of LWAGE
Categorized by values of MARRIED and UNION
Date: 07/31/07   Time: 01:29
Sample: 1 1000
Included observations: 1000

Method	df	Value	Probability
Med. Chi-square	1	95.40100	0.0000
Adj. Med. Chi-square	1	92.99015	0.0000
Kruskal-Wallis	3	116.1189	0.0000
Kruskal-Wallis (tie-adj.)	3	116.1557	0.0000
van der Waerden	3	112.5606	0.0000

In addition to the test statistics and p-values, EViews reports values for the components of the test statistics for each subgroup of the sample. For example, the column labeled **Mean Score** contains the mean values of the van der Waerden scores (the smoothed ranks) for each subgroup.

Category Statistics

UNION	MARRIED	Count	Median	> Overall Median	Mean Rank	Mean Score
0	0	305	1.906575	89	358.9082	-0.489333
0	1	479	2.327278	245	540.5073	0.161730
1	0	54	2.409132	35	568.6852	0.194415
1	1	162	2.525729	109	626.0556	0.380258
	All	1000	2.302585	478	500.5000	0.000322

Variance Equality Tests

Variance equality tests evaluate the null hypothesis that the variances in all G subgroups are equal against the alternative that at least one subgroup has a different variance. See Conover, *et al.* (1981) for a general discussion of variance equality testing.

- F-**test**. This test statistic is reported only for tests with two subgroups ($G = 2$). First, compute the variance for each subgroup and denote the subgroup with the larger variance as L and the subgroup with the smaller variance as S. Then the F-statistic is given by:

$$F = s_L^2 / s_S^2 \qquad (11.18)$$

  where s_g^2 is the variance in subgroup g. This F-statistic has an F-distribution with $n_L - 1$ numerator degrees of freedom and $n_S - 1$ denominator degrees of freedom under the null hypothesis of equal variance and independent normal samples.

- **Siegel-Tukey test**. This test statistic is reported only for tests with two subgroups ($G = 2$). The test assumes the two subgroups are independent and have equal medians. The test statistic is computed using the same steps as the Kruskal-Wallis test described above for the median equality tests ("Median (Distribution) Equality Tests" on page 414), with a different assignment of ranks. The ranking for the Siegel-Tukey test alternates from the lowest to the highest value for every other rank. The Siegel-Tukey test first orders all observations from lowest to highest. Next, assign rank 1 to the lowest value, rank 2 to the *highest* value, rank 3 to the second highest value, rank 4 to the second lowest value, rank 5 to the third lowest value, and so on. EViews reports the normal approximation to the Siegel-Tukey statistic with a continuity correction (Sheskin, 1997, p. 196–207).

- **Bartlett test**. This test compares the logarithm of the weighted average variance with the weighted sum of the logarithms of the variances. Under the joint null hypothesis that the subgroup variances are equal and that the sample is normally distributed, the test statistic is approximately distributed as a χ^2 with $G = 1$ degrees of freedom. Note, however, that the joint hypothesis implies that this test is sensitive to departures

from normality. EViews reports the adjusted Bartlett statistic. For details, see Sokal and Rohlf (1995) and Judge, *et al.* (1985).

- **Levene test**. This test is based on an analysis of variance (ANOVA) of the absolute difference from the mean. The F-statistic for the Levene test has an approximate F-distribution with $G = 1$ numerator degrees of freedom and $N - G$ denominator degrees of freedom under the null hypothesis of equal variances in each subgroup (Levene, 1960).

- **Brown-Forsythe (modified Levene) test**. This is a modification of the Levene test in which we replace the absolute *mean* difference with the absolute *median* difference. The Brown-Forsythe test appears to be a superior in terms of robustness and power to Levene (Conover, *et al.* (1981), Brown and Forsythe (1974a, 1974b), Neter, *et al.* (1996)).

As with the other equality tests, the top portion of the output displays the test results:

Test for Equality of Variances of LWAGE
Categorized by values of UNION and MARRIED
Date: 07/31/07   Time: 01:44
Sample: 1 1000
Included observations: 1000

Method	df	Value	Probability
Bartlett	3	42.78468	0.0000
Levene	(3, 996)	16.08021	0.0000
Brown-Forsythe	(3, 996)	14.88998	0.0000

The bottom portion of the output shows the intermediate calculations used in forming the test statistic:

Category Statistics

MARRIED	UNION	Count	Std. Dev.	Mean Abs. Mean Diff.	Mean Abs. Median Diff.
0	0	305	0.574636	0.479773	0.474788
0	1	54	0.395838	0.312659	0.311047
1	0	479	0.557405	0.445270	0.444236
1	1	162	0.380441	0.291903	0.290293
	All	1000	0.563464	0.423787	0.421424

Bartlett weighted standard deviation: 0.530689

Empirical Distribution Tests

EViews provides built-in Kolmogorov-Smirnov, Lilliefors, Cramer-von Mises, Anderson-Darling, and Watson empirical distribution tests. These tests are based on the comparison

between the empirical distribution and the specified theoretical distribution function. For a general description of empirical distribution function testing, see D'Agostino and Stephens (1986).

You can test whether your series is normally distributed, or whether it comes from, among others, an exponential, extreme value, logistic, chi-square, Weibull, or gamma distribution. You may provide parameters for the distribution, or EViews will estimate the parameters for you.

To carry out the test, simply double click on the series and select **View/Descriptive Statistics & Tests/Empirical Distribution Tests...** from the series window.

There are two tabs in the dialog. The **Test Specification** tab allows you to specify the parametric distribution against which you want to test the empirical distribution of the series. Simply select the distribution of interest from the drop-down menu. The small display window will change to show you the parameterization of the specified distribution.

You can specify the values of any known parameters in the edit field or fields. If you leave any field blank, EViews will estimate the corresponding parameter using the data contained in the series.
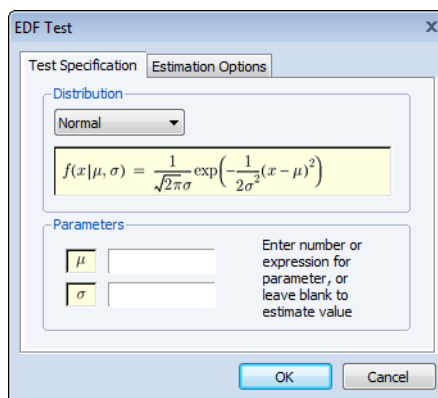
The **Estimation Options** tab provides control over any iterative estimation that is required. You should not need to use this tab unless the output indicates failure in the estimation process. Most of the options in this tab should be self-explanatory. If you select **User-specified starting values**, EViews will take the starting values from the C coefficient vector.

It is worth noting that some distributions have positive probability on a restricted domain. If the series data take values outside this domain, EViews will report an out-of-range error. Similarly, some of the distributions have restrictions on domain of the parameter values. If you specify a parameter value that does not satisfy this restriction, EViews will report an error message.

The output from this view consists of two parts. The first part displays the test statistics and associated probability values.

Empirical Distribution Test for LWAGE
Hypothesis: Normal
Date: 08/18/09   Time: 16:51
Sample: 1 1000
Included observations: 1000

Method	Value	Adj. Value	Probability
Lilliefors (D)	0.033172	NA	0.0115
Cramer-von Mises (W2)	0.132835	0.132902	0.0402
Watson (U2)	0.132577	0.132644	0.0288
Anderson-Darling (A2)	1.149731	1.150596	0.0052

Here, we show the output from a test for normality where both the mean and the variance are estimated from the series data. The first column, "Value", reports the asymptotic test statistics while the second column, "Adj. Value", reports test statistics that have a finite sample correction or adjusted for parameter uncertainty (in case the parameters are estimated). The third column reports p-value for the adjusted statistics.

All of the reported EViews p-values will account for the fact that parameters in the distribution have been estimated. In cases where estimation of parameters is involved, the distributions of the goodness-of-fit statistics are non-standard and distribution dependent, so that EViews may report a subset of tests and/or only a range of p-value. In this case, for example, EViews reports the Lilliefors test statistic instead of the Kolmogorov statistic since the parameters of the normal have been estimated. Details on the computation of the test statistics and the associated p-values may be found in Anderson and Darling (1952, 1954), Lewis (1961), Durbin (1970), Dallal and Wilkinson (1986), Davis and Stephens (1989), Csörgö and Faraway (1996) and Stephens (1986).

Method: Maximum Likelihood - d.f. corrected (Exact Solution)

Parameter	Value	Std. Error	z-Statistic	Prob.
MU	2.275496	0.017818	127.7056	0.0000
SIGMA	0.563464	0.012606	44.69899	0.0000

Log likelihood	-844.7868	Mean dependent var.		2.275496
No. of Coefficients	2	S.D. dependent var.		0.563464

The second part of the output table displays the parameter values used to compute the theoretical distribution function. Any parameters that are specified to estimate are estimated by maximum likelihood (for the normal distribution, the ML estimate of the standard deviation is subsequently degree of freedom corrected if the mean is not specified *a priori*). For parameters that do not have a closed form analytic solution, the likelihood function is maximized using analytic first and second derivatives. These estimated parameters are reported with a standard error and p-value based on the asymptotic normal distribution.

One-Way Tabulation

This view tabulates the series in ascending order, optionally displaying the counts, percentage counts, and cumulative counts. When you select **View/One-Way Tabulation…** the **Tabulate Series** dialog box will be displayed.
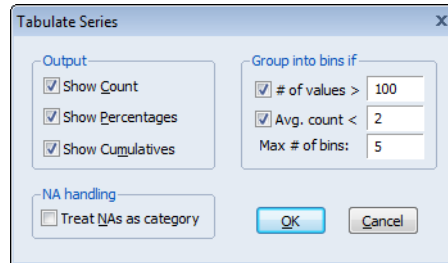
The **Output** options control which statistics to display in the table. You should specify the NA handling and the grouping options as described above in the discussion of "Stats by Classification" on page 405.

The **Output** options control which statistics to display in the table. You should specify the NA handling and the grouping options as described above in the discussion of "Stats by Classification" on page 405.

Value	Count	Percent	Cumulative Count	Cumulative Percent
0	1	0.10	1	0.10
2	1	0.10	2	0.20
3	1	0.10	3	0.30
4	2	0.20	5	0.50
5	3	0.30	8	0.80
6	6	0.60	14	1.40
7	8	0.80	22	2.20
8	17	1.70	39	3.90
9	30	3.00	69	6.90
10	45	4.50	114	11.40
11	40	4.00	154	15.40
12	412	41.20	566	56.60
13	76	7.60	642	64.20

Cross-tabulation (n-way tabulation) is also available as a group view. See "N-Way Tabulation" on page 585 for details.

Correlogram

This view displays the autocorrelation and partial autocorrelation functions up to the specified order of lags. These functions characterize the pattern of temporal dependence in the series and typically make sense only for time series data. When you select **View/Correlogram…** the **Correlogram Specification** dialog box appears.
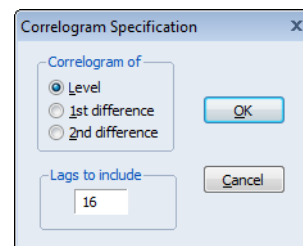
You may choose to plot the correlogram of the raw series (level) `x`, the first difference `d(x)=x-x(-1)`, or the second difference

```
d(x)-d(x(-1))  =  x-2x(-1)+x(-2)
```

of the series.

You should also specify the highest order of lag to display the correlogram; type in a positive integer in the field box. The series view displays the correlogram and associated statistics:
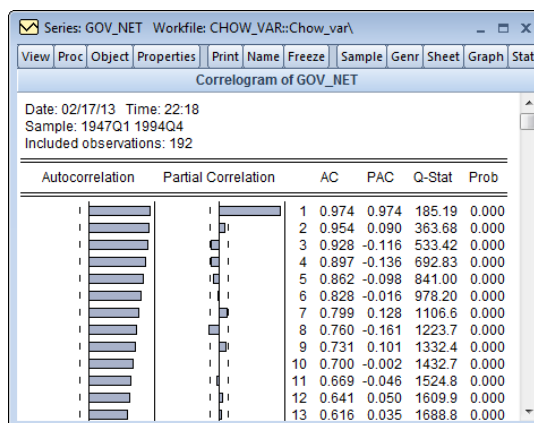
Autocorrelations (AC)

The autocorrelation of a series Y at lag k is estimated by:

$$\tau_k = \frac{\displaystyle\sum_{t=k+1}^{T} (Y_t - \overline{Y})(Y_{t-k} - \overline{Y})}{\displaystyle\sum_{t=1}^{T} (Y_t - \overline{Y})^2} \tag{11.19}$$

where $\overline{Y}$ is the sample mean of Y. This is the correlation coefficient for values of the series k periods apart. If τ_1 is nonzero, it means that the series is first order serially correlated. If τ_k dies off more or less geometrically with increasing lag k, it is a sign that the series obeys a low-order autoregressive (AR) process. If τ_k drops to zero after a small number of lags, it is a sign that the series obeys a low-order moving-average (MA) process. See "Background" on page 99 of *User's Guide II* for a more complete description of AR and MA processes.

Note that the autocorrelations estimated by EViews, which follow those in the literature, differ slightly from the theoretical definition of the autocorrelation:

$$\tau_k = \frac{\displaystyle\sum_{t = k+1}^{T} ((Y_t - \overline{Y})(Y_{t-k} - \overline{Y}_{t-k}))/(T-K)}{\displaystyle\sum_{t=1}^{T} (Y_t - \overline{Y})^2 / T} \tag{11.20}$$

where $\overline{Y}_{t-k} = \sum Y_{t-k}/(T-k)$. The difference arises since, for computational simplicity, EViews employs the same overall sample mean $\overline{Y}$ as the mean of both Y_t and Y_{t-k}. While both formulations are consistent estimators, the EViews formulation biases the result toward zero in finite samples.

The dotted lines in the plots of the autocorrelations are the approximate two standard error bounds computed as $\pm 2/(\sqrt{T})$. If the autocorrelation is within these bounds, it is not significantly different from zero at (approximately) the 5% significance level.

Partial Autocorrelations (PAC)

The partial autocorrelation at lag k is the regression coefficient on Y_{t-k} when Y_t is regressed on a constant, $Y_{t-1}, \ldots, Y_{t-k}$. This is a *partial* correlation since it measures the correlation of Y values that are k periods apart after removing the correlation from the intervening lags. If the pattern of autocorrelation is one that can be captured by an autoregression of order less than k, then the partial autocorrelation at lag k will be close to zero.

The PAC of a pure autoregressive process of order p, AR(p), cuts off at lag p, while the PAC of a pure moving average (MA) process asymptotes gradually to zero.

EViews estimates the partial autocorrelation at lag k recursively by

$$\phi_k = \begin{cases} \tau_1 & \text{for } k = 1 \\[2em] \dfrac{\tau_k - \sum\limits_{j=1}^{k-1} \phi_{k-1,j}\tau_{k-j}}{1 - \sum\limits_{j=1}^{k-1} \phi_{k-1,j}\tau_{k-j}} & \text{for } k > 1 \end{cases} \tag{11.21}$$

where τ_k is the estimated autocorrelation at lag k and where,

$$\phi_{k,j} = \phi_{k-1,j} - \phi_k \phi_{k-1,k-j}. \tag{11.22}$$

This is a consistent approximation of the partial autocorrelation. The algorithm is described in Box and Jenkins (1976, Part V, Description of computer programs). To obtain a more precise estimate of ϕ, simply run the regression:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \ldots + \beta_{k-1} Y_{t-(k-1)} + \phi_k Y_{t-k} + e_t \tag{11.23}$$

where e_t is a residual. The dotted lines in the plots of the partial autocorrelations are the approximate two standard error bounds computed as $\pm 2/(\sqrt{T})$. If the partial autocorrelation is within these bounds, it is not significantly different from zero at (approximately) the 5% significance level.

Q-Statistics

The last two columns reported in the correlogram are the Ljung-Box Q-statistics and their p-values. The Q-statistic at lag k is a test statistic for the null hypothesis that there is no autocorrelation up to order k and is computed as:

$$Q_{LB} = T(T+2) \sum_{j=1}^{k} \frac{\tau_j^2}{T-J} \tag{11.24}$$

where τ_j is the j-th autocorrelation and T is the number of observations. If the series is not based upon the results of ARIMA estimation, then under the null hypothesis, Q is asymptotically distributed as a χ^2 with degrees of freedom equal to the number of autocorrelations. If the series represents the residuals from ARIMA estimation, the appropriate degrees of freedom should be adjusted to represent the number of autocorrelations less the number of AR and MA terms previously estimated. Note also that some care should be taken in interpreting the results of a Ljung-Box test applied to the residuals from an ARMAX specification (see Dezhbaksh, 1990, for simulation evidence on the finite sample performance of the test in this setting).

The Q-statistic is often used as a test of whether the series is white noise. There remains the practical problem of choosing the order of lag to use for the test. If you choose too small a

lag, the test may not detect serial correlation at high-order lags. However, if you choose too large a lag, the test may have low power since the significant correlation at one lag may be diluted by insignificant correlations at other lags. For further discussion, see Ljung and Box (1979) or Harvey (1990, 1993).

Long-run Variance

You may compute estimates of the long-run variance of a series using this view. See Appendix F. "Long-run Covariance Estimation," beginning on page 1115 of *User's Guide II* for a discussion of the long-run variance and role of this statistic in modern econometric analysis.

The group object version of this view, which is almost identical to the series object version, is described in detail in "Long-run Covariance" on page 604.

Unit Root Test

This view carries out the Augmented Dickey-Fuller (ADF), GLS transformed Dickey-Fuller (DFGLS), Phillips-Perron (PP), Kwiatkowski, *et. al.* (KPSS), Elliot, Richardson and Stock (ERS) Point Optimal, and Ng and Perron (NP) unit root tests for whether the series (or it's first or second difference) is stationary.

In the panel context, this view carries out Levin, Lin and Chu, Breitung, Im, Pesaran and Shin, Fisher-type tests using ADF and PP tests (Maddala and Wu, and Choi), and Hadri panel unit root tests.

See "Unit Root Testing" on page 589 of *User's Guide II* for a discussion of stationary and nonstationary time series and additional details on how to carry out the unit roots tests in EViews.

Breakpoint Unit Root Test

This view carries out unit root tests which allow for a structural break in the trend process (Perron, 1989). EViews offers support for several types of modified augmented Dickey-Fuller tests which allow for levels and trends that differ across a single break date. You may compute unit root tests with a single break under various specifications for the break.

See "Unit Root Tests with a Breakpoint" on page 601 of *User's Guide II* for details.

Variance Ratio Test

The variance ratio test view EViews allows you to perform the Lo and MacKinlay variance ratio test to determine whether differences in a series are uncorrelated, or follow a random walk or martingale property.

See "Variance Ratio Test" on page 627 of *User's Guide II* for a discussion of the Lo and MacKinlay test and other variance ratio tests that you may perform using the series view.

BDS Independence Test

This view carries out the BDS test for independence, as described in Brock, Dechert, Scheinkman and LeBaron (1996).

The BDS test is a portmanteau test for time based dependence in a series. It can be used for testing against a variety of possible deviations from independence, including linear dependence, non-linear dependence, or chaos. The test can be applied to a series of estimated residuals to check whether the residuals are independent and identically distributed (*iid*). For example, the residuals from an ARMA model can be tested to see if there is any non-linear dependence in the series after the linear ARMA model has been fitted.

See "BDS Independence Test" on page 636 of *User's Guide II* for a description of the BDS test.

Forecast Evaluation

When constructing a forecast of future values of a variable, economic decision makers often have access to different forecasts; perhaps from different models they have created themselves or from forecasts obtained from external sources. When faced with competing forecasts of a single variable, it can be difficult to decide which single or composite forecast is "best". Fortunately, EViews provides tools for evaluating the quality of a forecast which can help you determine which single forecast to use, or whether constructing a composite forecast by averaging would be more appropriate.

Methodology

Evaluation of the quality of a forecast requires comparing the forecast values to actual values of the target value over a forecast period. A standard procedure is to set aside some history of your actual data for use as a *comparison sample* in which you will compare of the true and forecasted values.

EViews allows you to use the comparison sample to: (1) construct a forecast evaluation statistic to provide a measure of forecast accuracy, and (2) perform Combination testing to determine whether a composite average of forecasts outperforms single forecasts.

Forecast Evaluation Statistics

EViews offers four different measures of forecast accuracy; RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), and the Theil Inequality Coefficient. These statistics all provide a measure of the distance of the true from the forecasted values.

Suppose the forecast sample is $j = T + 1, T + 2, \ldots, T + h$, and denote the actual and forecasted value in period t as y_t and $\hat{y}_t$, respectively. The forecast evaluation measures are defined as:

Root Mean Squared Error	$\sqrt{\displaystyle\sum_{t=T+1}^{T+h} (\hat{y}_t - y_t)^2 / h}$
Mean Absolute Error	$\displaystyle\sum_{t=T+1}^{T+h} |\hat{y}_t - y_t| / h$
Mean Absolute Percentage Error	$100 \displaystyle\sum_{t=T+1}^{T+h} \left| \frac{\hat{y}_t - y_t}{y_t} \right| / h$
Theil Inequality Coefficient	$\dfrac{\sqrt{\displaystyle\sum_{t=T+1}^{T+h} (\hat{y}_t - y_t)^2 / h}}{\sqrt{\displaystyle\sum_{t=T+1}^{T+h} \hat{y}_t^2 / h} + \sqrt{\displaystyle\sum_{t=T+1}^{T+h} y_t^2 / h}}$

Combination Tests

To test whether an average, or combination, of the individual forecasts may perform better than the individual forecasts themselves, EViews offers the Combination Test, or Forecast Encompassing Test of Chong and Hendry (1986) and refined by Timmermann (2006). The idea underlying this test is that if a single forecast contains all information contained in the other individual forecasts, that forecast will be just as good as a combination of all of the forecasts. A test of this hypothesis can be conducted by performing a regression of the model:

$$Y_{t+h} - \hat{Y}_{t+h,i} = \beta_0 + \sum_{j \neq i}^{N} \beta_j \hat{Y}_{t+h,j} \tag{11.25}$$

where Y_{t+h} is the vector of actual values over the forecast period and $\hat{Y}_{t+h,i}$ is the vector of forecast values over the same period for forecast i. A test for whether forecast i contains all the information of the other forecasts may be performed by testing whether $\beta_j = 0, \forall (j \neq i)$; if the difference between the true values and the forecasted values from forecast i is not related to the forecasts from all other models, then forecast i can be used individually. If the differences are affected by the other forecasts, then the latter forecasts should be included in the formation of a composite forecast.

Diebold-Mariano Test

The Diebold-Mariano test is a test of whether two competing forecasts have equal predictive accuracy. For one-step ahead forecasts, the test statistic is computed as:

$$S = \frac{\bar{d}}{s_d} \tag{11.26}$$

where

$$d = L_1 - L_2 \tag{11.27}$$

and L_i, $i = 1, 2$ is either a squared or absolute difference between the forecast and the actual,

$$L_i = (\hat{y}_i - y)^2 \tag{11.28}$$

or

$$L_i = \left| \hat{y}_i - y \right| \tag{11.29}$$

where $\bar{d}$ and s_d are the mean and sample standard deviation of d. (Note that while Diebold and Mariano define an n-step statistic, EViews only computes the one-step version.)

Following Harvey, Leybourne, and Newbold (HLN), EViews calculates the standard deviation using a small-sample bias corrected variance calculation.

The test-statistic follows a Student's t-distribution with $T - 1$ degrees of freedom.

EViews will only display the Diebold-Mariano test statistic if exactly two forecasts are being evaluated.

Forecast Evaluation in EViews

To perform forecast evaluation in EViews, you must have a series containing the observed values of the variable for which you wish to evaluate forecasts. To begin, open up the series and click on **View/Forecast Evaluation...**, which will open the **Forecast Evaluation** dialog box:
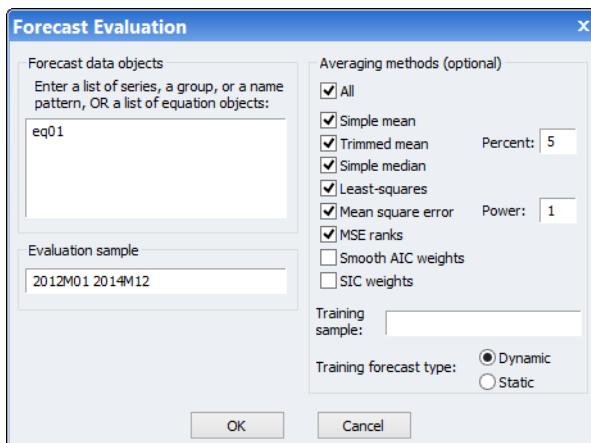
The **Forecast data objects** box specifies the forecasts to be used for evaluation. Forecasts can be entered either as a collection of series (in which case the names of the series, a series naming pattern, or the name of a group are entered), or as a list of equation objects. If equation objects are entered, EViews will automatically perform a dynamic forecast over the forecast period from each of those equation objects to generate the forecast data.

When using equation objects, rather than forecast series, as the forecast data, the following should be noted:

- Each equation must have an identical dependent variable, which is identical to the series from which you are performing the forecast evaluation. i.e., if you are forecasting from series Y, each equation must have Y as the dependent variable. Currently transformations (such as LOG(Y)) are not allowed.

- If using smoothed AIC or BMA/SIC averaging methods, the weight calculations are only strictly valid if the underlying estimation objects were estimated on identical samples. It is up to the user to ensure that the samples are identical.

- Only equation objects are allowed. If a different type of estimation object (system, VAR, Sspace, *etc.*) is used, or if forecast was obtained from a non-EViews estimation source, the forecasts cannot be specified by equation.

- If using one of the MSE based or the OLS based weighting methods, historical forecasts (along with actual values) are needed for use in the weighting calculation. Note that EViews will not re-estimate the equations, it will use the same coefficient values for both the historical forecast and the actual forecasts, based on whatever sample was used when the equation were originally estimated. If you wish to use different estimation samples for the comparison forecast and actual forecast, you

must perform the estimation and forecasts manually and specify the forecast data by series.

The **Evaluation sample** box specifies the sample over which the forecasts will be evaluated.

The **Averaging methods (optional)** area selects which forecast averaging methods to evaluate. For more details on each averaging method, see FORECAST AVERAGING ENTRY. If the **Trimmed mean** averaging method is selected, the **Percent:** box specifies the level of trimming (from both ends). If the **Mean square error** method is selected, the Power: box specifies the power to which the MSE is raised.

Note the **Smooth AIC weights** and **SIC weights** options are only available if a list of equations is entered in the **Forecast data objects** box, since they require information from the estimation rather than just the raw forecast data.

The Least-squares, Mean square error, MSE ranks, Smooth AIC weights, and SIC weights averaging methods require a training sample - a sample over which the averaging weights are computed. If any of these averaging methods are selected, a sample must be entered in the **Training sample** box. If a list of equations is entered in the **Forecast data objects** box, the **Training forecast type** radio buttons select which type of forecast is used over the training sample.

An Example

As an example of forecast evaluation in EViews, we evaluate six monthly forecasts of electricity demand in England and Wales, using the workfile "elecdmd.wf1". This workfile contains monthly electricity demand data from April 2005 until April 2014 (in the series ELECDMD), along with five evaluation sample forecasts of electricity demand (series ELECF_FE1–ELECF_FE5), and five out-of-sample forecasts (series ELECF_FF1–ELECF_FF5). The different forecast series correspond to different five different models used to generate forecasts.

Each of the evaluation sample forecast series contains actual data until December 2011, and then forecast data from January 2012 until December 2013.

We will evaluate the five models' forecast accuracy using the evaluation sample forecast series. To begin, we open the ELECDMD series and click on **View/Forecast Evaluation...** and enter the names of our forecast series in the **Forecast data** objects box of the **Forecast Evaluation** dialog:
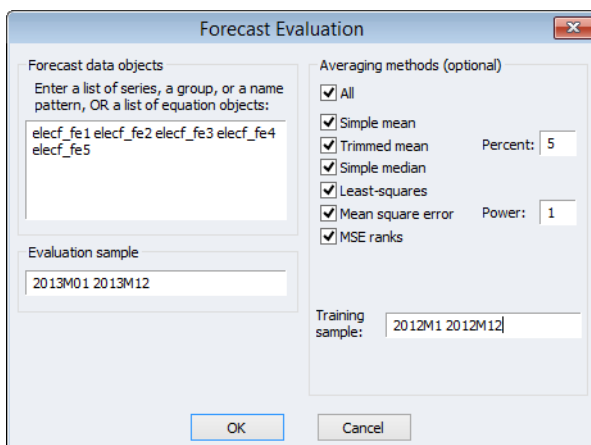
We set the evaluation sample to "2013M1 2013M12", giving us twelve months of forecasts to evaluate. We choose to evaluate each of the available averaging methods, and set the training sample for the **Least-squares**, **Mean square error** and **MSE ranks** methods to be "2012M1 2012M12".

The top of the output provides summary information about the evaluation performed, including the time and date it was performed, the number of observations included (12 in this case) and the number of forecasts evaluated, including the averaging methods.

```
Forecast Evaluation
Date: 03/07/15   Time: 21:21
Sample: 2013M01 2013M12
Included observations: 12
Evaluation sample: 2013M01 2013M12
Training sample: 2012M01 2012M12
Number of forecasts: 10
```

The "Combination tests" section displays the results of the combination test for each of the individual forecasts. In our case the null hypothesis is non-rejected for each of the forecasts, other than the first, which is rejected at a 5% level.

Combination tests
Null hypothesis: Forecast i includes all information contained in others

Forecast	F-stat	F-prob
ELECF_FE1	4.138355	0.0495
ELECF_FE2	1.443315	0.3146
ELECF_FE3	1.533069	0.2913
ELECF_FE4	1.558450	0.2851
ELECF_FE5	0.420428	0.7898

The "Evaluation statistics" section shows the RMSE, MAE, MAPE and Theil statistics for each of the five forecasts, along with the five averaging methods. The trimmed mean averaging method could not be calculated with only 5 forecast series.

Evaluation statistics

Forecast	RMSE	MAE	MAPE	Theil
ELECF_FE1	329.3182	299.3388	6.582262	0.034294
ELECF_FE2	108.1292	94.06769	2.046802	0.011568
ELECF_FE3	219.1630	193.9669	4.293012	0.024040
ELECF_FE4	154.1023	126.3301	2.793056	0.016757
ELECF_FE5	145.5928	123.5517	2.692389	0.015454
Simple mean	103.3829	89.38450	1.940429	0.011073
Simple median	108.1292	94.06769	2.046802	0.011568
Least-squares	862.9722	858.1411	18.75875	0.084943
Mean square error	98.15188	80.04863	1.759876	0.010547
MSE ranks	99.04570	84.72580	1.855211	0.010624

EViews has shaded the forecast or averaging method that performed the best under each of the evaluation statistics. In our case the MSE ranks method outperforms every other forecast or averaging method in each of the evaluation criteria.

EViews also produces a graph of each of the individual forecasts, the averages, and the actual values over the training and evaluation periods, allowing a quick visual comparison of each:
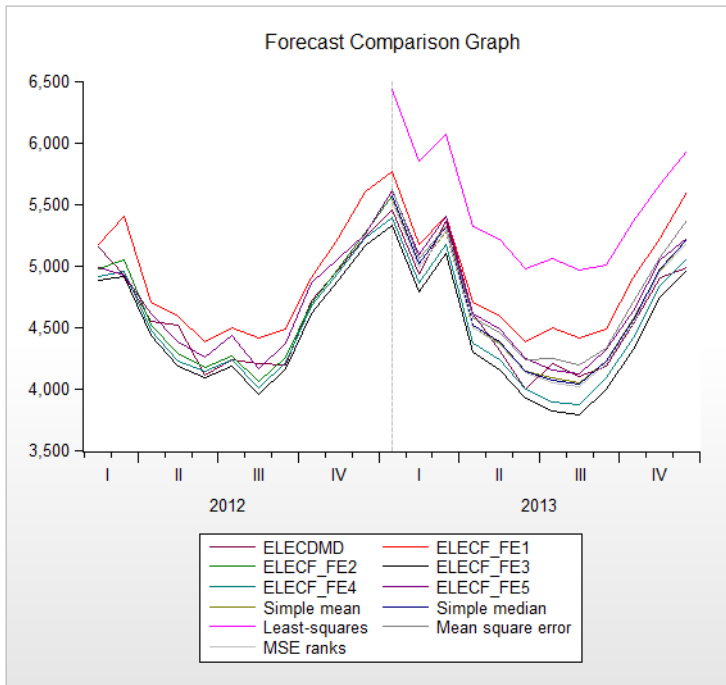
Label

This view displays a description of the series object.

You can edit any of the field cells in the series label, except the **Last Update** cell which displays the date/time the series was last modified. Each field contains a single line, except for the **Remarks and History** fields which can contain up to 20 comment lines. Note that if you insert a line, the last (of the 20) line of these fields will be deleted.
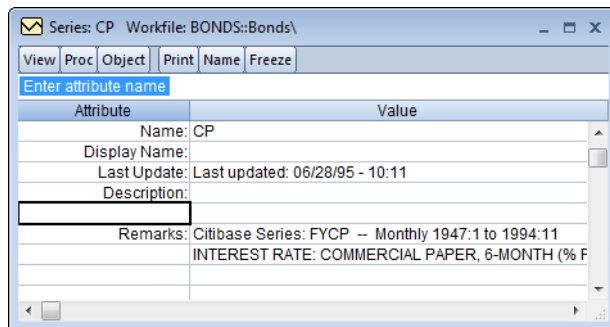
The **Name** is the series name as it appears in the workfile; you can rename your series by editing this cell. If you fill in the **Display Name** field, this name may be used in tables and

graphs in place of the standard object name. Unlike ordinary object names, **Display Names** may contain spaces and preserve capitalization (upper and lower case letters).

You may click in the left-hand cell above the **Remarks** field to define a custom attribute for use in the workfile details view and for database search. Simply enter the name of your attribute in the left-hand cell, and the attribute value in the right-hand cell. See "Adding Custom Attributes in the Label View," on page 117 for discussion.

See Chapter 10. "EViews Databases," on page 317 for further discussion of label fields and their use in Database searches.

Properties

Clicking on the **Properties** button on the series toolbar provides access to the dialog controlling various series properties.

There are several tabs in the dialog. The first tab, labeled **Display**, allows you to set the default display characteristics for the series (see "Changing the Spreadsheet Display" on page 124). The **Values** tab may be used to define or modify a formula, turning the series into an auto-updating series, or to freeze the series values at their current levels (see "Defining an Auto-Updating Series" on page 204). The last **Value Map** tab should be used to assign value maps to the series (see "Value Maps" on page 219).

In dated workfiles, the **Freq Conversion** tab will also be displayed. You may use this tab to set the default frequency conversion settings for the series. Recall that when you fetch a series from an EViews database or when you copy a series to a workfile or workfile page with a different frequency, the series will automatically be converted to the frequency of the destination workfile. The conversion options view allows you to set the method that will be used to perform these conversions (see "Reading EViews Data using Other Applications" on page 164).
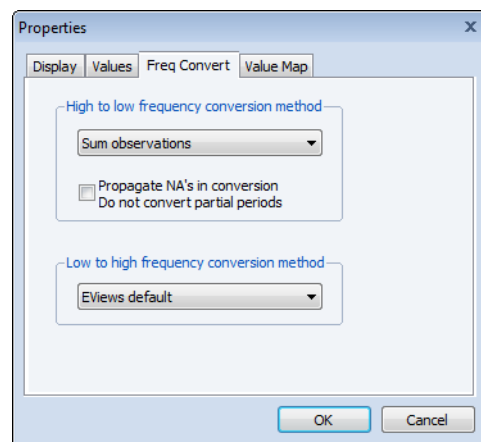
Each series has a default up and down frequency conversion method. By default, the series will take its settings from the EViews global options (see "Frequency Conversion" on page 864). This default series setting is labeled **EViews default**. You may, of course, override these settings for a given series. Here, instead of using the global defaults, the high to low conversion method is set to **Sum observations** without propagating NAs.

Series Adjust

EViews offers an adjust mode which may be enabled by toggling the **Adjust + /-** button on the spreadsheet toolbar (you may need to widen the object window to see this button).

When in adjust mode, EViews will add additional columns to the spreadsheet view of the series. The first new column will be titled "Unadjusted" and will contain a copy of the series values *at the moment you entered adjust mode.* This "Unadjusted" column is fixed and lets you easily see the original data in the series to compare any changes you subsequently make while in adjust mode.

Adjust mode has added two additional columns to the spreadsheet. The first contains "Delta", which shows the raw difference between the original "Unadjusted" series and the current values in the series. The second contains "Delta%", which is the percentage difference. Note that for ease of use, in cases where there is no difference between the two series, the "Delta" and "Delta%" columns are left blank instead of showing "0". Hence when first entering adjust mode, since no adjustments have yet been made, both columns are blank.

To adjust the series while in adjust mode, you can edit any of the cells in the raw series column (the left most column), or edit the "Delta" or "Delta%" columns. For example, if you want to edit the first value of the series to be "3", simply type the number three in the first column. If you want the value to be 3 greater than the original value, a quick way to do it is to simply enter "3" in the "Delta" column. Similarly, a quick way to make the series 10% higher than its original value is to type "10" in the Delta% column. Similar to when you are in Edit mode, you may select multiple cells and modify the values using array expression.

See Appendix B. "Enhanced Spreadsheet Editing," on page 885 of *User's Guide I* for additional tools for spreadsheet editing and examples. These tools allow you to perform sophisticated transformations of your data and to see the resulting changes in an interactive fashion.

Crucially, unlike simple editing of a series in which changes are permanent, when you exit adjust mode, either by toggling the **Adjust + /-** button, or by closing the series, EViews will ask if you would like to keep the changes you made. You should note that any changes made while in adjust mode are "live", meaning that any operations performed using that series will use the current values, but exiting adjust mode offers you the chance to revert the series back to the pre-adjust values. This behavior allows you to make changes to a series to perform quick "what if" analysis without permanently changing the series.

Series Procs Overview

Series procedures may be used to generate new series that are based upon the data in the original series. You may generate new series using expressions, or you may generate series by classifying the original series.

When working with numeric series, you may also use series procs to resample from the original series, to perform seasonal adjustment or exponential smoothing, to filter the series using the Hodrick-Prescott or band-pass filters, or to output data used in constructing various distribution graphs such as histograms and kernel density plots.

Generate by Equation...
Generate by Classification...
Resample...
Interpolate...
Seasonal Adjustment ▶
Exponential Smoothing ▶
Hodrick-Prescott Filter...
Frequency Filter...
Make Whitened...
Make Distribution Plot Data...
Add-ins ▶

For alpha series you may use a series proc to make a val-mapped numeric series.

EViews will create a new numeric series and valmap so that each value in the numeric series is mapped to the original alpha series value.
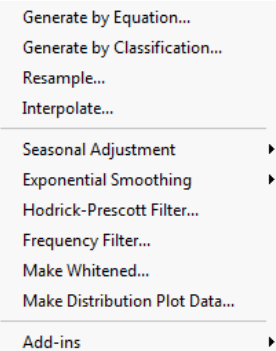
Make mapped series

Generate by Equation

This is a general procedure that allows you to create new series by using expressions to transform the values in the existing series. The rules governing the generation of series are explained in detail in "Series Expressions" on page 181.

It is equivalent to using the `genr` command.

Generate by Classification

The series classification procedure generates a categorical series using ranges, or bins, of values in the numeric source series. You may assign individuals into one of k classes based any of the following: equally sized ranges, ranges defined by quantile values, arbitrarily defined ranges. A variety of options allow you to control the exact definition of the bins, the method of encoding, and the assignment of value maps to the new categorical series.

We illustrate these features using data on the 2005 Academic Performance Index (API) for California public schools and local educational agencies ("Api05btx.WF1"). The API is a numeric index ranging from 200 to 1000 formed by taking the weighted average of the student results from annual statewide testing at grades two through eleven.

The series API5B contains the base API index. Open the series and select **Proc/Generate by Classification...** to display the dialog. For the moment, we will focus on the **Output** and the **Specification** sections.

Output

In the **Output** section you will list the name of the target series to hold the classifications, and optionally, the name of a valmap object to hold information about the mapping. Here, we will save the step size classification into the series API5B_CT and save the mapping description in API5B_MP. If the classification series already exists, it will be overwritten; if an object with the map name already exists, the map will be saved in the next available name ("API5B_MP01", *etc.*).

Specification

The **Specification** section is where you will define the basic method of classification. The **Method** dropdown allows you to choose from the four methods of defining ranges: **Step Size**, **Number of Bins**, **Quantile Values**, **Limit Values**. The first two methods specify equal sized bins, the latter two define variable sized bins.

Step Size

We will begin by selecting the default **Step Size** method and entering "100" and "200" for the **Step size** and **Grid start** edit fields. The step size method defines a grid of bins of fixed size (the step size) beginning at the specified grid start, and continuing through the grid end. In this example, we have specified a step size of 100, and a **Grid start** value of 200. The

Grid end is left blank so EViews uses the data maximum extended by 5%, ensuring that the rightmost bin extends beyond the data values. These settings define a set of ranges of the form: [100, 200), [200, 300), ..., [1000, 1100). Note that by default the ranges are closed on the left so that we say x lies in the first bin if $100 \le x < 200$.

Click on **OK** to accept these settings, then display the spreadsheet view of API5B_CT. We see that observations 1 and 3 fall in the [500, 600) bin, while observations 4 and 5 fall in the [400, 600) bin. Observations 2 and 6 were NAs in the original data and those values have been carried over to the classification.

Keep in mind that since we have created both the classification series and a value

map, the values displayed in the spreadsheet are mapped values, not the underlying data. To see the underlying classification data, you may go to the series toolbar and change the **Default** setting to **Raw Data**.

Opening the valmap API5B_MP, we see that the actual data in API5B_CT are integer values from 1 to 9, and that observations 1 and 3 are coded as 4s, while observations 4 and 5 are coded as 3s.

Number of Bins

The second method of creating equal sized bins is to select **Number of Bins** in the **Method** dropdown. The label for the second edit field will change from "Bin size" to "# of bins", prompting you for an

integer value k . EViews will define a set of bins by dividing the grid range into k equal sized bins. For example, specifying 9 bins beginning at 200 and ending at 1100 generates a classification that is the same as the one specified using the step size of 100.

Quantile Values

One commonly employed method of classifying observations is to divide the data into quantiles. In the previous example, each school was assigned a value 1 to 9 depending on which of 9 equally sized bins contained its API. We may instead wish to assign each school an index for its decile. In this way we can determine whether a given school falls in the lowest 10% of schools, second lowest 10%, *etc.*

To create a decile classification, display the dialog, select **Quantile Values** from the **Method** dropdown, and enter the number of quantile values, in this case "10".

We see that the first 4 (non-NA) values are all in the first decile (< 583.8), while observations 7 and 8 lie in the eighth decile [780, 815). As before, these values are the mapped values; the underlying values are encoded with integer values from 1 to 10.

It is worth emphasizing that the mapped values are text representations of the quantile values, akin to labels, and will generally not be displayed in full precision.

Limit Values

You may also define your bins by providing an arbitrary set of two or more limit points. Simply select **Limit Values** from the **Method** dropdown and provide a list of numeric values, scalars, or vectors.
EViews will sort the numbers and define a set of bins using those limits.

Options

EViews provides various options that allow you to fine tune the classification method or to alter the encoding of classification values.

Encoding

The dropdown menu labeled **Encoding** allows you to select different methods of assigning values to the classified observations.
By default, EViews classifies observations using the integers 1, 2, *etc.* so that the observations falling in the first bin are assigned the value 1, observations in the second bin are assigned 2, and so forth.
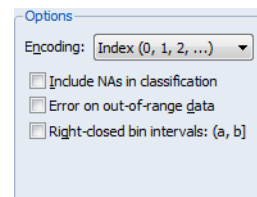
In addition to the default **Index (0, 1, 2,...)** method, you may elect to use the **Left edges of bins**, the **Right edges of bins**, or the **Midpoints of bins**. Each of these encoding methods

should be self-explanatory. Note that index encoding is the only method available for classification by quantile values.

Value maps are not created for classifications employing non-index encoding.

NA classification

By default, observations in the original series which are NA are given the value NA in the classification series. If you treat the NA as a category by checking **Include NAs in classification**, EViews will assign NAs the index value of 0 in the classification, and will map this value to the label "NA".

For example, re-running our first example (step size of 100, a grid start value of 200) but keeping NAs, we have the following *raw* data. Note that the mapped values will be identical to those from the earlier example, so we have switched the series to **Raw Data** to see the values for NA:

We see that the observations 2 and 6, which were missing values in the original data, are encoded with the index value 0.

Out-of-Range Values

You may instruct EViews to generate an error if it encounters an observation that does not fall into one of the specified bins; by default, observations that lie outside the bin range are simply assigned an NA.

This option is irrelevant for quantile value classification.

Right-closed Bins

Bins are left-closed by default, so that x lies in the bin defined by a and b if $a \leq x < b$. To change the non-strict inequality from the left to the right $a < x \leq b$, you must instruct EViews to use right-closed bins by checking the box **Right-closed bin intervals (a, b]**. This setting should have little effect, if any, if your data are reasonably continuous.

A Couple of Warnings

In determining whether an observation is in a particular interval, EViews performs comparisons of real numbers to assess, for example, whether $a \le x$ and whether $x < b$. We urge you to exercise caution in cases where a and b are finite precision representations of real numbers.

To take a specific example, suppose that we wish to divide the interval from 0 to 1 into bins of size 0.1 (so that our limit points are 0.1, 0.2, 0.3, *etc.*). Since 0.1 cannot be represented in floating point by a computer, comparisons will be made with numbers that are close to, but not exactly equal to 0.1. As a result, when there are data values approximately equal to the limit points, classifications may behave unexpectedly since they are based on a comparison of two floating point numbers.

A related issue occurs when you classify by number of bins, and set the start or end values equal to the data minimum or maximum. In this situation, depending upon whether you have selected left or right-closed intervals, observations with values equal to the start or end may fall out-of-range.

To illustrate this issue, we extend our example by assuming that we wish to divide the 0–1 range into right-closed bins. Each bin will be of the form $[0.1(i-1), 0.1i)$. with the first interval given by $(0, 1]$. Now consider classifying those data values that are exactly equal to 0, and note that these values are out-of-range since they lie outside the first interval. The same is true for observations equal to 1 if we have defined left-closed bins. The comparison is more complex if the relevant endpoint is real valued since the out-of-range status would depend upon a floating point comparison.

The obvious recommendation in this latter case are that: (1) you specify at most only one of range start and range end at the data extremes, and (2) if you set either the start or end to the corresponding data extreme, you define the intervals so that they are closed on that end (*i.e.,* starts that equal the minimum have left-closed intervals, and ends that equal the maximum have right-closed intervals) and you set the range so that it extends past the other extreme. Our first example above adopts this strategy, setting the low value to the data minimum (200), setting the step size, and leaving the upper limit unspecified.

More generally, we urge you to exercise caution when defining intervals with real-valued limits.

Resample

The series resampling procedure selects from the observations in a series to create a new series (the resampled series). You may draw your new sample with replacement (allow a given observation to be drawn multiple times) or without replacement. When you select **Proc/Resample...** from the series window, you will be prompted to specify various options.

Input Sample

Describes the sample from which observations are to be drawn. The default is the current workfile sample.

If you select the **Draw without replacement** option, each row will be drawn at most once. This option requires the input sample to be at least as large as the output sample. If you do not select this option, each row will be drawn *with* replacement.

Output Sample

Specifies the sample into which the resampled series will be saved. Any value outside the output sample will not be changed. The default output sample is the current workfile sample. If you select the **Draw without replacement** option, the output sample cannot be larger than the input sample.

NA Handling

The default **Include NAs in draws** instructs EViews to draw from every observation in the input sample, including those that contain missing values. Alternatively, you may select the **Exclude NAs from draws** option so that you draw only from observations in the input sample that do not contain any missing values. Finally, the **Exclude NAs from draws but copy NA rows to output** option first copies matching observations in the input sample that contain missing values to the output sample. The remaining rows of the output sample are then filled by drawing from observations in the input sample that do not contain any missing values. This option keeps observations with missing values fixed and resamples those that do not contain any missing values.

Series Name

The new series will be named using the specified series name. You may provide a series name or a wildcard expression. If you use a wildcard expression, EViews will substitute the existing series name in place of the wildcard. For example, if you are sampling from the series X and specify "*_SMP" as the output series, EViews will save the results in the series X_SMP. You may not specify a destination series that is the same as the original series.

If another series with the specified name exists in the workfile, the existing values in the output sample will be overwritten with the resampled values. Any values outside the output sample will remain unchanged. If there is a non-series object with the specified name, EViews will return an error message.

Because of these naming conventions, your original series cannot be an auto-series. For example, if the original series is X(-1) or LOG(X), EViews will issue an error. You will have to generate a new series, say by setting XLAG = X(-1) or LOGX = LOG(X), and then resample from the newly generated series.

Weighting

By default, the procedure draws from each row in the input sample with equal probabilities. If you want to attach different probabilities to the rows (importance sampling), you can specify a name of an existing series that contains weights that are proportional to the desired probabilities in each row. The weight series must have non-missing non-negative values in the input sample, but the weights need not add up to 1 since EViews will normalize the weights.

Block Length

By default, sets the *block length* to 1, meaning that we draw one observation at a time from the input sample. If you specify a block length larger than 1, EViews will draw blocks of consecutive rows of the specified length. The blocks drawn in the procedure form a set of *overlapping moving* blocks in the input sample. The drawn blocks will be appended one after the other in the output series until it fills the output sample (the final block will be truncated if the block size is not an integer multiple of the output sample size). Block resampling with a block length larger than 1 makes the most sense when resampling time series data.

Block resampling requires a continuous output sample. Therefore a block length larger than 1 cannot be used when the output sample contains "gaps" or when you have selected the **Exclude NAs from draws but copy NA rows to output** option. If you choose the **Exclude NAs from draws** option and the block length is larger than 1, the input sample will shrink in the presence of NAs in order to ensure that there are no missing values in any of the drawn blocks.

Interpolate

The interpolation procedure for a series fills in missing values, or NAs, within a series by interpolating from values that are non-missing. EViews offers a number of different algorithms for performing the interpolation: Linear, Log-Linear, the Catmull-Rom Spline, and the Cardinal Spline.

The Linear interpolation method simply computes a linear approximation based on the previous non-missing value and the next non-missing value. The interpolated value is then calculated as:

$$IV_{Lin} = (1 - \lambda)P_{i-1} + \lambda P_{i+1} \tag{11.30}$$

where P_{i-1} is the previous missing value, P_{i+1} is the next non-missing value, and λ is the relative position of the missing value divided by the total number of missing values in a row.

Thus if you have a single NA, then the interpolated value will be half way between the previous value and the next value. If you have two NAs in a row, the first will be interpolated as 1/3 of the distance between the previous value and the next. The second value will be interpolated as 2/3 of the distance.

Log-linear interpolation is calculated in the same way as the Linear method, but the series values are logged before the calculation is made, then the interpolated value is exponentiated. If the source series contains negative values before and after the missing values, then the absolute value is used for interpolation, with the result being negated. If the values to either side of the missing values contain both a negative and a positive number, the interpolation routine will return an NA. Thus the interpolated value is calculated as:

$$IV_{pos} \ = \ EXP[(1-\lambda)\log(P_{i-1}) + \lambda\log(P_{i+1})] \tag{11.31}$$

$$IV_{neg} \ = \ -EXP[(1-\lambda)\log(-P_{i-1}) + \lambda\log(-P_{i+1})] \tag{11.32}$$

Note that both the linear and log-linear routines will return an NA if there are no previous or next non-missing values available – i.e. if the first observation in the sample is an NA, it will not be interpolated since there is no previous value available.

The Cardinal Spline is based on the previous two non-missing values, and the next two non-missing values, and tries to fit the missing data to a non-linear, or curved, pattern. Denoting P_{i-2} and P_{i-1} as the previous two non-missing values, and P_{i+1} and P_{i+2} as the next two non-missing values, the interpolated value is calculated as:

$$IV_{CS} \ = \ (2\lambda^3 - 3\lambda^2 + 1)P_{i-1} + (1-t)(\lambda^3 - 2\lambda^2 + \lambda)(P_{i+1} - P_{i-2}) - \tag{11.33}$$
$$(2\lambda^3 - 3\lambda^2)P_{i+1} + (1-t)(\lambda^3 - \lambda^2)(P_{i+2} - P_{i-1})$$

where t is a called a tension parameter and affects the curvature of the spline.

The Catmull-Rom Spline is a special case of the Cardinal Spline, with the tension parameter, t, set at 0.

Note that as with the linear and log-linear routines, both spline routines will return an NA if either P_{i-1} or P_{i+1} are missing. However if P_{i-2} is missing, it is simply set equal to P_{i-1}, and if P_{i+2} is missing, it is set equal to P_{i+1}.

For example, the following series is interpolated using Linear and Log-Linear interpolation:

Obs	Data	λ	Linear	Log-Linear	Catmull-Rom
1	-1	--	-1	-1	-1
2	-0.8	--	-0.8	-0.8	-0.8
3	NA	1/2	-0.6	-0.56569	-0.4750
4	-0.4	--	-0.4	-0.4	-0.4
5	NA	1/3	-0.66667	-0.56769	-0.60741
6	NA	2/3	-0.93333	-0.83203	-0.90370
7	-1.2	--	-1.2	-1.2	-1.2

For this series, the first missing value is at observation 3. The linear interpolation of this value is simply 1/2 of the value of observation two (the previous non-missing observation) plus 1/2 of the value of observation four (the next non-missing observation), *i.e.* 1/2*-0.8 + 1/2*(-0.4). The second missing value is at observation five, and is equal to 2/3 of the value of observation four, plus (1-1/3) of the value of observation 7, *i.e.*, 2/3*(-0.4) + 1/3*(-1.2). The final interpolated value, at observation six, is calculated as 1/3*(-0.4) + 2/3*(-1.2)

For log-linear interpolation, the calculated value for observation three is equal to the negative of the exponent of 1/2 of the logged value at observation two (negated since it is negative) plus 1/2 of the value of observation five (negated):

$$-\exp\left(\frac{1}{2}\log(0.8) + \frac{1}{2}\log(0.4)\right) \tag{11.34}$$

The calculations for observations five and six, respectively, are:

$$-\exp\left(\frac{2}{3}\log(0.4) + \frac{1}{3}\log(1.2)\right) \tag{11.35}$$

$$-\exp\left(\frac{1}{3}\log(0.4) + \frac{2}{2}\log(1.2)\right) \tag{11.36}$$

For the Catmull-Rom spline, the calculated value for observation three is equal to:

$$\left(2\left(\frac{1}{2}\right)^3 - 3\left(\frac{1}{2}\right)^2 + 1\right)(-0.8) + \left(\left(\frac{1}{2}\right)^3 - 2\left(\frac{1}{2}\right)^2 + \frac{1}{2}\right)(-0.4 + 1) - \tag{11.37}$$
$$\left(2\left(\frac{1}{2}\right)^3 - 3\left(\frac{1}{2}\right)^2\right)(-0.4) + \left(\left(\frac{1}{2}\right)^3 - \left(\frac{1}{2}\right)^2\right)(-1.2 + 0.8) = -0.4750$$

The values at observations five and six are calculated similarly, although note that both for both observations, P_{i+1} and P_{i+2} are equal to -1.2.

To perform interpolation from within EViews, open up the series, then click on **Proc/Interpolate…** which will open up the **Interpolation** dialog. The I**nterpolation method** box lets you select which interpolation method you would like to use.

The **Output series name** box lets you enter a name for the output series in which the interpolated values will be stored. If left blank an untitled series will be created.
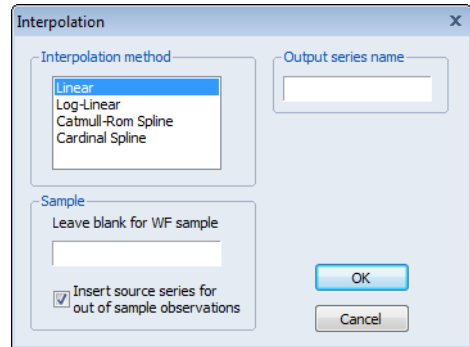
The **Sample** box lets you select a sample over which the interpolation will run. If left blank, the default workfile sample will be used. Note that observations outside of the selected sample will not be interpolated, nor will they be used as part of the interpolation calculations. The checkbox labeled **Insert source series for out of sample observations** controls whether the output series will be filled with the source series values for out of sample observations, or will be filled with NAs.

Seasonal Adjustment

Time series observed at quarterly and monthly frequencies often exhibit cyclical movements that recur every month or quarter. For example, ice cream sales may surge during summer every year and toy sales may reach a peak every December during Christmas sales. Seasonal adjustment refers to the process of removing these cyclical seasonal movements from a series and extracting the underlying trend component of the series.

The EViews seasonal adjustment procedures are available only for quarterly and monthly series. To seasonally adjust a series, click on **Proc/Seasonal Adjustment** in the series window toolbar and select the adjustment method from the submenu entries (**Census X-13**, **Census X-12**, **X-11 (Historical)**, **Tramo/Seats** or **Moving Average Methods**).
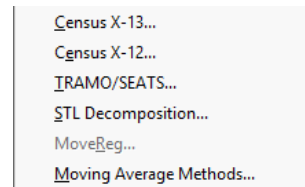
Census X-13

EViews offers an easy-to-use front-end for working with the U.S. Census Bureau's X-13 seasonal adjustment tools. The publicly available Census *X-13ARIMA-SEATS* seasonal adjustment program "X13.EXE" is included with your copy of EViews and is pre-installed in your EViews directory.

In addition to providing a wide range of new features, X-13 is capable of performing updated versions of X-11/X-12 or TRAMO/SEATS ARIMA seasonal adjustment. For backwards com-

patibility, EViews still supports the older X-11, X-12 and TRAMO/SEATS routines, but we recommend that you use the newer X-13 routines.

When you perform X-13 seasonal adjustment, EViews carries out the following steps:

- write out a specification file and data files for the series

- execute the X-13 program in the background, using the contents of the specification file

- read back the output file and saved data into your EViews workfile

The remainder of this section offers a brief description of the EViews interactive interface to *X-13ARIMA-SEATS*. EViews provides a general command interface that also supports X-13 features not available via the dialogs (see `Series::x13` (p. 630) in the *Object Reference*).

Users who require a detailed discussion of the X-13 procedures and capabilities should consult the Census Bureau documentation. The full documentation for the Census program, *X-13ARIMA-SEATS Reference Manual*, can be found in the "docs" subdirectory of your EViews installation directory in the PDF file "X-13 Reference Manual.pdf".

Note that X-13 combines standard least-squares regression, ARIMA, and regARIMA estimation, with X-11 or SEATS seasonal adjustment. To simplify matters, we will refer to regression, ARIMA, and regARIMA estimation collectively as the "ARIMA" or "ARIMA regression" portion of the X-13 seasonal adjustment procedure.

To perform X-13 seasonal adjustment, select **Proc/Seasonal Adjustment/Census X-13...** from the series window menu in a quarterly or monthly workfile. EViews will open a tree structured dialog containing branches for setting options for variables, ARIMA estimation and forecasting, seasonal adjustment, and output.

X-13 Variable Options

The *X-13ARIMA-SEATS* procedure allows you to perform ARIMA regression prior to the seasonal adjustment step. The first branch of the X-13 dialog includes options for defining the source series and exogenous variables used in the ARIMA step.

Transform

The **Transform** settings allow you to transform the source series and regressors before fitting the ARIMA model.

When you click on the **Transform** entry you will be prompted to choose a transformation method. You may choose between no transformation (**None**), or a natural log (**Log**), Box-Cox (**Box-Cox**), logistic (**Logit**), or automatically selected (**Auto**) transformation.
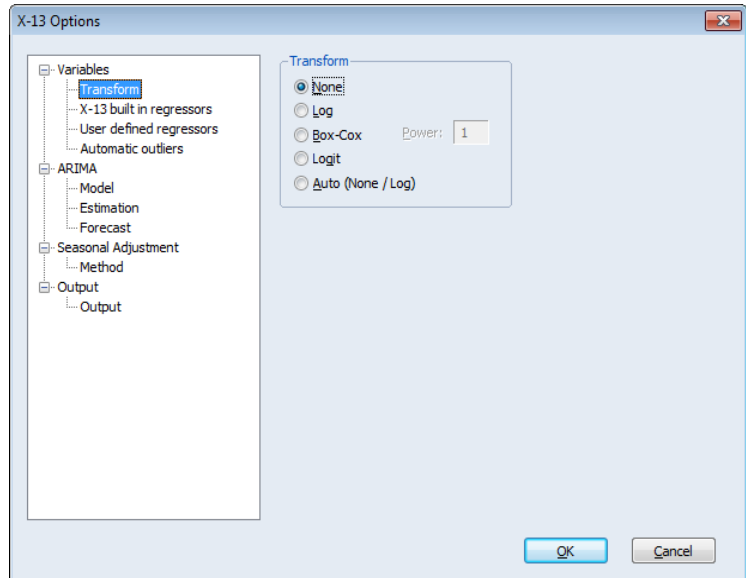
The **Auto** option chooses between no transformation and a log transformation based on the Akaike information criterion. The **Logit** option transforms the series y to $\log(y/(1-y))$ and is defined only for series with values that are strictly between 0 and 1. For the **Box-Cox** option, you must provide the parameter value λ for the transformation:

$$
\begin{cases}
\log(y) & \text{if } \lambda = 0 \\
\lambda^2 + (y^\lambda - 1)/\lambda & \text{if } \lambda \neq 0
\end{cases}
\tag{11.38}
$$

for $y > 0$. See "Transform" (Section 7.18) of the *X-13ARIMA-SEATS Reference Manual* for further details.

X-13 built-in regressors

The **X-13 built in regressors** page lets you specify any of the X-13 built-in regressors for the ARIMA regression model:
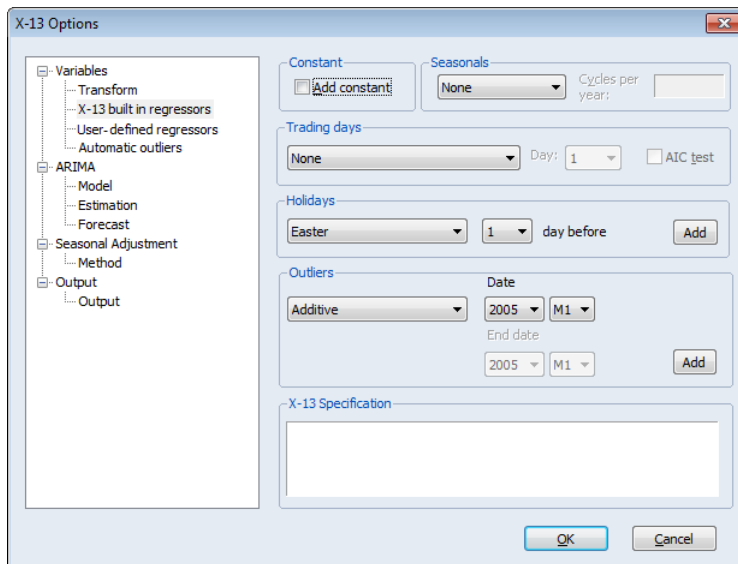
You may use the drop-down menus to add a constant, seasonal variables (such as seasonal dummies), trading-day effects, holidays or outliers to the ARIMA model. Only one type of seasonal variable and one type of trading-day effect can be selected, but multiple holidays or outliers are supported. To add a holiday or outlier, simply specify the variable to add in the appropriate section of the dialog and click on the **Add** button.

As you make selections or add variables, the **X-13 Specification** box will display the corresponding X-13 text specification. You may manually add, edit, or remove entries in this box and the dialog settings will change to reflect your entries, where relevant. For more details on the built-in variable types, see the "Variables" part of Section 7.13 ("Regression"), and Table 7.27 of the *X-13ARIMA-SEATS Reference Manual*.

X-13 user-defined regressors

The **User-defined regressors** page allows non-X-13 built-in variables to be used as regressors in the ARIMA model:
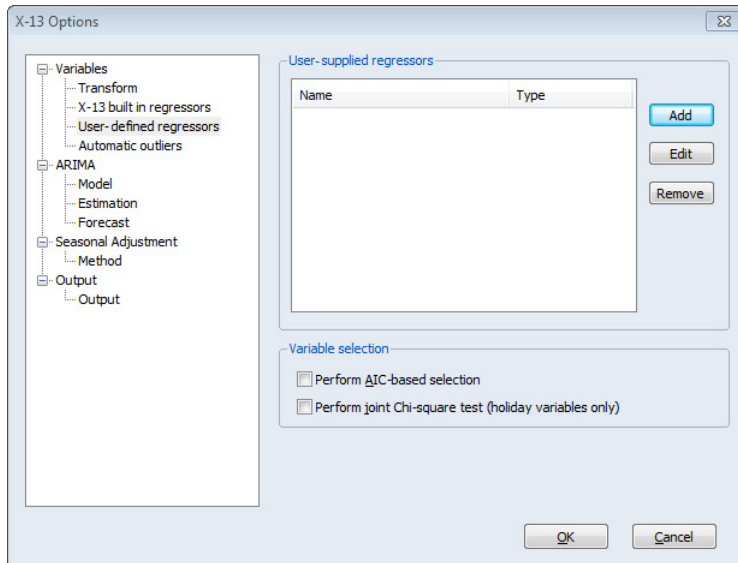
Clicking on the **Add** button brings up a dialog asking you to type in the name of the workfile variable that you would like to include as a regressor:
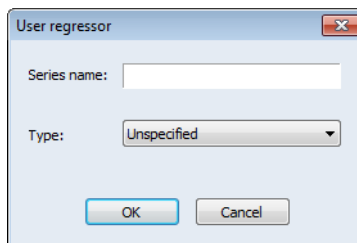
You may enter any valid EViews series name or expression, such as "X" or "log(X)", or "@pch(X)" in the **Series name** edit field.

The **Type** dropdown menu should be used to specify a regressor type for the variable you are including (*i.e.,* **Seasonal**, **Trading-day**, **Transitory**, **Holiday**). Knowledge of the variable type may be used in internal calculations. Note that you may define up to five different holiday types and you may include multiple variables of a given holiday type. If your regressor variable does not fit in any of the specific categories, you should designate it as **Unspecified**.

Once you have added user-defined regressors you may choose whether X-13 will perform model-selection for the specified variables using AIC-based selection or Chi-square tests by checking one or both of the **Variable selection** checkboxes.

For additional details on the elements of the **User-defined regressors** page, see Section 7.13 of the *X-13ARIMA-SEATS Reference Manual*, notably the entries for "user", "usertype", "aictest" and "chi2test".

Automatic Outliers

The **Automatic outliers** entry allows specification of automatic outlier detection routines.
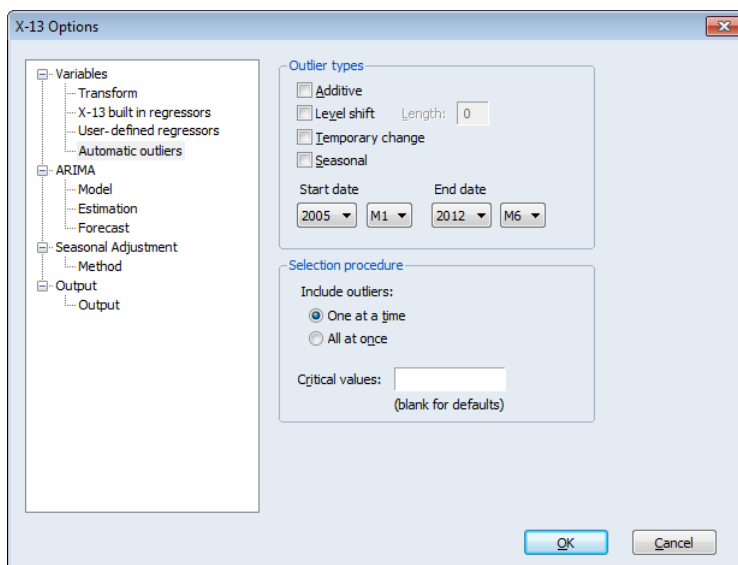
The controls in the **Outlier types** box may be used to select which type of outliers to detect, and the range of dates over which to detect them.

The **Selection procedure** box lets you specify the method used to select outliers (**One at a time** or **All at once)** and associated critical values against which the absolute *t*-statistics for the tests will be compared. You may enter a single critical value to apply to all of the outlier types, or a list of critical values, one for each type. The default settings, which are outlined in Table 7.22 of the *X-13ARIMA-SEATS Reference Manual*, vary with the number of observations.

Details of the outlier types and the selection procedures can be found in Section 7.11 of the *X-13ARIMA-SEATS Reference Manual*. The "DETAILS" section has a detailed description of the two different selection methods, "addone" (**One at a time)** and "addall" (**All at once**).

X-13 ARIMA Options

The **ARIMA** options branch is used to specify the ARIMA portion of the model, and to as provide access to basic estimation and forecasting options.

ARIMA Model

You will specify the ARIMA model to be estimated on the **Model** page, using the radio buttons in **ARIMA method** section:

- The default setting, **None**, tells X-13 not to include an ARIMA specification. (Note that SEATS seasonal adjustment requires an ARIMA specification, so you may not select **None** setting if you wish to use SEATS.)

- If you select **Manual**, **X11-Auto**, or **TRAMO Auto**, the dialog will change to show you the settings corresponding to your choice. We discuss each of these methods in turn.

Manual

If you select **Manual** as the ARIMA method, you will be prompted to specify the exact ARIMA specification in "(p, d, q) (P, D, Q)" format, where the "(p, d, q)" are the standard ARIMA components (for the AR, differencing and MA orders, respectively) and the "(P, D, Q)" are the seasonal ARIMA components at the workfile frequency.:
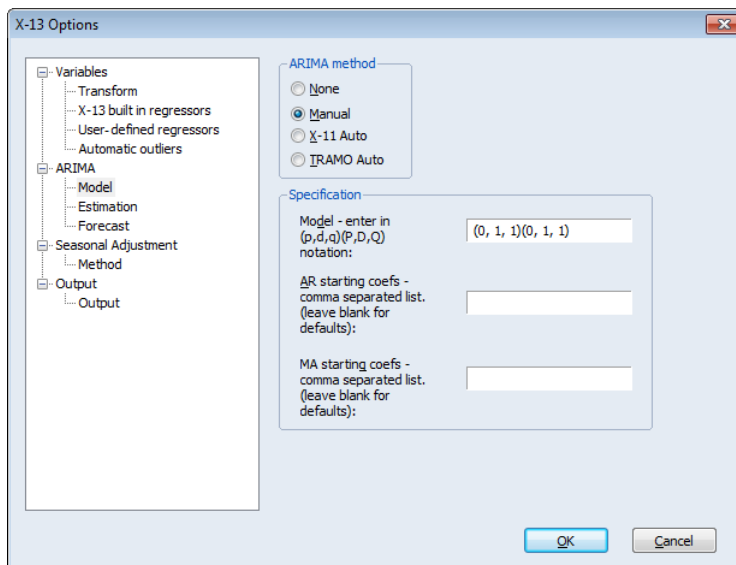
The first edit field, labeled **Model**, should be used to enter the text specification of the model, For example, to estimate a model with an AR(1), first differencing, an MA(1), a seasonal AR(1), no seasonal differencing and no seasonal MA terms, you would enter:

```
(1, 1, 1)(1, 0, 0)
```

The **AR starting coefs** edit field allows you to specify starting values for the AR coefficients in the ARIMA model. By default this field is left blank, meaning that X-13 default values will be used for starting values. If you wish to specify starting values, you must enter a starting value for every AR term (including seasonal AR terms), *separated by a comma.*

In addition, note that:

- Individual terms may be left blank between commas to indicate that the X-13 default should be used for that specific term. For example, for a (2, 1, 1)(1, 0, 0) model, you may enter "0.2, , 0.3" to indicate that a starting value of 0.2 should be used for the first AR term, the X-13 default starting value should be used for the second AR term, and 0.3 should be used for the seasonal AR term.

- You may specify that a specific term should be fixed at its starting value (not estimated) by including an "f" after its value. For example entering "0.2f, , 0.3" means that the first AR term is fixed at 0.2, the second AR term uses the default X-13 starting value, and the seasonal AR term has a starting value of 0.3.

The **MA starting coefs** edit field allows you to specify starting values for the MA terms in the ARIMA specification. The syntax for this edit field is the same as the one used for setting the AR starting coefs.

For more information on the specification of a specific ARIMA model and setting the starting values, see Section 7.1 (ARIMA) of the *X-13ARIMA-SEATS Reference Manual*.

X-11 Auto

If you choose **X-11 Auto** as the **ARIMA Method**, X-13 will automatically select the best ARIMA model using an algorithm from X-11-ARIMA. You will be prompted to specify a set of models from which to choose and a set of selection options.
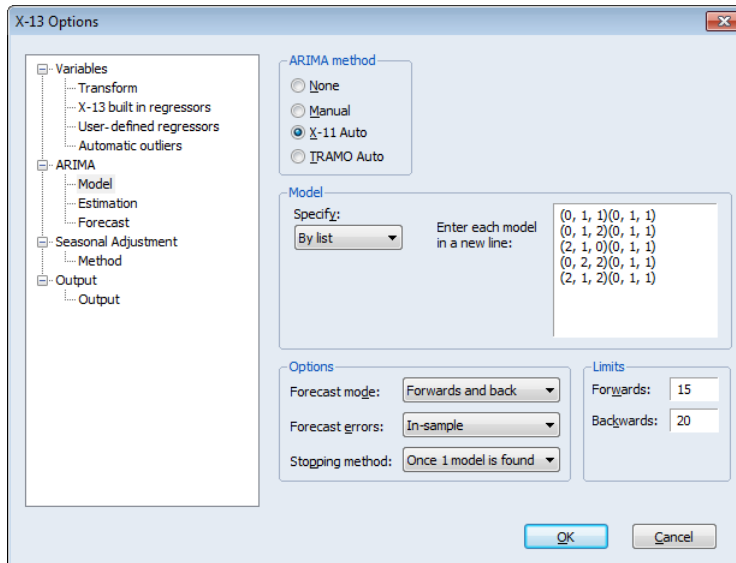
The X-11-ARIMA model selection algorithm requires that you specify a set of ARIMA models from which to choose. The **Specify** drop-down menu in the **Model** section offers a choice between three methods for specifying the candidate models: **By list**, **With limits**, or **By file**.

When you select one of these entries, the remaining portion of the dialog will change to show the corresponding options:

- If you select **By list**, you will be prompted to specify candidate models in a list box on the right side of the dialog.
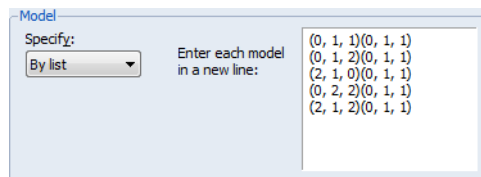
Each line should contain a candidate specification specified in "(p, d, q) (P, D, Q)" format. The list box will be pre-filled with a default set of specifications to test.

- If you select **With limits**, you will be prompted to set the maximum order of each term in the ARIMA model.

EViews will build up the set of candidate models by choosing every combination of orders up to the maximums you select. Note that X-11-ARIMA may not be able to identify the preferred model if you set the maximum number of terms to large values.

- If you select **By file**, you will be prompted to specify a file on disk containing a list of candidate models:

For details of the specification of the file, see the DETAILS part of Section 7.12 (Pickmdl) of the *X-13ARIMA-SEATS Reference Manual*.

The **Options** group box provides access to important **X-11 Auto** options.

- **Forecast Mode** specifies whether both forecasts and backcasts (**Forwards and back**) or only forecasts (**Forecast only**) will be used during the model selection process.

- The **Forecast Errors** option allows you to select whether **In-sample** or **Out-of-sample** forecast errors are used during the model selection.

- **Stopping Method** allows you select the model selection routine. If you choose **Once 1 model is found**, X-13 will stop searching through the list of available models as soon as it finds one that meets the selection criteria. If you select **Check all models**, X-13 will select the best model out of all models that meet the selection criteria.

The **Limits** section allows you to specify the limit values for forward, and (if relevant) backward selection.

The X11 selection procedure evaluates models using the forecast and backcast errors or forecast errors alone. A model is accepted if the absolute average percentage error of the forecast is less than the limit, and likewise for the backcast, if relevant.

Additional details on these options may be found in Section 7.12 ("Pickmdl") of the *X-13ARIMA-SEATS Reference Manual*.

TRAMO Auto

Selecting **TRAMO Auto** as the **ARIMA Method** instructs X-13 to choose the best ARIMA model using an algorithm from TRAMO/SEATS.
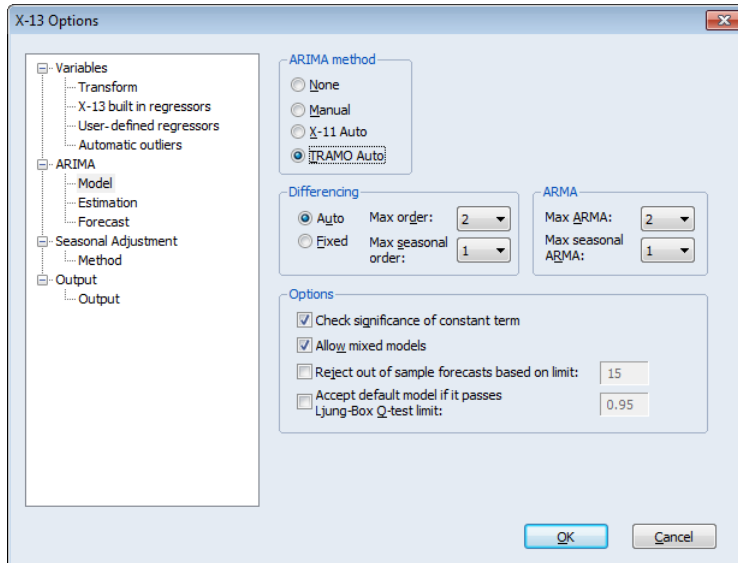
The **Differencing** and **ARMA** sections of the dialog let you specify the maximum order of differencing and ARMA terms. Note that the **Auto**/**Fixed** radio buttons allow you to set the level of differencing at a fixed value or to choose the level automatically.

The **Options** box contains settings for the TRAMO model selection routine:

- The **Check significance of constant term** check box specifies whether TRAMO will test for statistical significance of the constant term in the ARIMA model.

- **Allow mixed models** determines whether TRAMO will allow mixed ARIMA models (*i.e.* models with both AR and MA terms) or not.

- **Reject out of sample forecasts** tells TRAMO to test the out-of-sample forecast errors for the final three years of data, and to suppress forecasts if the forecast error surpasses the given limit.

- The **Accept default model if it passes Ljung-Box Q-test limit** option specifies whether TRAMO will automatically accept the default model "$(0, 1, 1)(0, 1, 1)$" and ignore all other models if the default passes a Ljung-Box Q-statistic test.

ARIMA Estimation

The **Estimation** page lets you specify basic estimation options for the ARIMA with regressors model.
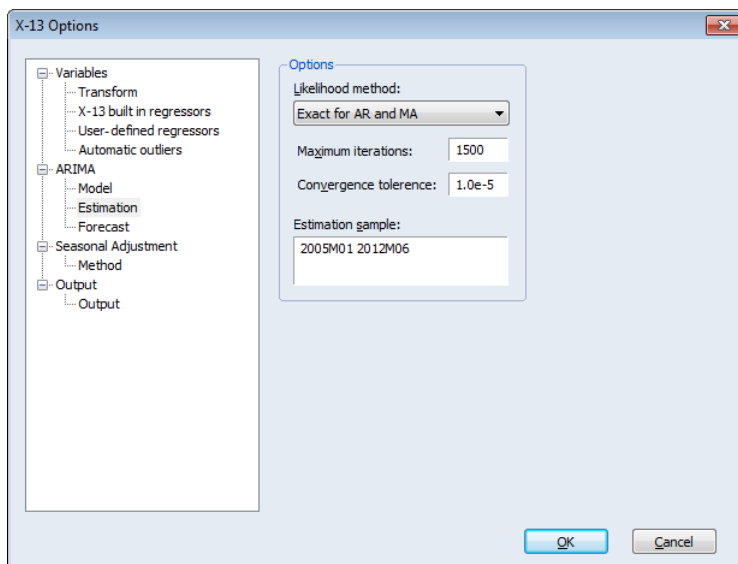
The **Likelihood method** drop-down allows you to specify whether an exact or a conditional likelihood is used for estimation and forecasting. The accompanying edit fields allow you to specify a maximum number of iterations and convergence tolerance.

You may use the **Estimation sample** edit field to specify a sample for the estimation of the model. By default the estimation sample is the same as the sample used for seasonal adjustment (*i.e.* the current workfile sample), but you may specify a contiguous sub-sample of the seasonal adjustment sample.

ARIMA Forecasting

The **Forecast** page offers options for forecasting or backcasting from the chosen model once it has been estimated.
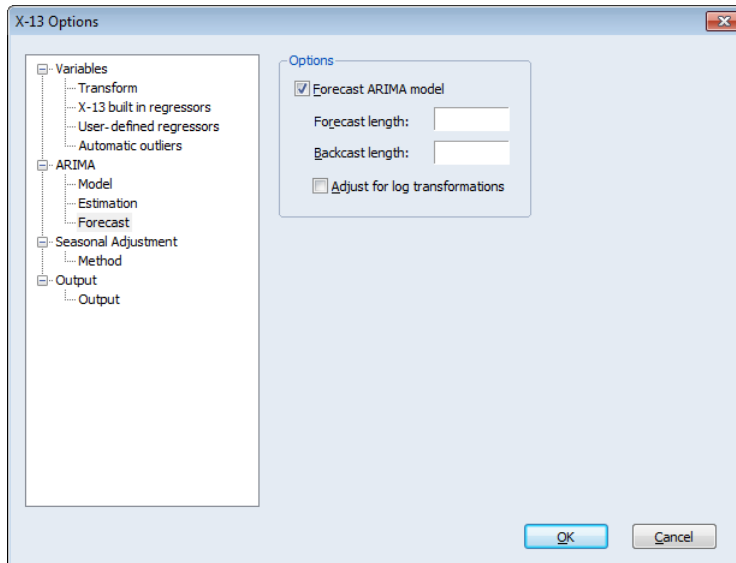
You may specify a forecast and backcast length (*i.e.,* the number of observations to be forecasted or backcasted), and you may instruct X-13 to make an adjustment to the forecasts to reflect the fact that they are generated from a log-normal distribution whenever a log-transformation is specified (see "Transform" on page 446).

Note that specifying a forecast or backcast length that produces data outside of the current workfile range will not produce an error, but any observations outside of the workfile range will not be brought into EViews.

X-13 Seasonal Adjustment Options

The **Seasonal Adjustment** branch of the dialog contains the options for the actual seasonal adjustment process undertaken by X-13.
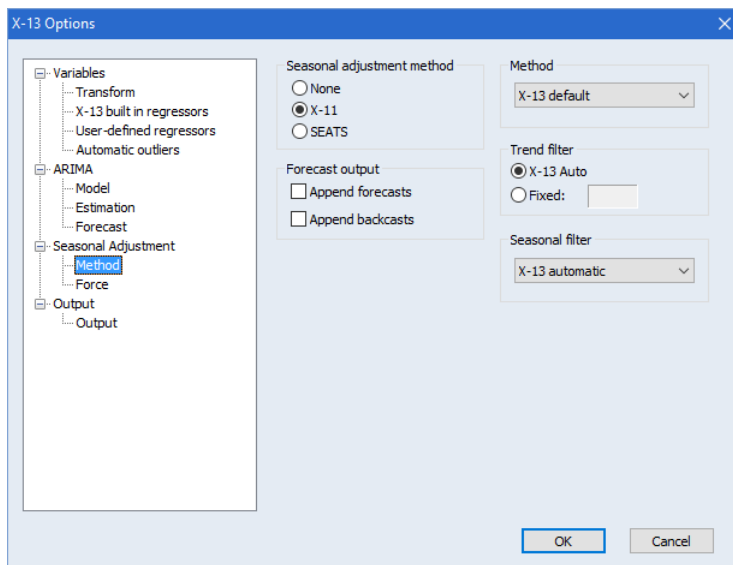
You may use the radio buttons in the **Seasonal adjustment method** section to select either **X-11** or **SEATS** based seasonal adjustment, or you may chose **None** to perform no seasonal adjustment so that X-13 simply estimates the specified ARIMA regression model.

If you specify a seasonal adjustment method, EViews will display a set of **Forecast output** options. The two checkboxes allow you to instruct X-13 to append one year of forecasted and/or backcasted values to certain output series (*e.g.*, the seasonal factors D10, if performing X-11 adjustment and the seasonal factors S10, if performing SEATS adjustment), and to display the results in the table output.

For a full list of series that can be forecasted/backcasted see the "appendfcst" and "append-bcst" entries in Section 7.19 of the *X-13ARIMA-SEATS Reference Manual* (if performing X-11 seasonal adjustment) and the "appendfcst" and "appendbcst" entries in Section 7.14 of the *X-13ARIMA-SEATS Reference Manual* (if performing SEATS).

The remaining portion of the dialog will show options for the X-11 and the SEATS seasonal adjustment methods.

- If you specify **X-11** as the seasonal adjustment method you will be presented with a few X-11 based options.

The **Method** drop-down menu lets you set the seasonal adjustment decomposition method. You may choose between **Multiplicative**, **Additive**, **Pseudo-additive**, **Log-additive**, or **X-13 default**, where the latter lets X-13 choose the mode depending upon the transformation (if any) given on the **Variables** branch.

The **Trend filter** box allows you to choose the Henderson trend moving average used for the final trend-cycle. You can let X-13 choose a value for you, or you may specify a fixed length (note only odd integers between 1 and 101 can be used).

The **Seasonal filter** drop-down specifies the type of seasonal moving average to employ. By default, X-13 will choose the type of seasonal filter automatically, but you may, if you prefer, choose a specific filter.

See Table 7.51 of the *X-13ARIMA-SEATS Reference Manual* for details on each filter type.

- Selecting **SEATS** as the seasonal adjustment method brings up SEATS based options.

    You may select whether to decompose the trend-cycle into a long term cycle using the Hodrick-Prescott filter by selecting the **Use HP filter** option.

    The **Do not allow stationary seasonal models** option tells X-13 to disallow adjustment of any seasonal model with no differencing. If this option is selected and a stationary seasonal ARIMA model is specified, X-13 will replace the seasonal ARIMA component with a (0, 1, 1) specification.

    The **Ljung-Box Q limit** sets the limit SEATS uses to test whether the provided ARIMA model is of acceptable quality.

X-13 Force Options

The Force section provides the ability to force annual totals of the adjusted series to match those of the original series. X-13 provides two methods for forcing the totals; the original Denton method of X-11, and a newer regression benchmarking method of Cholette and Dagum (1994) as adapted by Quenneville, Cholette, Huot, Chiu, and Fonzo (2004).
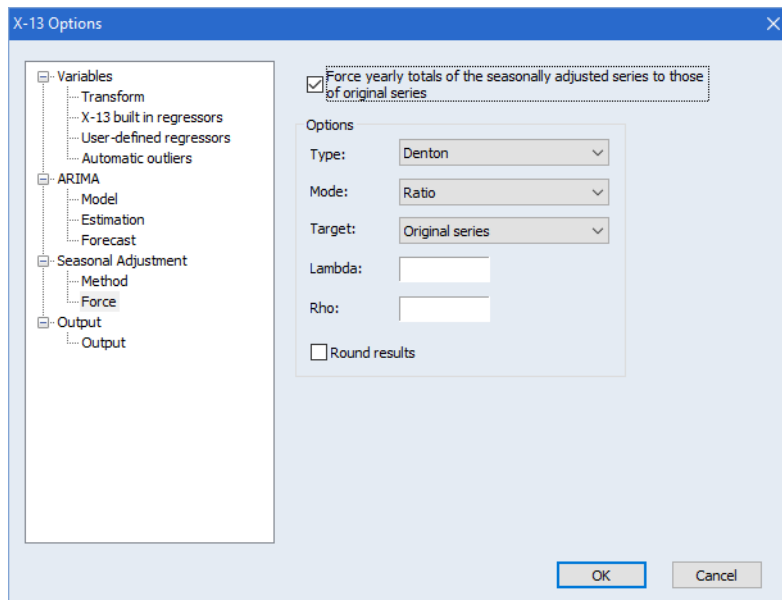
Selecting the **Force yearly totals…** check box instructs X-13 to force the annual totals, and enables the **Options** section of the dialog. Those options allow you to change the method of forcing from Denton to the regression method using the **Type:** combo-box, switch the whether the ratio or difference of the annual totals is used with the **Mode:** combo-box, and change the target series from the original data to an adjusted version of the original data with the **Target:** combo-box.

The **Lambda:** and **Rho:** fields can be used to fine tune the parameters of the forcing algorithms, and the **Round results** checkbox is used to instruct X-13 to use rounded data rather than the exact data.

More information on the X-13 force option can be found in Section 7.6 of the *X-13ARIMA-SEATS Reference Manual*.

X-13 Output Options

The **Output** section allows you to set which output series will be added to the workfile.
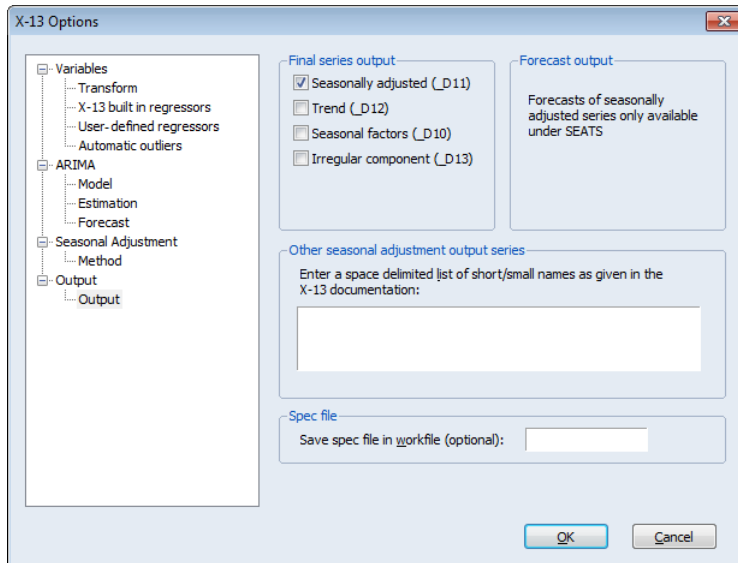
When performing either X-11 or SEATS seasonal adjustment, you may elect to output the seasonally adjusted data (called D11 in X-11 short name terminology, and S11 in SEATS), the trend component (D12, S12), seasonal factors (D10, S10) or the irregular component (D13, S13) by selecting the corresponding check boxes in the **Final series output** section.

If you chose SEATS as the seasonal adjustment method, the **Final series output** section of the dialog offers the option of saving the transitory component (S14), and the **Forecast output** section allows you to save the forecasted values of the seasonally adjusted series (AFD), trend (TFD), transitory component (YFD), seasonal factors (SFD).
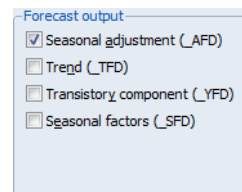
Each of these output series will be imported into EViews with a name given by the original series name plus an underscore followed by the X-13 short name. For example, if your original series is called GDP, the X-11 seasonal factors will be added to the workfile with the name GDP_D10.

If you chose to perform forecasting by setting the **ARIMA Forecast** options (see "ARIMA Forecasting" on page 455), you will also be given the choice to output the forecasted values and confidence bounds of the forecasted values. The point forecasts will be brought into the workfile with a name suffix of "_f", and the lower and upper bounds will have suffixes of "_fl" and "fu" respectively.
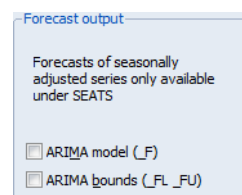
You may save other X-11 or SEATS output series to your workfile by typing their short names into the **Other seasonal adjustment output series** edit field. A full list of available output series, and their corresponding short names, can be found in Table 7.46 in Section 7.19 of the *X-13ARIMA-SEATS Reference Manual* (if performing X-11), and Tables 7.30 and 7.31 in Section 7.14 (if performing SEATS).

Finally, you may enter a name in the **Spec file** box to save the X-13 specification file as a text object in your workfile. Saving the specification file lets you reuse or modify the specification for further use (note that you can specify X-13 via the command line using a text object in the workfile).

Example

As an example of using the EViews X-13 proc, we will seasonally adjust monthly employment data obtained from the Federal Reserve Economic Data (FRED) database. The EViews workfile, "X13_Macro.wf1" contains monthly non-seasonally adjusted US unemployment data from January 2005 to June 2012 in a series called UNRATENSA. We will perform two types of seasonally adjustment: X-11 with automatic ARIMA selection performed using X-11-Auto, and SEATS based with automatic ARIMA selection performed using TRAMO.

To perform X-11 based seasonal adjustment with automatic ARIMA selection using X-11-Auto, open the UNRATENSA series and click on **Proc/Seasonal Adjustment/Census X-13**. EViews will display the X-13 dialog"

- Note that this example will not be using any exogenous variables, transformations or outliers in the model so skip the settings in the **Variables** branch of the tree structure

- Click on the **ARIMA** branch to display the ARIMA options. Select **X-11 Auto** as the **ARIMA Method**, and leave the remaining options at their default values.

- Open the **Seasonal Adjustment** branch and select **X-11** as the **Seasonal adjustment method**.

- Select the **Output** branch and check all four of the **Final series output** boxes so that EViews will import the seasonally adjusted values, the trend values (D12), the seasonal factors (D10) and the irregular components (D13) into the workfile.

After you click on **OK**, EViews will run the *X-13ARIMA-SEATS* program using the specified settings and will display the text output produced by the program in the series window. In addition, EViews will save the four output series UNRATENSA_D10, UNRATENSA_D11, UNRATENSA_D12 and UNRATENSA_D13 in the workfile.

UNRATENSA_D11 contains the final seasonally adjusted series. Select this series along with the original UNRATENSA, right-click and **Open/As group**, then select **View/Graph** and click on **OK** to display a line graph of the adjusted and unadjusted series:

The smoothed series UNRATENSA_D11 clearly exhibits less seasonally volatility than the original series UNRATENSA.

Displaying the graph views of the remaining series lets you view the individual components of the smoothing. For example, showing a graph of UNRATENSA_D12 enables you to see the trend component.

For comparison purposes, we perform SEATS based seasonal adjustment with automatic outlier detection on the same data. Select **Proc/Seasonal Adjustment/Census X-13** from the series menu to display the X-13 dialog:
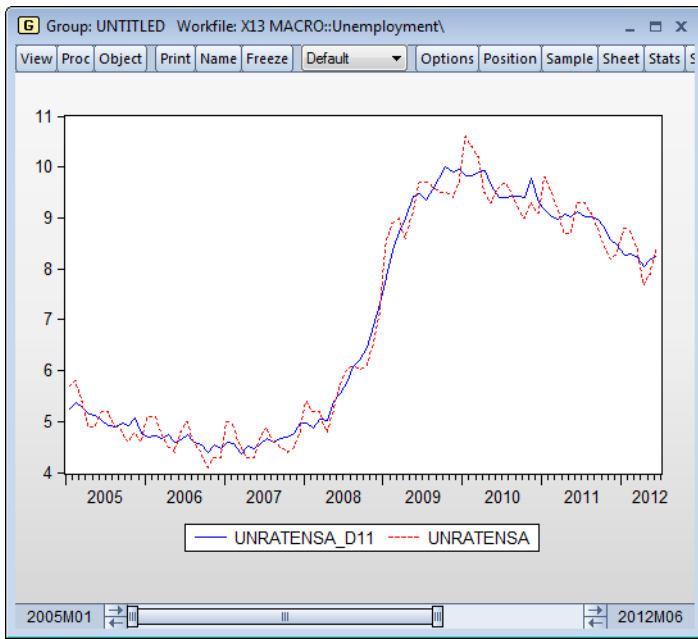
- Open the **Variables** branch of the dialog and click on the **Automatic outliers** node. Instruct EViews to perform automatic outlier selection by checking all four of the **Outlier types** boxes (**Additive**, **Level shift**, **Temporary change**, **Seasonal**). Leave the remaining settings at their defaults.

- Open the **ARIMA** branch of the tree, select the **Model** node, and select **TRAMO Auto** as the **ARIMA Method**. Leave all other options at their defaults.

- Open the **Seasonal Adjustment** branch, select **SEATS** as the **Seasonal adjustment method**, and select the **Append forecasts** option.

- Finally, switch to the **Output** branch, select the first four **Final series output** boxes to instruct EViews to import the seasonally adjusted values, the trend values, the seasonal factors, and the irregular components into the workfile. Note that by default the forecasted seasonal adjustment values (AFD) is selected in the **Forecast output** box.

- Click on **OK** to run X-13 with these settings and bring the five output series from X-13 to our workfile. Note that EViews will save the results in the new series

UNRATENSA_S10, UNRATENSA_S11, UNRATENSA_S12 and UNRATENSA_S13, and UNRATENSA_AFD.

We may open up a group object containing the seasonally adjusted series, UNRATENSA_S11, along the X-11 seasonally adjusted series UNRATENSA_D11 by selecting the three series, right-clicking and selecting **Open/As** group. Extend the workfile sample by two years (24 months) by issuing the command

```
smpl @first 2012m06+24
```

select **View/Graph** and click on **OK** to display a line graph of the two adjusted and one unadjusted series:

Notice that the SEATS seasonally adjusted series, UNEMPNSA_S11, closely tracks the X-11 adjusted series, UNEMPNSA_D11, with both offering a seasonally smoother version of the original series.

Also note that since we used the **Append forecasts** to instruct SEATS to append two years of forecasts to the final seasonally adjusted value, the UNEMPSNA_S11 series continues until June 2014.
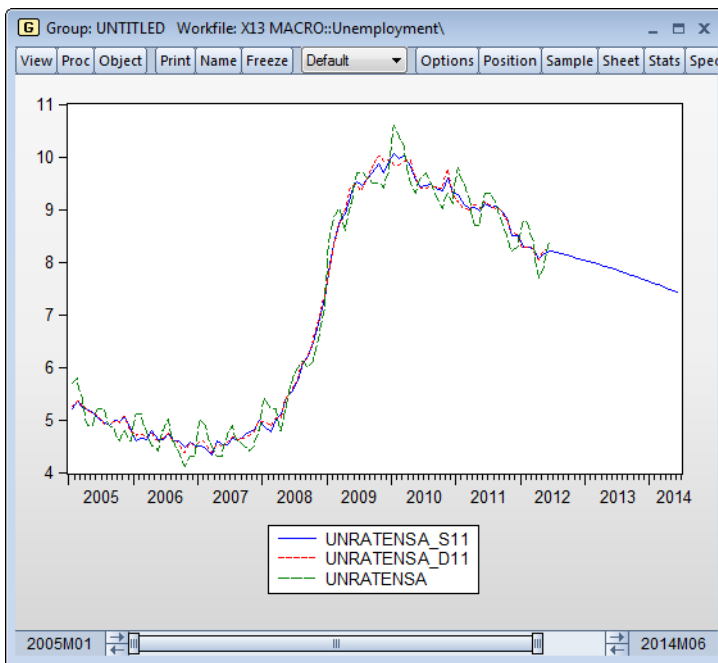
STL Decomposition

STL decomposition is a seasonal adjustment method that decomposes a series into seasonal, trend and remainder components using a filtering algorithm based upon LOESS regressions. Introduced by Cleveland, Cleveland, McRae and Terpenning (*Cleveland* henceforth) in their 1990 paper, STL has two main advantages over other seasonal adjustment methods; it works on any frequency of data, and can be calculated on time series data with irregular patterns and missing values.

STL assumes an additive relationship between the seasonal, trend and remainder components:

$$Y_t = S_t + T_t + R_t \tag{11.39}$$

The calculation of each component involves a double recursive procedure. Specifically, the algorithm is composed of inner and outer loops with the following specification:

The Inner Loop

Let k denote the iteration parameter, running from 1 to N. np denotes the number of periods per cycle (e.g. 12 for monthly data), N the number of years. The total number of observations is then $N*np$ (assuming we start and end at same point in cycle). Let RW denote a passed in weighting series.

1. Create a series of detrended values, $DT_t^k = Y_t^k - T_t^{k-1}$, using previous Trend estimate, T_t^{k-1}.

2. Split the detrended series into np sub-series each of length NI. E.g. create a series containing just January values, another containing just February values. For each of these sub-series, perform LOESS to create smoothed values. In computation of the LOESS weights, the weights are further weighted by RW.

   Smoothed values are also created for any missing values, and for 1 cycle (year) prior to and after the data sample. For example, if we have monthly data from 2000 to 2010, then we first create smoothed values for Jan 1999, Jan 2000, Jan 2001, …, Jan 2010, Jan 2011. Then smoothed values for Feb 1999, Feb 2000, …, Feb 2010, Feb 2011, and so on. This results in having 2 extra smoothed observations per sub-series. Or $(2+N)*np$ in total. The results are stored in series C_t^k.

3. For each smoothed sub-series, perform 3 moving average calculations, recursively. These moving average calculations result in the loss of 2 observations per sub-series, but since we created an extra 2 observations in Step 2, we end up with the correct number of observations. The thrice MA'd series are then smoothed with another LOESS smoothing and stored in series L_t^k.

4. Calculate an estimate of the seasonal values, $S_t^k = C_t^k - L_t^k$.

5. Deseasonalize, $DS_t^k = Y_t^k - S_t^k$.

6. Smooth DS_t^k with LOESS smoothing to compute T_t^k. In computation of the LOESS weights, the weights are further weighted by RW.

The initial trend estimate, T_t^0, is set equal to zero.

The Outer Loop

After the iterations of the inner loop are completed, we are left with an estimate of the trend, T_t^{NI}, and of the seasonal component, S_t^{NI}. We can estimate the remainder component as:

$$R_t^{NI} = Y_t - S_t^{NI} - T_t^{NI} \tag{11.40}$$

To correct for outliers, the remainder series is used to calculate robustness weights. Let $h = 6 \cdot \text{median}(|R_t^{NI}|)$, then $RW_t = B(|R_t^{NI}|/h)$, where B is the bisquare function:

$$B(x) = \begin{array}{ll} (1-x^2)^2 & \text{for } 0 \le x < 1 \\ 0 & x \ge 1 \end{array} \tag{11.41}$$

RW is then fed in to the next set of iterations of the inner loop. The initial set of *RW* are set at 1.

LOESS Estimation

The heart of the STL procedure is the heavy use of LOESS estimation to create smoothed values of data. STL uses LOESS with a tri-cube function with either a locally linear or locally quadratic polynomial. To smooth the time series A, we first create a trend variable T, running from 1 to T (number of observations in A). We smooth each value in A one at a time. Each value of A, A_t, has a corresponding trend value, τ. To calculate the smoothed value of A_t we take the k observations of A and T around (A_t, τ) to form new smaller time series, A^k and T^k.

Weights are then calculated as:

$$\omega_t = W\left(\frac{|T_t^k - \tau|}{h_\tau}\right)$$

Where W is the tri-cube function:

$$W(x) = \begin{array}{ll} (1-x^3)^3 & \text{for } 0 \le x < 1 \\ 0 & x \ge 1 \end{array}$$

And h_τ is the furthest distance of τ from the values in T^k:

$$h_\tau = \max(\tau - \min(T^k), \max(T^k) - \tau)$$

The weights may then be used to either calculate a weighted average of the A^k as an estimate of A_t (if using a locally-linear polynomial) or to perform weighted least squares of A^k against T^k and use the coefficients to produce a fitted value of A_t (if using a locally quadratic polynomial).

When estimating using the LOESS, the size of the window, k must be decided. Cleveland recommend a seasonal smoothing (Step 2) window of at least 7. Larger values increase the smoothness of the seasonal component.

Setting the seasonal smoothing window larger than the sample size will guarantee that the seasonal components are constant across cycles (e.g. the January value is the same each year, the February value is the same each year, and so on). Shorter values can lead to large variations in the seasonal components across years.

Selecting the correct smoothing windows size can be difficult. Cleveland suggest a graphical diagnostic tool, termed the seasonal-diagnostic plot, which graphs the relationship between the demeaned seasonal component and the demeaned seasonal component plus the remainder component for each period. Smooth seasonal lines imply a reasonable choice of seasonal window.

For the trend smoothing (Step 6), Cleveland recommend a value between $1.5np$ and $2np$. For the filter window (Step 3), the lowest integer greater than np is recommended.

Missing Values

The LOESS smoothing technique used at to produce the seasonal and trend components of the procedure is able to produce values even when the underlying data is missing. The procedure is only unable to calculate a smoothed value if there are not enough observations in the LOESS window with valid data. A locally linear polynomial requires at least one valid observation inside the window, and the locally quadratic polynomial requires at least two.

STL Decomposition in EViews

To perform STL Decomposition of a series in EViews click on **Proc- > Seasonal Adjustment- > STL Decomposition…** to bring up the **STL Decomposition** dialog:

The **Cycle specification** box allows you to specify the periodicity of your data - number of periods per cycle. This will usually be set to the number of periods per year for the frequency of your data, for example monthly data has a periodicity of 12. This number must be set to a positive integer.

Weekly and daily data are problematic since there is not an integer number of weeks or days in a year. EViews can still compute STL Decomposition in these cases by expanding each year of data so that it has a full set of periods. For example, for weekly data each year is expanded so that it has exactly 53 weeks. Any years that actually have 52 weeks are expanded to have an extra observation (filled with an NA). Similarly, daily data is expanded to have 366 observations per year. To instruct EViews to perform this expansion on weekly data, enter **w** rather than an integer number as the periodicity. For daily data, enter **d**.

If your workfile is an irregular dated workfile (i.e. some observations are missing from the workfile), and has a frequency of either semi-annual, quarterly, monthly, weekly or daily, and you set the periodicity to be one year (e.g. 12 for monthly data, 2 for semi-annual data, **d** for daily data), EViews will adjust the STL procedure so that the dates in each year continue to line up, even with the missing observations.

The **Polynomial degree** area specifies the polynomial degree of the LOESS estimations. The seasonal polynomial degree corresponds to the that used during Step 2 of the inner loop of the STL procedure. The trend polynomial degree is used during Step 6, and the filter polynomial is used during Step 3.

Similarly, the **Smoothing Window** specifies the size of the window, k, used during the LOESS estimations of Step 2 (seasonal), Step 6 (trend) and Step 3 (filter). These smoothing window lengths must be odd integers.

The **Output series** area can be used to specify the names of the seasonal and trend series that will be created as part of the STL procedure. You may also optionally provide a forecast length. If specified, EViews will take the average of the last three cycle's seasonal values as a forecast of the seasonal component, and will linearly extrapolate from the last three cycle's trend values as a forecast of the trend component. The **Display seasonal diagnostics graph** checkbox instructs EViews to display the seasonal diagnostics plot recommended by Cleveland to determine the correct seasonal window length.

Iteration control specifies the number of times the inner loop and outer loop are computed. If you believe your data has a small number of outliers, the outer loop iterations can be set to a small number such as 1 or 2. Cleveland's original paper suggests that the inner loop converges quickly, so a low number can also be used for the number of inner loop iterations.

The **Estimation sample** box allows you to specify the sample used to perform the decomposition. Note that unlike the other seasonal adjustment methods in EViews, STL Decomposition does not require a continuous sample with no missing values.

Clicking **OK** on the STL dialog will produce the results of the decomposition. EViews will display a graph of the original data, along with the seasonal, trend and remainder components, and if requested, the seasonal diagnostics plot.

Examples

As an example of using STL decomposition, we will replicate one of the studies performed in the original Cleveland paper using monthly averages of measurements of atmospheric CO_2 by the Scripps Institute of Oceanography between 1959 and 1987 (the data are in the CO2.wf1 workfile). A quick graph of the original data shows it exhibits strong seasonality with a clear upward trend:

We will use the STL procedure to decompose this time series into seasonal and trend components. We click on **Proc/Seasonal Adjustment/STL Decomposition** to open the **STL Decomposition** dialog:

Following Cleveland's example, we set the seasonal polynomial degree to **0**, and the other two to **1**. We use a seasonal smoothing window of **35**, a trend window of **19** and a filter of **13**. Cleveland use short iteration loops, setting the inner loop at just **2** iterations, and the outer loop at **0**. We choose to produce the seasonal diagnostic plot. Clicking **OK** produces the two sets of output graphs:
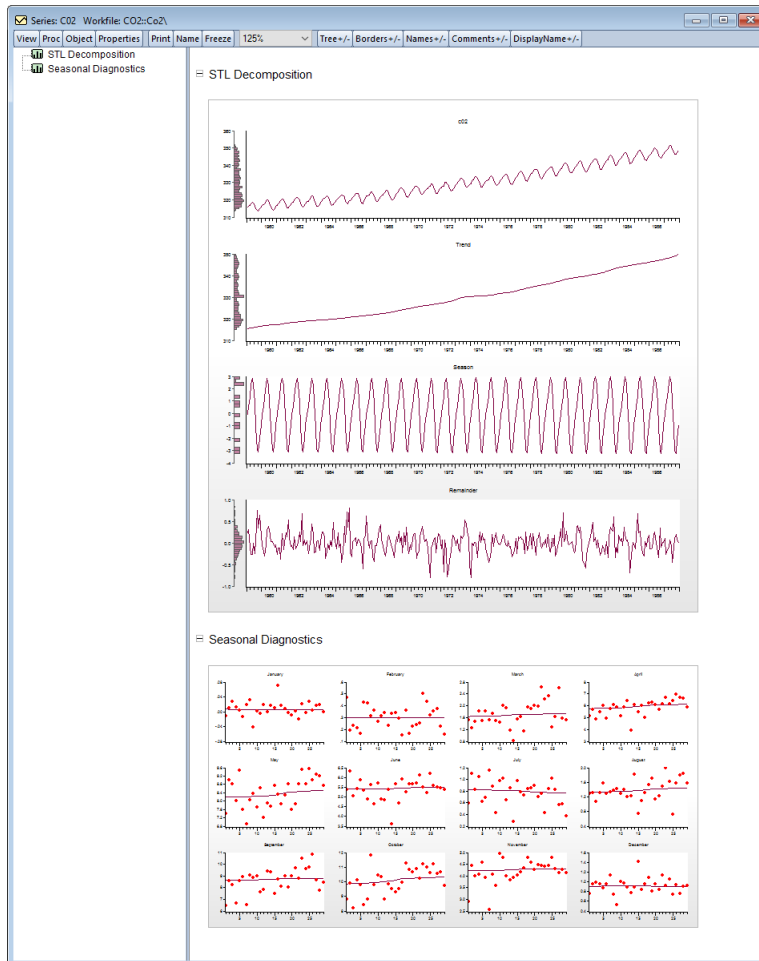
The first set of graphs shows the original data, labeled CO2, the decomposed trend component, the seasonal component and the remainder. The trend is a near straight upward trending line, and the seasonal component, although allowed to change through time, is consistent with high values approaching 3 every May and low values passing -3 every October. The relatively flat lines in each of the Seasonal Diagnostics panels indicate that the choice of 35 as the seasonal window was appropriate.

As a second example we will decompose a series of weekly electricity demand in England and Wales between April 2005 and April 2013. The data is missing values in April 2005 and June 2005. The data are contained in the workfile elecdmd_weekly.wf1, which has a range of April 2005 until April 2015 (i.e. two extra years of observations are included in the workfile). A graph of the data displays clear seasonal patterns:

ELECDMD
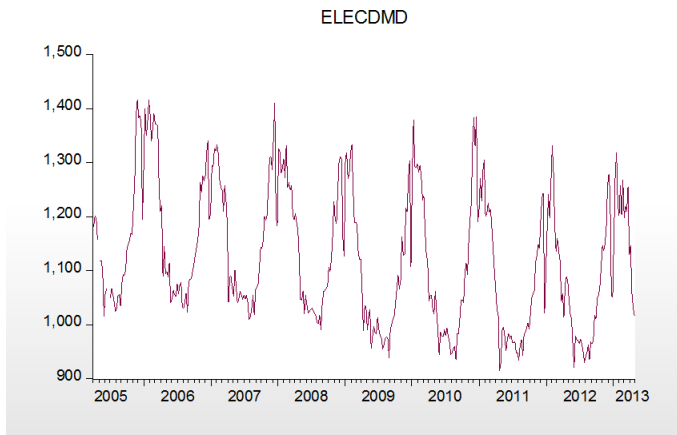
Note the gaps in the line corresponding to the missing values in early 2005.

We will use the STL procedure to decompose this time series into seasonal and trend components. We click on **Proc/Seasonal Adjustment/STL Decomposition** to open the **STL Decomposition** dialog:

We will leave all settings at their default value, other than changing the length of the seasonal smoother to **11**. Note that since we have weekly data, the Periodicity box is pre-filled in with the value "w". Our data runs from 2005 until 2013. Over that time period each year has 52 weeks apart from 2010, which has 53 weeks.

Also note that we have selected to forecast the seasonal and trend components beyond the end of the data, by entering "104" as the number of forecast periods (two years).

Clicking **OK** produces a graph of the estimated seasonal, trend and remainder components:
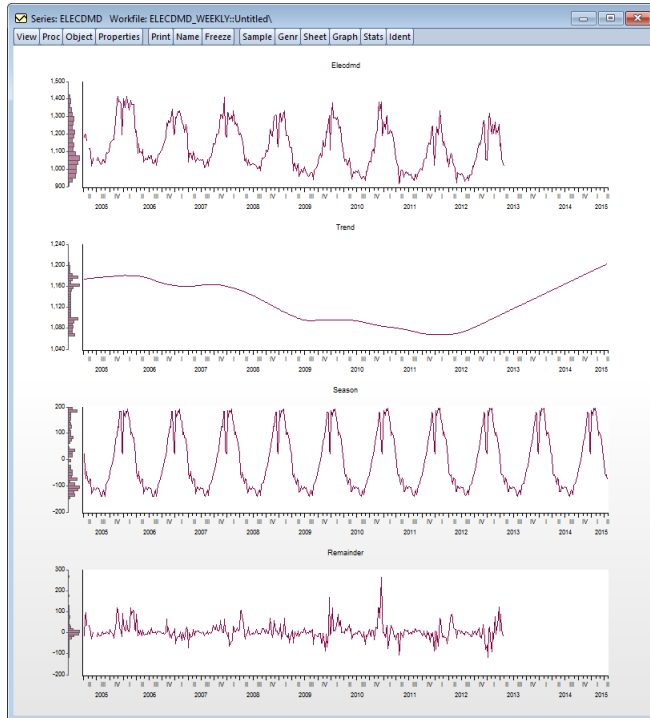
The trend components is less linear than our previous example, with a clear downward path between 2006 and 2011, followed by an upward path.

MoveReg Weekly Seasonal Adjustment

EViews offers a front-end interface to the U.S. Bureau of Labor's MoveReg weekly seasonal adjustment program. Most seasonal adjustment routines, including the U.S. Census Bureau's X-11, X-12 and X-13 packages, require the data be sampled at a monthly or quarterly frequency. However, many economic series are sampled on a weekly basis, meaning that these seasonal adjustment techniques cannot be used. The MoveReg program rectifies this gap by providing a seasonal adjustment method aimed directly at weekly data.

Similar to many of the other seasonal adjustment methods in EViews, the MoveReg routine performs the following steps:

- Write out a specification and data file for the series and the provided options.

- Execute the MoveReg program in the background using the created specification and data files.

- Read the output back into the EViews workfile.

To perform MoveReg seasonal adjustment, open the series and select **Proc/Seasonal Adjustment/MoveReg…** EViews will then open a dialog allowing you to specify the options for the MoveReg procedure:
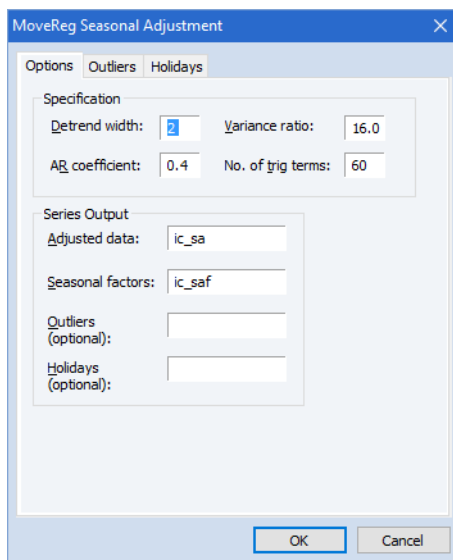
The dialog has three tabs. The first tab, **Options**, provides the basic MoveReg specification options, as well some output options.

To change the width of the detrending filter used by the MoveReg procedure, use the **Detrend width:** edit field. The **AR coefficient:** and **Variance ratio:** fields can be used to change the AR parameter and variance ratio term of the ARMA specification respectively. To modify the number of trigonometric terms used in the regression, change the **No. of trig terms:** field. Note that this is the total number of terms, including both sine and cosine terms. Since the number of sine and cosine must be equal, this number must be an even positive integer.

The **Series Output** area specifies the names of the series that will be imported into the EViews workfile. To change the name of the seasonally adjusted data and the seasonal factors, use the **Adjusted data:** and **Seasonal factors:** edit fields respectively. If you have specified outliers on the **Outlier** tab, you can choose to import the outlier factors by entering a name into the **Outliers (optional):** field. Similarly, if you have specified holidays in the **Holidays** tab, you can import them by providing a name in the **Holidays (optional:)** field.

The **Outliers** tab allows specification of both additive and level shift outliers:
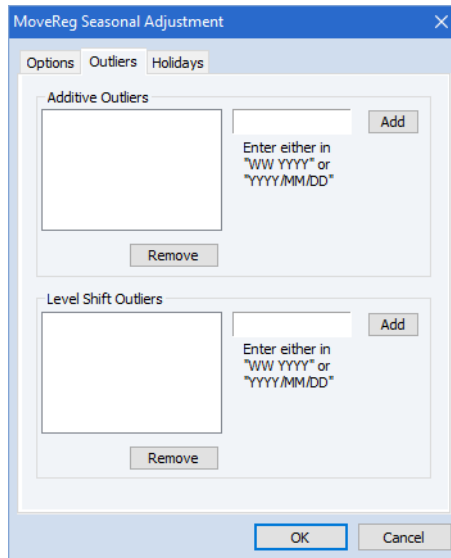
To add an additive outlier, enter the outlier's date into the **Additive Outliers** edit field. Note you can enter the date either as a week number followed by the year, or as an explicit date in "YYYY/MM/DD" format. EViews will convert a date into week number year format. Once you have entered the date, click the **Add** button to add the outlier to the outlier list. If you would like to remove an added outlier, select it and then click the **Remove** button.

Level shift outliers can be specified in an identical manner to the additive ouliers.

Holidays can be specified on the **Holidays** tab:
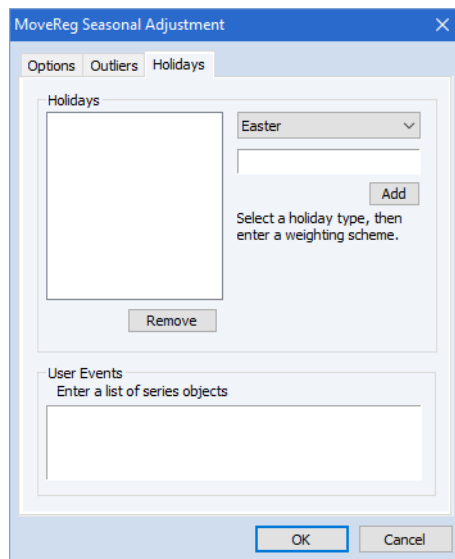
MoveReg allows for two types of holidays – built in special holidays or user-defined events.

To select a built in holiday, use the **Holidays** combo box to choose one of the pre-defined holidays, then enter a weighting scheme in the edit field below the combo. The weighting scheme should be specified as defined by the MoveReg application – the first digit should be the number of weights (the length of the holiday effect, in weeks), the second should be position of the holiday in the weights, and the final digits should be the weights themselves. The digits are separated by a space.

For example, entering "1 1 1" means the holiday lasts one observation (week), the position of the holiday is the first weight, and that weight has a value of 1. "3 2 1 2 1" means the holiday last three weeks, where the second weight is the holiday itself. The week before the holiday date has a weight of one, the week of the holiday a weight of two, and the week following the holiday date has a weight of one.

Once you have selected the holiday and entered its weighting scheme, you can click the **Add** button to add it to the **Holidays** list. If you would like to remove a previously added holiday, you can select it and click the **Remove** button.

The **User Events** edit field can be used to specify your own holidays/events. Simply enter the name of one or more series in the workfile containing binary information on whether an observation is a holiday/event or not.

Troubleshooting

The MoveReg procedure can be very sensitive to the specification of holidays. In particular:

- If a user event is provided, data for that event must extend two years beyond the last date in the adjustment sample. If data is not provided, MoveReg will fail. If a user event series is provided, and data is not available for two years (either because of NAs in the series, or because the workfile range does not extend for two years beyond the current sample), EViews will fill the event series with zeros for the missing observations.

- MoveReg will only forecast seasonal adjustment factors for two years beyond the adjustment sample if holidays or user events are included in the specification.

- MoveReg will often fail if sparse (mainly zeros with few ones) user events are included, or if none are included.

If holidays are required, we strongly recommend always including at least one user-event with regular ones.

An Example

As an example we will perform MoveReg seasonal adjustment on U.S. weekly unemployment initial claims (non-seasonally adjusted) data between the 30th December 1989 and 22nd October 2016.

We first obtain the data by creating a weekly workfile with those dates and fetching the data from the FRED database, with the following EViews commands:

```
wfcreate w 30/12/1989 22/10/2016
fetch(db=fred) icnsa
```

We open the ICNSA series by double clicking on it, and then view a graph of the series by clicking on **View/Graph**, and then selecting **OK**:
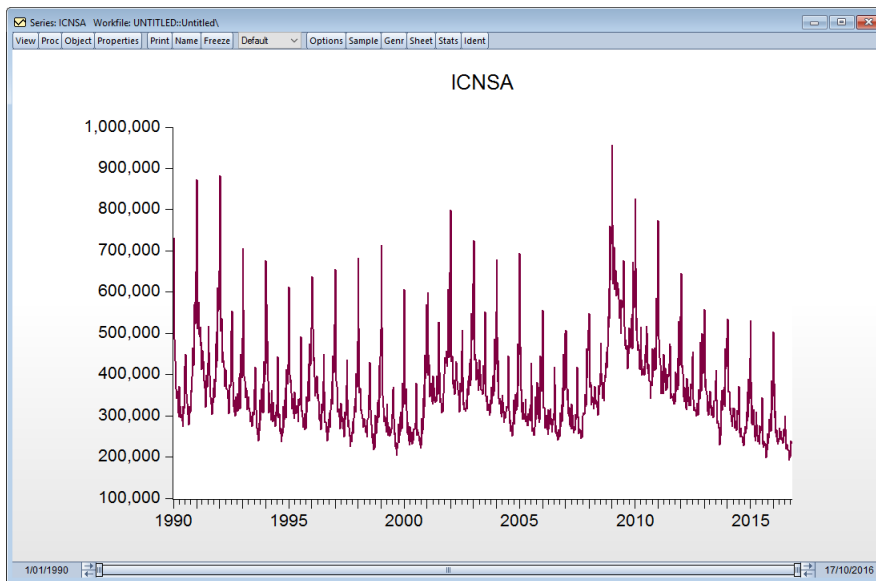
These data exhibit strong seasonal patterns. To remove this seasonality we perform Mov-eReg seasonal adjustment by clicking on **Proc/Seasonal Adjustment/MoveReg…** To begin we perform the adjustment without allowing for holiday or outlier effects, and so keep all dialog options at their defaults:
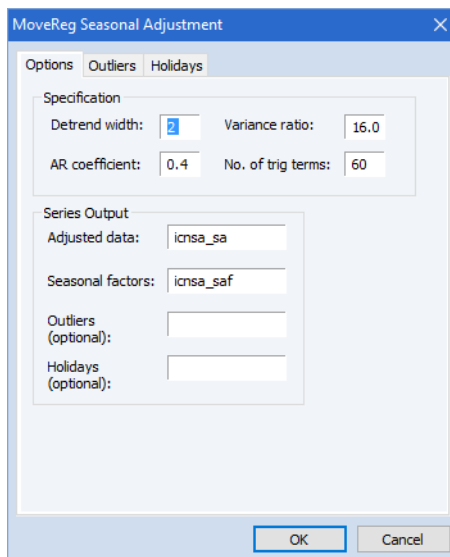
Clicking **OK** performs the adjustment, producing the following output:
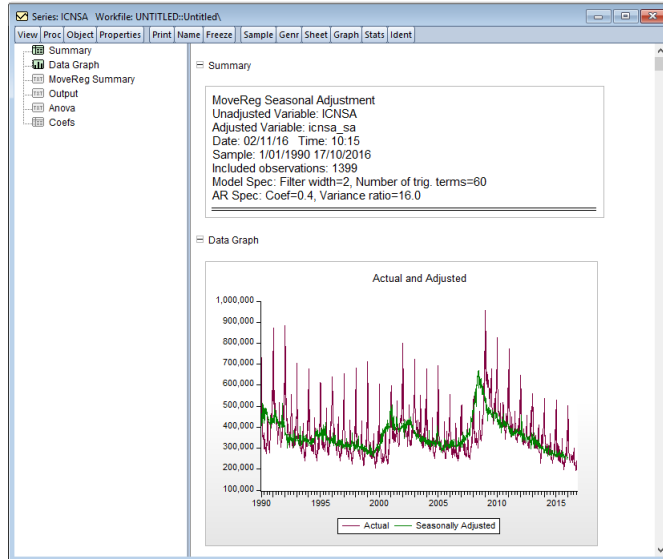
The first part of the spooled output provides a brief description of what we did, including the name of the underlying series, the name of the created seasonally adjusted series, the number of observations that were used, and the options that we chose.

The second part provides a graph showing the original, unadjusted series (in red in our case), alongside the produced seasonally adjust series (in green). It is clear that for our data the MoveReg procedure did a good job of removing the seasonal effects.

The remainder of the spooled output contains information and output provided by the MoveReg procedure itself. This information can be useful in performing diagnostics on the procedure.

We continue our analysis by adding in some outlier and holiday options to the procedure. Clicking **Proc/Seasonal Adjustment/MoveReg…** brings up the dialog again. Since we will be adding outliers and holidays, and wish to see their effects, we enter series names into their output boxes:

Switching to the **Outliers** tab we believe there may be an additive outlier in the second week of September in 2008 (the Lehman Brother's bankruptcy), and one for 9/11 2001:

(here we have not yet clicked the **Add** button to enter the 9/11 outlier).

On the **Holidays** tab we enter a Christmas holiday effect. We believe the holiday effect of Christmas will last 5 weeks, 3 weeks before Christmas, the week of the 25th, and the week

after. The week of and the week after will have slightly greater weights than the weeks before:
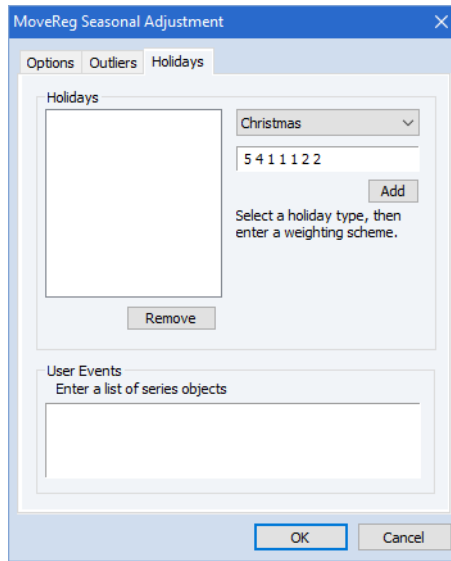
(again we have not yet clicked **Add** in this screenshot).

Clicking **OK** produces similar output:
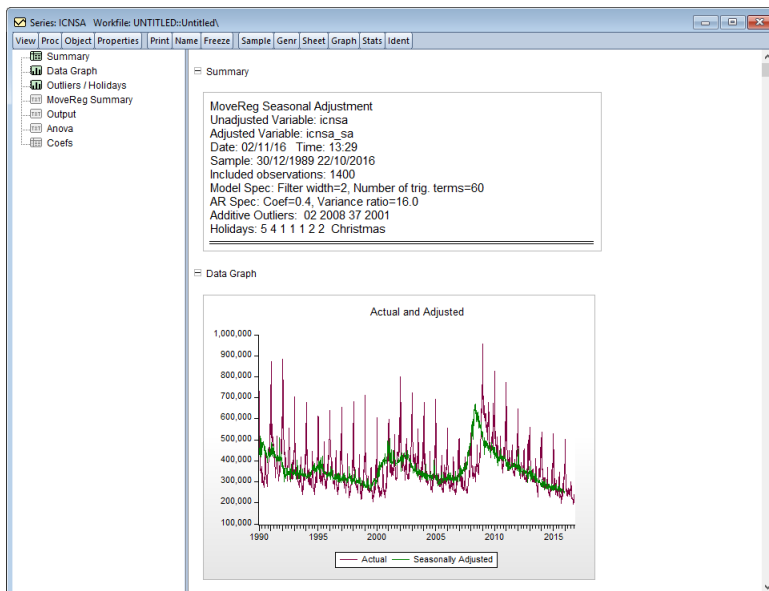
The summary information at the top again describes the options we chose, including the outlier and holiday specification. EViews has translated the outlier dates into week year format in the output.

The graph of the adjusted values looks very similar to the original (indeed comparison of the two adjusted series indicates that the addition of outliers and holidays led to only a 0.1% difference).

EViews now offers a new graph displaying the outlier and holiday factors as part of the spooled output:
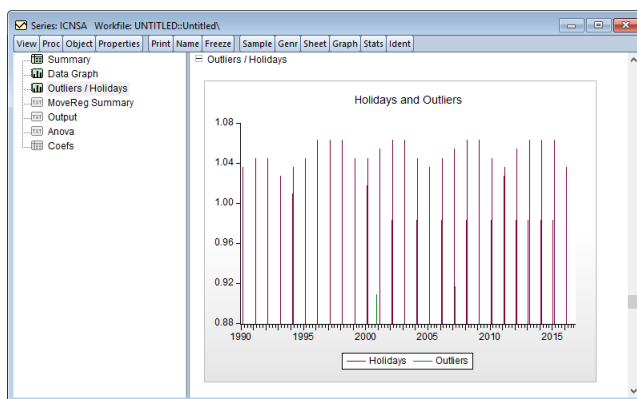
Census X12

EViews provides a convenient front-end for accessing the U.S. Census Bureau's X12 seasonal adjustment program from within EViews. The X12 seasonal adjustment program X12A.EXE is publicly provided by the Census and is installed in your EViews directory.

When you request X12 seasonal adjustment from EViews, EViews will perform all of the following steps:

- write out a specification file and data file for the series.
- execute the X12 program in the background, using the contents of the specification file.
- read back the output file and saved data into your EViews workfile.

The following is a brief description of the EViews menu interface to X12. While some parts of X12 are not available via the menus, EViews also provides a more general command interface to the program (see x12 (p. 625) in the *Command and Programming Reference*).

Users who desire a more detailed discussion of the X12 procedures and capabilities should consult the Census Bureau documentation. The full documentation for the Census program,

X12-ARIMA Reference Manual, can be found in the "docs" subdirectory of your EViews installation directory in the PDF files "Finalpt1.PDF" and "Finalpt2.PDF".

To call the X12 seasonal adjustment procedure, select **Proc/Seasonal Adjustment/Census X12...** from the series window menu. A dialog will open with several tabs for setting the X12 options for seasonal adjustment, ARIMA estimation, trading day/holiday adjustment, outlier handling, and diagnostic output.

It is worth noting that when you open the X12 dialog, the options will be set to those from the previously executed X12 dialog. One exception to this rule is the outlier list in the **Outliers** tab, which will be cleared unless the previous seasonal adjustment was performed on the same series.

Seasonal Adjustment Options

X11 Method specifies the form of the seasonal adjustment decomposition. A description of the four choices can be found in pages 75-77 of the *X12-ARIMA Reference Manual*. Be aware that the Pseudo-additive method must be accompanied by an ARIMA specification (see "ARIMA Options" on page 484 for details on specifying the form of your ARIMA).

Note that the multiplicative, pseudo-additive, and log-additive methods do not allow for zero or negative data.

The **Seasonal Filter** drop-down box allows you to select a seasonal moving average filter to be used when estimating the seasonal factors. The default **Auto (X12 default)** setting is an automatic procedure based on the moving seasonality ratio. For details on the remaining seasonal filters, consult the *X12-ARIMA Reference Manual*. To approximate the results from the previous X11 program's default filter, choose the X11-default option. You should note the following:

- The seasonal filter specified in the dialog is used for all frequencies. If you wish to apply different filters to different frequencies, you will have to use the more general X12 command language described in detail in x12 (p. 625) in the *Command and Programming Reference*.

- X12 will not allow you to specify a 3×15 seasonal filter for series shorter than 20 years.
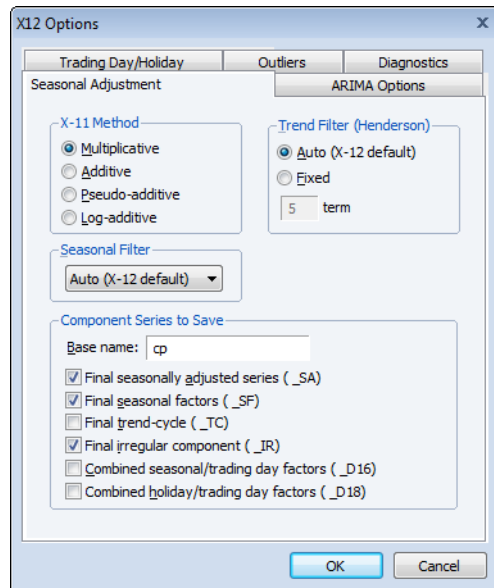
- The Census Bureau has confirmed that the X11-default filter option does not produce results which match those obtained from the previous version of X11. The difference arises due to changes in extreme value identification, replacement for the latest values, and the way the end weights of the Henderson filter is calculated. For comparability, we have retained the previous (historical) X11 routines as a separate procedure (see "Census X11 (Historical)" on page 490). Please note that the old X11 program is year 2000 compliant only through 2100 and supports only DOS 8.3 format filenames.

The **Trend Filter (Henderson)** settings allow you to specify the number of terms in the Henderson moving average used when estimating the trend-cycle component. You may use any odd number greater than 1 and less than or equal to 101. The default is the automatic procedure used by X12.

You must provide a base name for the series stored from the X12 procedure in the **Name for Adjusted Series/Component Series to Save** edit box. To save a series returned from X12 in the workfile, click on the appropriate check box. The saved series will have the indicated suffix appended to the base name. For example, if you enter a base name of "X" and ask to save the seasonal factors ("_SF"), EViews will save the seasonal factors as X_SF.

You should take care when using long base names, since EViews must be able to create a valid series using the base name and any appended Census designations. In interactive mode, EViews will warn you that the resulting name exceeds the maximum series name length; in batch mode, EViews will create a name using a truncated base name and appended Census designations.

The dialog only allows you to store the four most commonly used series. You may, however, store any additional series as listed on Table 6-8 (p. 74) of the *X12-ARIMA Reference Manual* by running X12 from the command line (see x12 (p. 625) in the *Command and Programming Reference*).

ARIMA Options

The X12 program also allows you to fit ARMA models to the series *prior* to seasonal adjustment. You can use X12 to remove deterministic effects (such as holiday and trading day effects) prior to seasonal adjustment and to obtain forecasts/backcasts that can be used for seasonal adjustment at the boundary of the sample. To fit an ARMA, select the **ARIMA Options** tab in the **X12 Options** dialog and fill in the desired options.

The **Data Transformation** setting allows you to transform the series before fitting an ARMA model. The **Auto** option selects between no transformation and a log transformation based on the Akaike information criterion. The **Logistic** option transforms the series y to

$\log(y/(1-y))$ and is defined only for series with values that are strictly between 0 and 1. For the **Box-Cox** option, you must provide the parameter value λ for the transformation

$$\begin{cases} \log(y_t) & \text{if } \lambda = 0 \\ \lambda^2 + (y_t^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \end{cases} \tag{11.42}$$

See the "transform spec" (p. 60–67) of the *X12-ARIMA Reference Manual* for further details.

ARIMA Specification allows you to choose between two different methods for specifying your ARIMA model. The **Specify in-line** option asks you to provide a single ARIMA specification to fit. The X12 syntax for the ARIMA specification is different from the one used by EViews and follows the Box-Jenkins notation "(p d q)(P D Q)" where:

p	nonseasonal AR order
d	order of nonseasonal differences
q	nonseasonal MA order
P	(multiplicative) seasonal AR order
D	order of seasonal differences
Q	(multiplicative) seasonal MA order

The default specification "(0 1 1)(0 1 1)" is the seasonal IMA model:
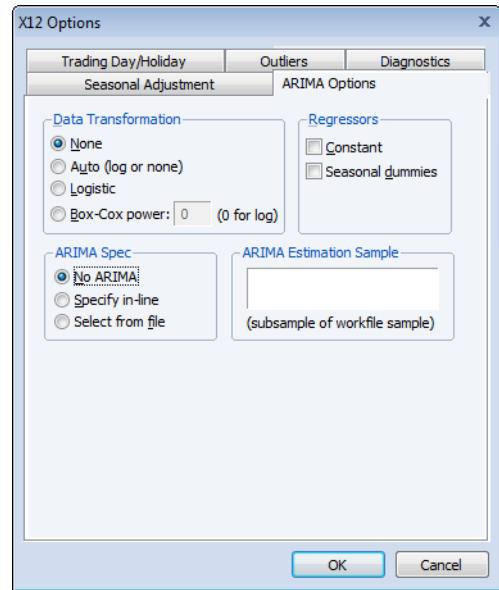
$$(1 - L)(1 - L^s)y_t = (1 - \theta_1 L)(1 - \theta_s L^s)\epsilon_t \tag{11.43}$$

Here are some other examples (L is the lag operator):

(1 0 0)	$(1 - \phi L)y_t = \epsilon_t$
(0 1 1)	$(1 - L)y_t = (1 - \theta L)\epsilon_t$
(1 0 1)(1 0 0)	$(1 - \phi_1 L)(1 - \phi_s L^s)y_t = (1 - \theta L)\epsilon_t$ where $s = 4$ for quarterly data and $s = 12$ for monthly data.

You can skip lags using square brackets and explicitly specify the seasonal order after the parentheses:

([2 3] 0 0)	$(1 - \phi_2 L^2 - \phi_3 L^3)y_t = \epsilon_t$
(0 1 1)12	$(1 - L^{12})y_t = (1 - \theta L^{12})\epsilon_t$

See the *X12-ARIMA Reference Manual* (p. 110–114) for further details and examples of ARIMA specification in X12. Note that there is a limit of 25 total AR, MA, and differencing coefficients in a model and that the maximum lag of any AR or MA parameter is 24 and the maximum number of differences in any ARIMA factor is 3.

Alternatively, if you choose **Select from file**, X12 will select an ARIMA model from a set of possible specifications provided in an external file. The selection process is based on a procedure developed by Statistics Canada for X11-ARIMA/88 and is described in the *X12-ARIMA Reference Manual* (p. 133). If you use this option, you will be asked to provide the name of a file that contains a set of possible ARIMA specifications. By default, EViews will use a file named X12A.MDL that contains a set of default specifications provided by Census (the list of specifications contained in this file is given below).

To provide your own list in a file, the ARIMA specification must follow the X12 syntax as explained in the ARIMA Specification section above. You must specify each model on a separate line, with an "X" at the end of each line except the last. You may also designate one of the models as a "default" model by marking the end of a line with an asterisk "*" instead of "X"; see p. 133 of the *X12-ARIMA Reference Manual* for an explanation of the use of a default model. To ensure that the last line is read, it should be terminated by hitting the return key.

For example, the default file (X12A.MDL) provided by X12 contains the following specifications:

```
(0 1 1)(0 1 1) *
```

```
(0 1 2)(0 1 1) x
(2 1 0)(0 1 1) x
(0 2 2)(0 1 1) x
(2 1 2)(0 1 1)
```

There are two additional options for **Select from file**. **Select best** checks all models in the list and looks for the model with minimum forecast error; the default is to select the first model that satisfies the model selection criteria. **Select by out-of-sample-fit** uses out-of-sample forecast errors (by leaving out some of the observations in the sample) for model evaluation; the default is to use within-sample forecast errors.

The **Regressors** option allows you to include pre-specified sets of exogenous regressors in your ARIMA model. Simply use the checkboxes to specify a constant term and/or (centered) seasonal dummy variables. Additional predefined regressors to capture trading day and/or holiday effects may be specified using the **Trading Day/Holiday** tab. You can also use the **Outlier** tab to capture outlier effects.

Trading Day and Holiday Effects

X12 provides options for handling trading day and/or holiday effects. To access these options, select the **Trading Day/Holiday** tab in the **X12 Options** dialog.

As a first step you should indicate whether you wish to make these adjustments in the ARIMA step or in the X11 seasonal adjustment step. To understand the distinction, note that there are two main procedures in the X12 program: the X11 seasonal adjustment step, and the ARIMA estimation step. The X11 step itself consists of several steps that decompose the series into the trend/cycle/irregular components. The X12 procedure may therefore be described as follows:

- optional preliminary X11 step (remove trading day/holiday effects from series, if requested).

- ARIMA step: fit an ARIMA model (with trading/holiday effects, if specified) to the series from step 1 or to the raw series.

- X11 step: seasonally adjust the series from step 2 using backcasts/forecasts from the ARIMA model.

While it is possible to perform trading day/holiday adjustments in *both* the X11 step and the ARIMA step, Census recommends against doing so (with a preference to performing the adjustment in the ARIMA step). EViews follows this advice by allowing you to perform the adjustment in only one of the two steps.

If you choose to perform the adjustment in the X11 step, there is an additional setting to consider. The checkbox **Apply only if significant (AIC)** instructs EViews to adjust only if warranted by examination of the Akaike information criterion.

It is worth noting that in X11, the significance tests for use of trading day/holiday adjustment are based on an *F*-test. For this, and a variety of other reasons, the X12 procedure with "X11 settings" will not produce results that match those obtained from historical X11. To obtain comparable results, you must use the historical X11 procedure (see "Census X11 (Historical)" on page 490).

Once you select your adjustment method, the dialog will present additional adjustment options:

- **Trading Day Effects** — There are two options for trading day effects, depending on whether the series is a flow series or a stock series (such as inventories). For a flow series, you may adjust for day-of-week effects or only for weekday-weekend contrasts. Trading day effects for stock series are available only for monthly series and the day of the month in which the series is observed must be provided.

- **Holiday Effects** — Holiday effect adjustments apply only to flow series. For each holiday effect, you must provide a number that specifies the duration of that effect *prior* to the holiday. For example, if you select 8, the level of daily activity changes on the seventh day *before* the holiday and remains at the new level until the holiday (or a day before the holiday, depending on the holiday).

Note that the holidays are as defined for the United States and may not apply to other countries. For further details, see the *X12-ARIMA Reference Manual*, Tables 6–15 (p. 94) and 6–18 (p. 133).

Outlier Effects

As with trading day/holiday adjustments, outlier effects can be adjusted either in the X11 step or in the ARIMA step (see the discussion in "Trading Day and Holiday Effects" on page 486). However, outlier adjustments in the X11 step are done only to robustify the trading day/holiday adjustments in the X11 step. Therefore, in order to perform outlier adjustment in the X11 step, you must perform trading day/ holiday adjustment in the X11 step. Only additive outliers are allowed in the X11 step; other types of outliers are available in the ARIMA step. For further information on the various types of outliers, see the *X12-ARIMA Reference Manual*, Tables 6–15 (p. 94) and 6–18 (p. 133).

If you do not know the exact date of an outlier, you may ask the program to test for an outlier using the built-in X12 diagnostics.

Diagnostics

This tab provides options for various diagnostics. The **Sliding spans** and **Historical revisions** options test for stability of the adjusted series. While **Sliding spans** checks the change in adjusted series over a moving sample of fixed size (overlapping subspans), **Historical revisions** checks the change in adjusted series over an increasing sample as new observations are added to the sample. See the *X12-ARIMA Reference Manual* for further details and references of the testing procedure. You may also choose to display various diagnostic output:

- **Residual diagnostics** will report standard residual diagnostics (such as the autocorrelation functions and Q-statistics). These diagnos-

tics may be used to assess the adequacy of the fitted ARIMA model. Note that this option requires estimation of an ARIMA model; if you do not provide an ARIMA model nor any exogenous regressors (including those from the **Trading day/Holiday** or **Outlier** tab), the diagnostics will be applied to the original series.
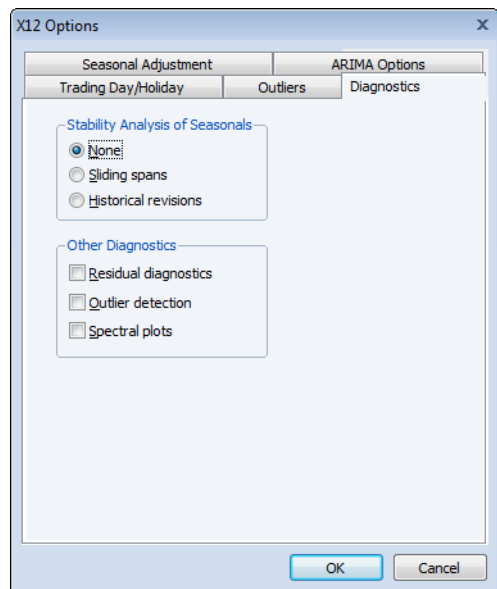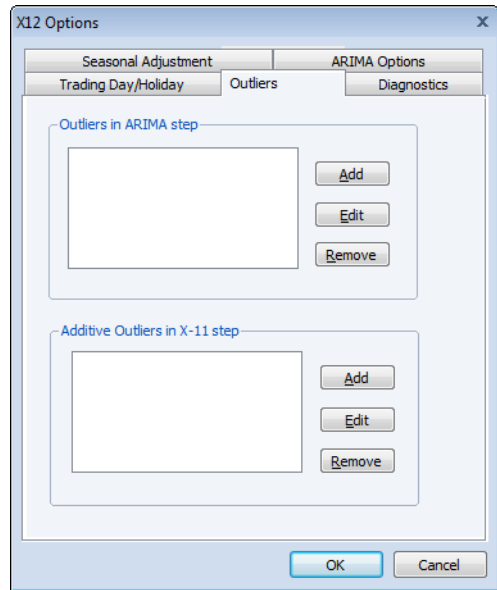
- **Outlier detection** automatically detects and reports outliers using the specified ARIMA model. This option requires an ARIMA specification or at least one exogenous regressor (including those from the **Trading day/Holiday** or **Outlier** tab); if no regression model is specified, the option is ignored.

- **Spectral plots** displays the spectra of the differenced seasonally adjusted series (SP1) and/or of the outlier modified irregular series (SP2). The red vertical dotted lines are the seasonal frequencies and the black vertical dashed lines are the trading day frequencies. If you observe peaks at these vertical lines it is an indication of inadequate adjustment. For further details, see Findley *et al.* (1998, section 2.1). If you request this option, data for the spectra will be stored in a matrix named *seriesname*_SA_SP1 and *seriesname*_SA_SP2 in your workfile. The first column of these matrices are the frequencies and the second column are 10 times the log spectra at the corresponding frequency.

X11/X12 Troubleshooting

The currently shipping versions of X11 and X12 as distributed by the Census have the following limitation regarding directory length. First, you will not be able to run X11/X12 if you are running EViews from a shared directory on a server which has spaces in its name. The solution is to map that directory to a letter drive on your local machine. Second, the temporary directory path used by EViews to read and write data cannot have more than four subdirectories. This temporary directory used by EViews can be changed by selecting **Options/ File Locations.../Temp File Path** in the main menu. If your temporary directory has more than four subdirectories, change the Temp File Path to a writeable path that has fewer subdirectories. Note that if the path contains spaces or has more than 8 characters, it may appear in shortened form compatible with the old DOS convention.

Census X11 (Historical)

The Census X11.2 methods (multiplicative and additive) are the standard methods used by the U.S. Bureau of Census to seasonally adjust publicly released data. The X11 routines are separate programs provided by the Census and are installed in the EViews directory in the files X11Q2.EXE and X11SS.EXE. The documentation for these programs can also be found in your EViews directory as text files X11DOC1.TXT through X11DOC3.TXT.
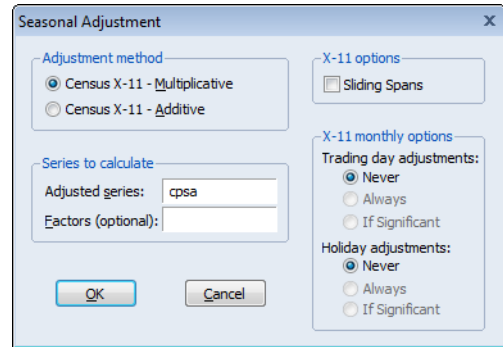
The X11 programs may be executed directly from DOS or from within EViews. If you run the X11 programs from within EViews, the adjusted series and the factor series will be automatically imported into your EViews workfile. X11 summary output and error messages will also be displayed in the series window at the end of the procedure.

The X11 method has many options, the most important of which are available in the Seasonal Adjustment dialog. However, there are other options not available in the EViews dialog; to use these other options, you should run the X11 programs from the DOS command line. All options available in the X11 methods are described in the X11DOC text files in your EViews directory.

You should note that there is a limit on the number of observations that you can seasonally adjust. X11 only works for quarterly and monthly frequencies, requires at least four full years of data, and can adjust only up to 20 years of monthly data and up to 30 years of quarterly data.

Tramo/Seats

Tramo ("Time Series Regression with ARIMA Noise, Missing Observations, and Outliers") performs estimation, forecasting, and interpolation of regression models with missing observations and ARIMA errors, in the presence of possibly several types of outliers. *Seats* ("Signal Extraction in ARIMA Time Series") performs an ARIMA-based decomposition of an observed time series into unobserved components. The two programs were developed by Victor Gomez and Agustin Maravall.

Used together, Tramo and Seats provide a commonly used alternative to the Census X12 program for seasonally adjusting a series. Typically, individuals will first "linearize" a series using Tramo and will then decompose the linearized series using Seats.

EViews provides a convenient front-end to the Tramo/Seats programs as a series proc. Simply select **Proc/Seasonal Adjustment/Tramo Seats...** and fill out the dialog. EViews writes an input file which is passed to Tramo/Seats via a call to a .DLL, and reads the output files from Tramo/Seats back into EViews (note: since EViews uses a new .DLL version of Tramo/Seats, results may differ from the older DOS version of the program).

Since EViews only provides an interface to an external program, we cannot provide any technical details or support for Tramo/Seats itself. Users who are interested in the technical details should consult the original documentation *Instructions for the User* which is provided as a .PDF file in the DOCS/TRAMOSEATS subdirectory of your EViews directory.

Dialog Options

The Tramo/Seats interface from the dialog provides access to the most frequently used options. Users who desire more control over the execution of Tramo/Seats may use the command line form of the procedure as documented in `tramoseats` (p. 612) in the *Command and Programming Reference*.

The dialog contains three tabs. The main tab controls the basic specification of your Tramo/Seats run.

- **Run mode**: You can choose either to run only Tramo or you can select the **Run Seats after Tramo** checkbox to run both. In the latter case, EViews uses the input file produced by Tramo to run Seats. If you wish to run only Seats, you must use the command line interface.

- **Forecast horizon**: You may set the number of periods to forecast outside the current sample. If you choose a number smaller than the number of forecasts required to run Seats, Tramo will automatically lengthen the forecast horizon as required.

- **Transformation**: Tramo/Seats is based on an ARIMA model of the series. You may choose to fit the ARIMA model to the level of the series or to the (natural) log of the series, or you select **Auto select level or log**. This option automatically chooses between the level model and the log transformed model using results from a trimmed
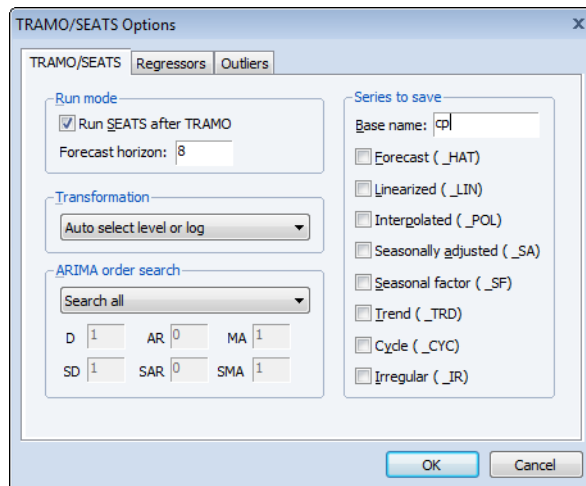
range-mean regression; see the original Tramo/Seats documentation for further details.

- **ARIMA order search**: You may either specify the orders of the ARIMA model to fit or ask Tramo to search for the "best" ARIMA model. If you select **Fix order** in the drop-down menu and specify the order of all of the ARIMA components, Tramo will use the specified values for all components where the implied ARIMA model is of the form:

$$y_t = x_t'\beta + u_t$$

$$\phi(L)\delta(L)u_t = \theta(L)\epsilon_t$$

$$\delta(L) = (1 - L)^D (1 - L^s)^{SD}$$

$$\phi(L) = (1 + \phi_1 L + \dots + \phi_{AR}L^{AR})(1 + \Phi_1 L^s + \dots + \Phi_{SAR}(L^s)^{SAR})$$

$$\theta(L) = (1 + \theta_1 L + \dots + \theta_{MA}L^{MA})(1 + \Theta_1 L^s + \dots + \Theta_{SMA}(L^s)^{SMA})$$

with seasonal frequency s. When you fix the order of your ARIMA you should specify non-negative integers in the edit fields for D, SD, AR, SAR, MA, and SMA.

Alternatively, if you select **Fix only difference orders**, Tramo will search for the best ARMA model for differenced data of the orders specified in the edit fields.

You can also instruct Tramo to choose all orders. Simply choose **Search all** or **Search all and unit complex roots** to have Tramo find the best ARIMA model subject to limitations imposed by Tramo. The two options differ in the handling of complex roots. Details are provided in the original Tramo/Seats documentation.

Warning: if you choose to run Seats after Tramo, note that Seats has the following limit on the ARIMA orders: $D \leq 3$, $AR \leq 3$, $MA \leq 3$, $SD \leq 2$, $SAR \leq 1$, $SMA \leq 1$.

- **Series to Save**: To save series output by Tramo/Seats in your workfile, provide a valid base name and check the series you wish to save. The saved series will have a postfix appended to the basename as indicated in the dialog. If the saved series contains only missing values, it indicates that Tramo/Seats did not return the requested series see "Trouble Shooting" on page 494.

If Tramo/Seats returns forecasts for the selected series, EViews will append them at the end of the stored series. The workfile *range* must have enough observations after the current workfile *sample* to store these forecasts.

If you need access to series that are not listed in the dialog options, see "Trouble Shooting" on page 494.

- **User specified exogenous series**: You may provide your own exogenous series to be used by Tramo. These must be a named series or a group in the current workfile and should not contain any missing values in the current sample and the forecast period.

If you selected a trading day adjustment option, you have the option of specifying exogenous series to be treated as a holiday series. The specification of the holiday series will depend on whether you chose a weekday/weekend adjustment or a 5-day adjustment. See the original Tramo/Seats documentation for further details.

If you are running Seats after Tramo, you must specify which component to allocate the regression effects. The Tramo default is to treat the regression effect as a separate additional component which is not included in the seasonally adjusted series.

EViews will write a separate data file for each entry in the exogenous series list which is passed to Tramo. If you have many exogenous series with the same specification, it is best to put them into one group.

- **Easter/Trading day adjustment**: These options are intended for monthly data; see the original Tramo/Seats documentation for details.

- **Outlier detection**: You may either ask Tramo to automatically detect possible outliers or you can specify your own outlier but not both. If you wish to do both, create a series corresponding to the known outlier and pass it as an exogenous series.

  Similarly, the built-in intervention option in Tramo is not supported from the dialog. You may obtain the same result by creating the intervention series in EViews and passing it as an exogenous series. See the example below.

  The original Tramo/Seats documentation provides definitions of the various outlier types and the method to detect them.

After you click **OK**, the series window will display the text output returned by Tramo/Seats. If you ran both Tramo and Seats, the output from Seats is appended at the end of Tramo output. Note that this text view will be lost if you change the series view. You should freeze the view into a text object if you wish to refer to the output file without having to run Tramo/Seats again.

It is worth noting that when you run Tramo/Seats, the dialog will generally contain the settings from the previous run of Tramo/Seats. A possible exception is the user specified outlier list which is cleared unless Tramo/Seats is called on the previously used series.

Comparing X12 and Tramo/Seats

Both X12 and Tramo/Seats are seasonal adjustment procedures based on extracting components from a given series. Methodologically, X12 uses a non-parametric moving average based method to extract its components, while Tramo/Seats bases its decomposition on an estimated parametric ARIMA model (the recent addition of ARIMA modelling in X12 appears to be used mainly to identify outliers and to obtain backcasts and forecasts for end-of-sample problems encountered when applying moving average methods.)

For the practitioner, the main difference between the two methods is that X12 does not allow missing values while Tramo/Seats will interpolate the missing values (based on the esti-

mated ARIMA model). While both handle quarterly and monthly data, Tramo/Seats also handles annual and semi-annual data. See the sample programs in the **Example Files** directory for a few results that compare X12 and Tramo/Seats.

Trouble Shooting

Error handling

As mentioned elsewhere, EViews writes an input file which is passed to Tramo/Seats via a call to a .DLL. Currently the Tramo/Seats .DLL does not return error codes. Therefore, the only way to tell that something went wrong is to examine the output file. If you get an error message indicating that the output file was not found, the first thing you should do is to check for errors in the input file.

When you call Tramo/Seats, EViews creates two subdirectories called Tramo and Seats in a temporary directory. This temporary directory is taken from the global option **Options/File Locations.../Temp File Path** (note that long directory names with spaces may appear in shortened DOS form). The Temp File Path can be retrieved in a program by a call to `@temp-path` (see `@temppath` (p. 755) in the *Command and Programming Reference*).

The Tramo input file written by EViews will be placed in the subdirectory TRAMO and is named SERIE. A Seats input file written by Tramo is also placed in subdirectory TRAMO and is named SEATS.ITR.

The input file used by Seats is located in the SEATS subdirectory and is named SERIE2. If Seats is run alone, then EViews will create the SERIE2 file. When Tramo and Seats are called together, the Tramo file SEATS.ITR is copied into SERIE2.

If you encounter the error message containing the expression "output file not found", it probably means that Tramo/Seats encountered an error in one of the input files. You should look for the input files SERIE and SERIE2 in your temp directories and check for any errors in these files.

Retrieving additional output

The output file displayed in the series window is placed in the OUTPUT subdirectory of the TRAMO and/or SEATS directories. The saved series are read from the files returned by Tramo/Seats that are placed in the GRAPH subdirectories. If you need to access other data files returned by Tramo/Seats that are not supported by EViews, you will have to read them back into the workfile using the `read` command from these GRAPH subdirectories. See the PDF documentation file for a description of these data file formats.

Warning: if you wish to examine these files, make sure to read these data files before you run the next Tramo/Seats procedure. EViews will clear these subdirectories before running the next Tramo/Seats command (this clearing is performed as a precautionary measure so that Tramo/Seats does not read results from a previous run).

Moving Average Methods

Ratio to moving average—multiplicative

The algorithm works as follows. Denote the series to be filtered by y_t.

1. First compute the centered moving average of y_t as:

$$x_t = \begin{cases} (0.5\,y_{t+6} + \ldots + y_t + \ldots 0.5\,y_{t-6})/12 & \text{if monthly} \\ (0.5\,y_{t+2} + y_{t+1} + y_t + y_{t-1} + 0.5\,y_{t-1})/4 & \text{if quarterly} \end{cases} \qquad (11.44)$$

2. Take the ratio $\tau_t = y_t / x_t$.

3. Compute the seasonal indices. For monthly series, the seasonal index i_m for month m is the average of τ_t using observations only for month m. For quarterly series, the seasonal index i_q for quarter q is the average of τ_t using observations only for quarter q.

4. We then adjust the seasonal indices so that they multiply to one. This is done by computing the seasonal factors as the ratio of the seasonal index to the geometric mean of the indices:

$$s = \begin{cases} i_m / (\sqrt[12]{i_1 i_2 \ldots i_{12}}) & \text{if monthly} \\ i_q / (\sqrt[4]{i_1 i_2 i_3 i_4}) & \text{if quarterly} \end{cases} \qquad (11.45)$$

5. These s are the reported *scaling factors* in the series window and are saved as series if you provide a name in the field box. The interpretation is that the series y is s_j percent higher in period j relative to the adjusted series.

6. The seasonally adjusted series is obtained by dividing y_t by the seasonal factors s_j.

Difference from moving average—additive

Suppose that we wish to filter y_t.

1. First compute the centered moving average of y_t as in .

2. Take the difference $d_t = y_t - x_t$.

3. Compute the seasonal indices. For monthly series, the seasonal index i_m for month m is the average of d_t using observations only for month m. For quarterly series, the seasonal index i_q for quarter q is the average of d_t using observations only for quarter q.

4. We then adjust the seasonal indices so that they add up to zero. This is done by setting $s_j = i_j - \bar{i}$ where $\bar{i}$ is the average of all seasonal indices. These s are the reported *scaling factors*. The interpretation is that the series y is s_j higher in period j relative to the adjusted series.

5. The seasonally adjusted series is obtained by subtracting the seasonal factors s_j from y_t.

The main difference between X11 and the moving average methods is that the seasonal factors may change from year to year in X11. The seasonal factors are assumed to be constant for the moving average method.

Automatic ARIMA Forecasting

Automatic ARIMA forecasting is a method of forecasting values for a single series based upon an ARIMA model.

Although EViews provides sophisticated tools for estimating and working with ARIMA models using the familiar equation object, there is considerable value in a quick-and-easy tool for performing this type of forecasting. EViews offers an automatic ARIMA forecasting series procedure that allows the user to quickly determine an appropriate ARIMAX specification and use it to forecast the series into the future.

Methodological Background

The series y_t follows an ARIMAX(p, d, q) model if:

$$
\begin{aligned}
D(y_t, d) &= \beta X_t + v_t \\
v_t &= \rho_1 v_{t-1} + \rho_2 v_{t-2} + \ldots + \rho_p v_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q}
\end{aligned}
\tag{11.46}
$$

(for notational simplicity, we ignore here the possibility of seasonal ARMA terms).

Often the exogenous variables X are simply a single constant or trend term. In such cases the only decision the forecaster has to make to set up his forecasts, is the form of the dependent variable, the level of differencing, and the number of AR and MA terms (*i.e.* – choose p and q). One method of choosing the number of AR and MA terms is through model selection/evaluation techniques.

ARIMAX models may be estimated through a number of different methods, including transforming the model into a non-linear least squares specification, or using GLS or maximum likelihood estimation. Since maximum likelihood estimation does not require dropping observations from the start of the sample, or backcasting to create observations, it lends itself nicely to model selection/comparison algorithms.

Automatic Model Specification

Specifying the ARIMAX model used for forecasting can be split into four steps:

1. Selecting any transformations of the dependent variable, such as taking logs.

2. Selecting the level of differencing of the dependent variable.

3. Selecting the exogenous regressors.

4. Selecting the order of the ARMA terms.

EViews' automatic forecasting procedure automatically performs steps 1., 2. and 4. The procedure will not select a set of exogenous regressors automatically, although it does allow the user to specify which regressors to include. As such we refer to the procedure as performing "automatic ARIMA forecasting", rather than "automatic ARIMAX forecasting".

Transformations and Differencing

Selecting a dependent variable transformation is often based on an underlying economic theory. Common transformations are to take logs, or to use the Box-Cox transformation. However it may be possible to determine whether taking logs is appropriate through a rule-of-thumb method which runs two simple regressions:

- $D(y_t)^2 = \alpha_1 + \beta_1 y_t$
- $D\log(y_t)^2 = \alpha_2 + \beta_2 \log(y_t)$

A comparison is then made between the t-statistic on β_1 and that on β_2. If the t-statistic on β_2 is lower than that on β_1, the log model is preferred.

The logic behind this test is that taking natural logs is often used on variables with an exponential growth rate — the change in growth increases or decreases over time. Such data, when used in a least squares estimation involving differences, will suffer from heteroskedasticity (since the change, or difference, in the data is not constant). Taking logs linearizes the relationship, and alleviates the problem of heteroskedasticity. Each of the two regressions is a simple, crude, test for heteroskedasticity, with a low t-statistic on β suggesting homoskedasticity rather than heteroskedasticity. Regression 1 being "more homoskedastic" than regression 2 would suggest the data does not need to be logged. Conversely, regression 2 being "more homoskedastic" than regression 1 suggests the data should be logged.

Step 2 is to decide upon appropriate level of differencing for the (possibly transformed) dependent variable. Following the suggestion of Hyndman and Khandakar (HK, 2008), EViews uses successive unit root tests to determine the correct level of differencing. HK recommend preferring under-differencing the model as opposed to over-differencing when forecasting (themselves following the earlier work of Smith and Yadav 1994). Consequently HK suggest using a unit-root test with the null-hypothesis of no unit-root, such as the KPSS test.

The KPSS test is first run on the data with no differencing. If the test rejects the null hypothesis, the data is then differenced and the test run again. This continues until we can no longer reject the null hypothesis.

ARMA Selection

EViews uses model selection to determine the appropriate ARMA order. Model selection is a way of determining which type of model best fits a set of data, and is often used to choose the best model from which to forecast that data.

Information Criteria

Information criteria are the most common model selection tool used in econometrics. EViews supports three types of information criteria for most estimation methods; Akaike Information Criterion (AIC), Schwarz Criterion (SIC or BIC), and the Hannan-Quinn Criterion (HQ). Each of these criteria are based upon the estimated log-likelihood of the model, the number of parameters in the model and the number of observations. Additional detail may be found in Appendix E. "Information Criteria," beginning on page 1111 of *User's Guide II*.

One issue with using information criteria is that the models not only need to be estimated on the same set of observations across models, but the dependent variable must also be of the same scale. You cannot, generally, evaluate models across transformations and differences of the dependent variable.

Thus information criteria based model selection can only be used in ARIMA models to determine the number of ARMA terms. It cannot be used to determine any transformations or differencing of the dependent variable. Determining the number of ARMA terms is typically done by specifying a maximum number of AR or MA coefficients, then estimating every model up to those maxima, and then evaluating each model using its information criterion.

Estimation of ARIMAX models by maximum likelihood makes comparison of different models using information criteria simple, since the log-likelihood is estimated as part of the estimation procedure. Once each model is estimated, its criterion can be calculated, and then the model with the lowest criterion value is chosen.

Mean Square Error (MSE) Evaluation

A second method of model selection is that of in-sample forecast evaluation. Here each model is estimated on a sub-sample of data (usually the first 80%-90%), and then forecasted over the remaining data (the remaining 10%-20%). Since we have data for the actual values of the dependent variable over the sub-sample forecast period, we can compare the forecasts with the actual data and calculate the mean square error (MSE):

$$MSE = \frac{1}{h} \sum_{t = T - h}^{T} (y_t - \hat{y}_t)^2 \tag{11.47}$$

where h is the number of periods in the forecast subsample, and T is the number of periods in the full sample.

Each model is estimated and forecasted over the smaller sample, and the model with the smallest MSE is chosen.

Starting Values

Maximum likelihood estimation of ARMA models requires starting values for the coefficients. EViews' automatic ARMA estimation routine uses a data-based algorithm to determine starting values. However if estimation using these starting values fails to converge, EViews will then try a set of fixed starting values. If that estimation too fails to converge, EViews will finally try a set of random starting values.

Forecasting

Once the best model's transformation, differencing and ARMA length has been selected, either through information criteria or via MSE, the model is used to calculate the final forecast.

Forecast Averaging

An alternative approach to selecting the "best" ARIMA model and then forecasting from it is to forecast from each of the individual ARIMA specifications under consideration, and then average over those forecasts to produce a final forecast.

EViews allows to forms of forecast averaging when performing automatic ARIMA forecasting; Smoothed Akaike Information Criterion (SAIC), and Bayesian Model Averaging (BMA). For details on these forecast averaging methods, see "Forecast Averaging" on page 505.

Since both of these methods are based upon information criteria, the same restrictions apply to them as to using information criteria for model selection—namely the samples used for estimation must be the same, and the dependent variable (the variable to be forecasted) must have the same scale. Consequently, when performing forecast averaging on ARIMAX models, only the subset of forecasts from models with the same transformation and differencing can be averaged.

When performing forecast averaging under automatic ARIMA forecasting, EViews then selects the form of transformation and differencing using the methods outlined above, and then forecasts from each combination of ARMA order to produce the set of forecasts available for averaging. The final produced forecast is then the weighted average of those forecasts.

Performing Automatic ARIMA Forecasting in EViews

To forecast a series automatically using ARIMA models, open up the series and click on **Proc/Automatic ARIMA...** which will bring up the **Automatic ARIMA Forecasting** dialog:
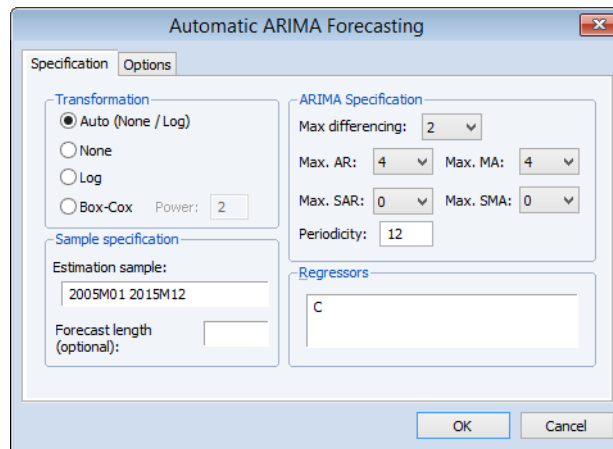
The first section of the **Specification** tab of the dialog allows selection of the type of dependent variable transformation by using the **Transformation** radio buttons. The default selection, **Auto (None/Log)**, instructs EViews to perform the rule-of-thumb test outlined above to determine whether to log the dependent variable or not. The remaining choices perform no transformation, take logs, or use the Box-Cox transformation. If **Box-Cox** is selected, a power parameter for the transform must also be provided. **Auto** and **Log** should only be used if your data are strictly positive.

The **ARIMA Specification** area of the dialog selects the type of ARIMA models that will be used during model selection or forecast averaging. To select the maximum level of differencing to be tested use the **Max differencing** dropdown box. EViews will perform successive KPSS tests on different levels of differencing, starting from zero and stopping only when the null hypothesis of the KPSS test cannot be rejected, or the maximum level of differencing selected by the user is reached.

The **Max. AR**, **Max. MA**, **Max. SAR** and **Max. SMA** dropdowns select the maximum order of the AR, MA, SAR and SMA terms of the ARIMA model. The periodicity of the seasonal terms can be entered in the **Periodicity** box. If the workfile is dated, EViews will default the periodicity to the number of observations per year, but this may be overwritten to model non-annual seasonalities.

The **Regressors** box allows entry of any exogenous regressors in the model. By default a constant is included.

The final section of this tab of the dialog includes the **Estimation Sample** box and the **Forecast length** box. **Estimation Sample** determines the observations used in determining the appropriate ARIMA model to use for forecasting - it specifies the observations used for the rule-of-thumb regressions determining whether to log the dependent variable or not, the

observations used in the successive KPSS tests for determining differencing order, as well as the observations used in the estimation of the individual ARMA models.

Forecast length specifies h, the number of observations that will be forecasted after estimation. The forecast sample will start immediately after the last observation of the estimation sample and will continue for h observations. Note the workfile must be sized such that h observations exist in the workfile after the estimation sample.

The **Options** tab of the dialog provides further options on model selection and output:
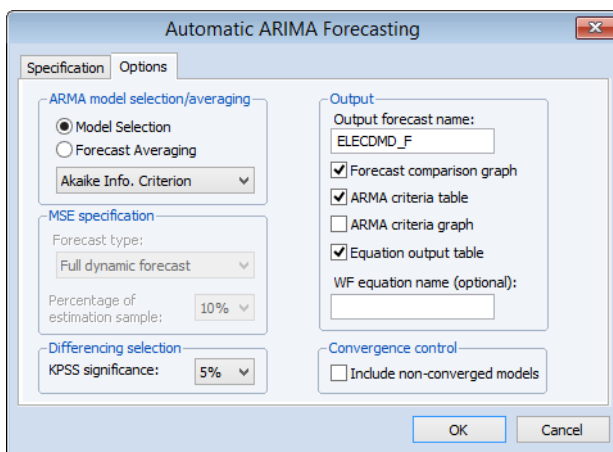
The **ARMA model selection/averaging** box selects the method used to choose the appropriate ARMA model, or the method of forecast averaging. The **Model Selection** and **Forecast Averaging** radio button select whether to use model selection or forecast averaging, with the dropdown box below them allowing selection of which type of model selection (AIC, BIC, HQ or MSE based), or forecast averaging (SAIC or BMA) to use.

If MSE based model selection is used, the **MSE specification** area allows specification of the MSE calculations. **Forecast type** selects whether the in-sample forecast used to compute the MSE is a dynamic forecast or a static forecast. The **Percentage of estimation sample** dropdown selects the part of the estimation sample (chosen on the Specification tab) that is used for in-sample forecasting for calculating the MSE.

The **KPSS significance** dropdown specifies the significance level to use when determining whether the null hypothesis of the KPSS test is rejected or not during differencing selection.

The **Convergence control** section includes a checkbox for specifying whether to include non-converged models amongst those included in model selection or forecast averaging. If left unchecked, only ARMA estimations that EViews believes are fully converged will be included in the selection/averaging. If the output of the automatic ARIMA forecasting procedure indicates that a large number of models didn't converge, and it is believed this may be

due to border solutions or a very flat likelihood, checking this option may improve the accuracy of the final forecast.

The **Output** area allows customization of the output from the procedure. The **Output forecast name:** box is used to name the final forecast series in the workfile. By default it is filled in with the name of the underlying series followed by an "_F".

Checking the **Forecast comparison graph check** box will produce a graph containing the final forecast (either the forecast from the selected model, or the averaged forecast) along side the forecasts from every other ARMA model considered. The final forecast will be colored red, with the other forecasts in grey. Note, the graph is only displayed if the **Forecast length** specified on the **Specification** tab is greater than zero (*i.e.*—a forecast is actually performed).

The **ARMA criteria table** and **ARMA criteria graph** check boxes specify whether to include a table, or graph, of the "best" 20 models used during model selection or forecast averaging. The graph shows the model selection value for the twenty "best" models. If you use either the Akaike Information Criterion (AIC), the Schwarz Criterion (BIC), or the Hannan-Quinn (HQ) criterion, the graph will show the twenty models with the lowest criterion value. The table form of the view shows the log-likelihood value, the AIC, BIC and HQ values of the top twenty models in tabular form.

Finally, If **Model Selection** is chosen, selecting the **Equation output table** option produces a standard EViews ARMA equation output table of the final selected equation. Similarly, entering a name in the **WF equation name (optional)** box will create a new equation object in the workfile with the same specification as the final chosen model. Outputting an equation object allows performance of post-estimation diagnostics and tests.

Example

As an example of using automatic ARIMA forecasting in EViews, we forecast monthly electricity demand in England and Wales, using the workfile "elecdmd.wf1". This workfile contains monthly electricity demand data from April 2005 until April 2014 (in the series ELECDMD), as well as real GDP data for the UK (a good proxy for real GDP in England and Wales) and average monthly temperature (series TEMPF).

We will use automatic forecasting to forecast the ELECDMD series from May 2014 until December 2015. To do this we open up the series and click on **Proc/Automatic ARIMA Forecasting…**, which brings up the automatic ARIMA dialog:
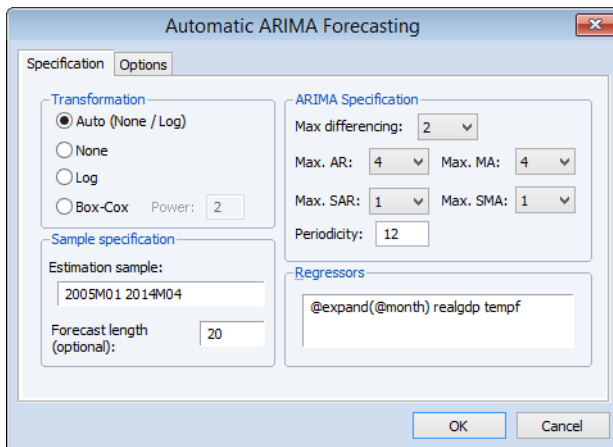
We'll let EViews decide on the best transformation by selecting **Auto (None / Log)** in the **Transformation** box.

We select an estimation sample of January 2005 until April 2014. In the **ARIMA Specification** area we'll leave most of the settings at their default values, other than, since our data has clear seasonal patterns, changing the maximum number of seasonal terms from 0 to 1.

We also add some monthly dummy variables using the `@expand(@month)` keyword, and add REALGDP and TEMPF as exogenous regressors.this

On the **Options** tab, we keep most of the settings at their default values.
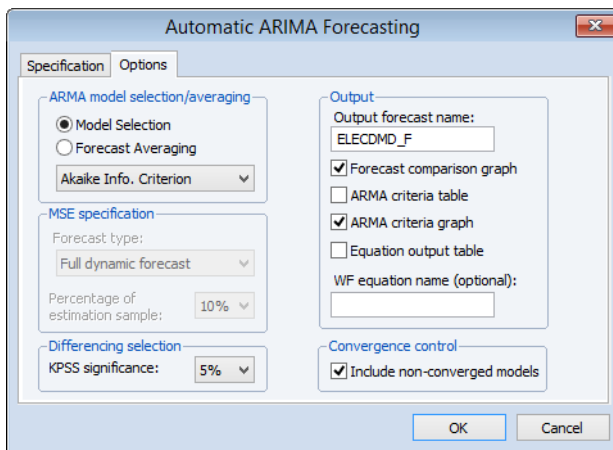
So that we may compare the forecasts of the ARIMA models, we select the **Forecast comparison graph** box.

The results of the auto-ARIMA estimation are shown below:

Automatic ARIMA Forecasting
Selected dependent variable: DLOG(ELECDMD)
Date: 03/07/15   Time: 22:02
Sample: 2005M01 2014M04
Included observations: 108
Forecast length: 20

Number of estimated ARMA models: 100
Number of non-converged estimations: 0
Selected ARMA model: (3,3)(0,1)
AIC value: -4.40741778342

The summary table indicates that out of the 100 different models estimated, the chosen ARMA was a (3,3)(0,1) model. The automatic transformation detection decided that logging our underlying series, ELECDMD, would provide a better model. A single order of difference was performed.
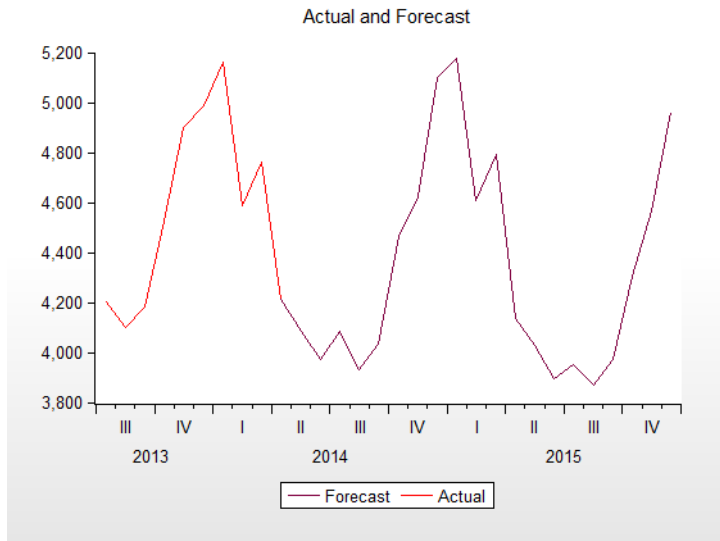
The Actual and Forecast graph indicates that the chosen model forecasted the actual values pretty well.

The Forecast Comparison Graph shows that each of the 100 models picked up the same cyclical patterns pretty well (undoubtedly due to the inclusion of our exogenous regressors).

Forecast Comparison Graph
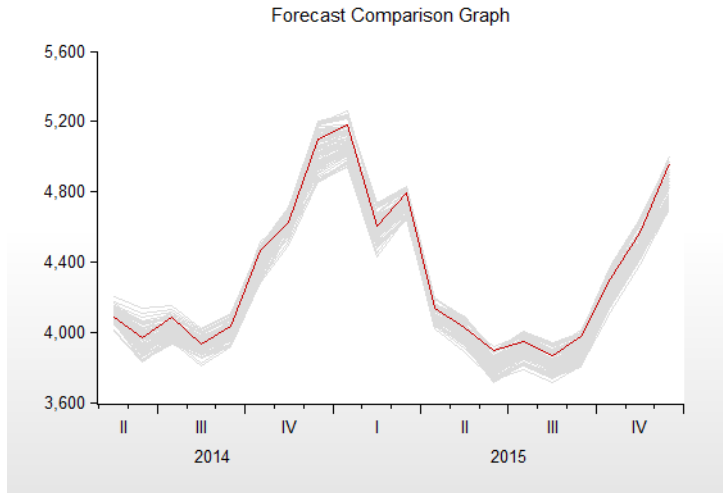
Forecast Averaging

Economic forecasters often have a variety of different models and forecasts of the same variable from which to choose. These models and forecasts may differ in the underlying assumptions, or may employ different information. Traditionally the forecasting decision was to pick which single forecast was "best" out of the individual forecasts available.

However, a number of studies have shown that averaging forecasts is more accurate than choosing a single best forecast. Timmermann (2006) offers a good overview of such studies, citing Clemen (1989) and Makridakis and Hibon (2000) in particular.

Forecast averaging, or forecast combining, is a methodology for combining multiple forecasts into a single forecast. The methods and ideas behind forecast averaging are founded in forecast evaluation and model selection techniques. Rather than calculating a measure of how good each forecast is, and then selecting the best forecast, the forecast evaluation measures are used to produce weights for calculating a weighted average of all the forecasts.

Methodology

Forecast averaging methods typically employ a simple weighted average computation—at each observation of the forecast sample, the cross-forecast average is computed, with different forecasts assigned different weights.

It is possible to compute complicated weighting schemes that allow the weights to vary through the forecast period. EViews, however, offers methods that generally employ constant weights through time: simple mean, least squares, mean square error, mean square

error ranks, smoothed AIC, approximate Bayesian model averaging. Notable exceptions where weights are time-varying are the trimmed mean and simple median methods).

We will briefly discuss each method in turn.

Simple Mean

The simple mean averaging method takes the arithmetic mean of the forecasts at each observation in the forecast sample. Note that in the simple mean method, every forecast is given the same weight.

Trimmed Mean

The trimmed mean is calculated in the same way as the simple mean, but at each observation the highest t % and the lowest t % of the forecast values are dropped from the mean calculation. Note that the selection of which forecasts to remove as part of the trimming is recalculated at each observation in the forecast sample so that the weights are time-varying—each forecast method may appear in the averaging calculation for some observations but not at others.

Simple Median

The simple median method calculates the median of the forecasts at every observation in the forecast sample. The implicit (0, 1) weights are time-varying as each forecast method may be the median for some observations but not others.

Least Squares Weights

Least squares weighting requires knowledge of the true values of the forecasted variable for some of the forecast period. It is calculated by regressing the forecasts against the actual values and then using the coefficients from the regression as weights. Details of this method are outlined in the Handbook of Economic Forecasting, Vol 1, page 159. EViews calculates the most general model which includes an intercept to the regression.

This method does not require that the underlying individual forecasts are unbiased, and the resulting average can lie outside the range of the underlying forecasts.

Mean Square Error (MSE) Weights

MSE weighting, proposed by Stock-Watson (2001), compares the individual forecasts with the actual values over some forecast period. The MSE of each forecast is computed and used to form individual forecast weights:

$$\omega_i = \frac{1/MSE_i^k}{\sum_{j=1}^{N} 1/MSE_j^k} \tag{11.48}$$

k is used to raise the MSE to different powers. $k = 1$ is the most commonly used power, and yields a weight based on the ratio of each forecast's MSE to the total of all the MSEs.

MSE Ranks

Aiolfi and Timmermann's (2006) MSE ranks method is similar to the MSE Weights method outlined above, but rather than computing the ratio of MSE vales, this method computes the MSE of each forecast, ranks them, and then computes the ratio of the inverse of the ranks. so that each forecast's weight is its rank divided by the sum of all ranks.

Smoothed AIC Weights

Smoothed AIC weights use the Akaike information criterion from the original estimation/ model that produced each forecast, then calculates the weights as:

$$\omega_i = \frac{\exp(-0.5\,AIC_i)}{\sum_{j=1}^{N} \exp(-0.5\,AIC_j)} \tag{11.49}$$

Smoothed AIC weights can only be computed if EViews has access to the underlying estimation results with associated AIC values.

Approximate Bayesian Model Averaging Weights

EViews calculates BMA weights in a similar fashion to Smoothed AIC weights, but uses the Schwarz criterion instead of the AIC:

$$\omega_i = \frac{\exp(-0.5\,BIC_i)}{\sum_{j=1}^{N} \exp(-0.5\,BIC_j)} \tag{11.50}$$

The approximate BMA uses the BIC to give a large sample approximation to computing Bayes factors. In finite samples the BIC may overly weight parsimonious models.

Forecast Averaging in EViews

To perform forecast averaging in EViews, you must start with a series representing the data to be forecasted. Some of the forecast weighting techniques require actual data to calculate the weights, and in these cases this series should contain actual values for the variable being forecasted for at least some of the observations for which forecast values are available.

To begin, open up the series and click on **Proc/Forecast Averaging...**, which will open the **Forecast Averaging** dialog box:
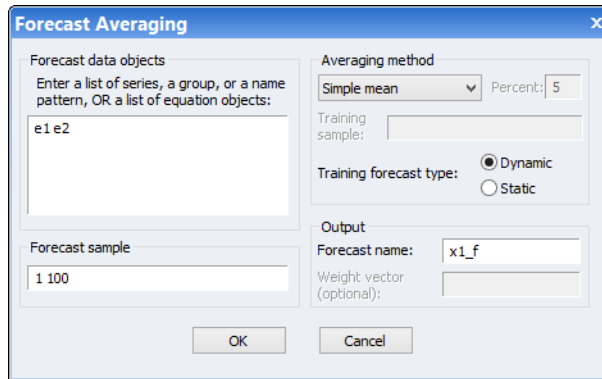
The **Forecast data objects** box specifies the forecasts to be used for averaging. Forecasts can be entered either as a collection of series (in which case the names of the series, a series naming pattern, or the name of a group are entered), or as a list of equation objects. If equation objects are entered, EViews will automatically perform a dynamic forecast over the forecast period from each of those equation objects to generate the forecast data.

When using equation objects, rather than forecast series, as the forecast data, the following should be noted:

- Each equation must have an identical dependent variable, which is identical to the series from which you are performing the forecast evaluation. i.e., if you are forecasting from series Y, each equation must have Y as the dependent variable. Currently transformations (such as LOG(Y)) are not allowed.

- If using smoothed AIC or BMA/SIC averaging methods, the weight calculations are only strictly valid if the underlying estimation objects were estimated on identical samples. It is up to the user to ensure that the samples are identical.

- Only equation objects are allowed. If a different type of estimation object (system, VAR, Sspace, *etc.*) is used, or if forecast was obtained from a non-EViews estimation source, the forecasts cannot be specified by equation.

- If using one of the MSE based or the OLS based weighting methods, historical forecasts (along with actual values) are needed for use in the weighting calculation. Note that EViews will not re-estimate the equations, it will use the same coefficient values for both the historical forecast and the actual forecasts, based on whatever sample was used when the equation were originally estimated. If you wish to use different estimation samples for the comparison forecast and actual forecast, you must perform the estimation and forecasts manually and specify the forecast data by series.

The **Forecast sample** box specifies the sample over which the forecasts will be averaged.

The **Averaging method** area specifies the forecast averaging procedure. The first dropdown selects the specific method of averaging. If the **Trimmed mean** averaging method is selected, the **Percent:** box specifies the level of trimming (from both ends). If the **Mean square error** method is selected, the Power: box specifies the power to which the MSE is raised.

Note the **Smooth AIC weights** and **SIC weights** selections are only available if a list of equations is entered in the **Forecast data objects** box, since they require information from the estimation rather than just the raw forecast data.

The Least-squares, Mean square error, MSE ranks, Smooth AIC weights, and SIC weights averaging methods require a training sample - a sample over which the averaging weights are computed. If any of these averaging methods are selected, a sample must be entered in the **Training sample:** box. If a list of equations is entered in the **Forecast data objects** box, the **Training forecast type:** radio buttons select which type of forecast is used over the training sample.
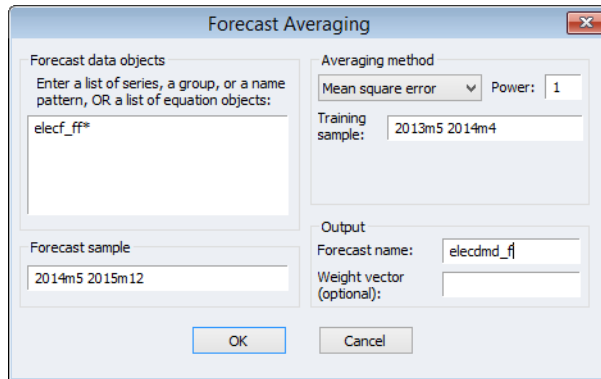
The **Output** section allows specification of the name of the final averaged forecast series (by default given a name equal to the underlying series plus "_f". If the selected averaging method requires calculation of averaging weights, the weighting vector can be saved into the workfile by entering a name in the **Weight vector (optional):** box.

An Example

As an example of forecast averaging in EViews, we average six monthly forecasts of electricity demand in England and Wales, using the workfile elecdmd.wf1. This workfile contains monthly electricity demand data from April 2005 until April 2014 (in the series ELECDMD), along with five evaluation sample forecasts of electricity demand (series ELECF_FE1-5), and five out-of-sample forecasts (series ELECF_FF1-5). The different forecast series correspond to different five different models used to generate forecasts.

Each of the out-of-sample forecast series contains actual data until April 2013, and then forecast data from May 2013 until December 2015. There is a year of overlap (May 2013 until April 2014) where actual data exists during the forecast period.

We will use forecast averaging to create a combination forecast from our five individual forecast series. To begin, we open the ELECDMD series and click on **Proc/Forecast Averaging...** and enter the names of our forecast series in the **Forecast data** objects box of the **Forecast Averaging** dialog, using the naming patter ELEC_FF* (to indicate any series matching that pattern):

We set the forecast sample to "2014m5 2015m12". We select **Mean square error** as the average method, with data between 2013M5 and 2014M4 used to calculate the MSE. We save the final averaged series in a new series called ELECDMD_F.

The output of the procedure is as follows:
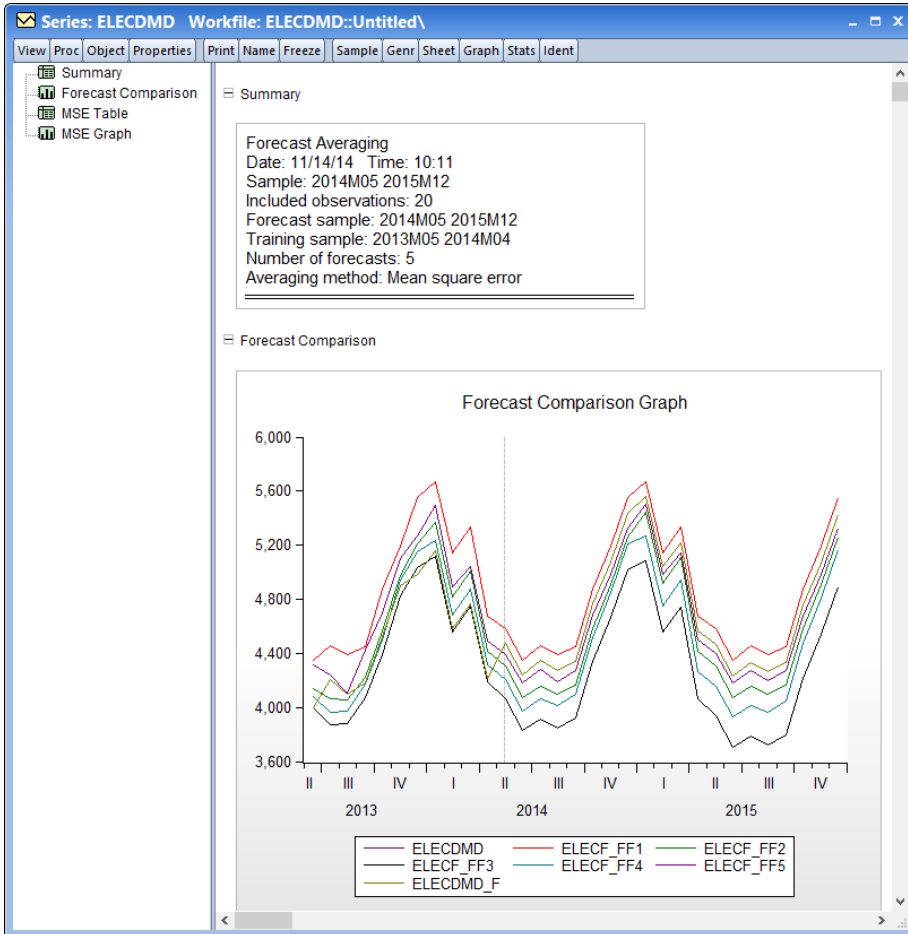
The top of the output provides summary information about the averaging procedure, including the time and date it was performed, the number of observations included (20 in this case) and the number of forecasts used in the averaging calculation, as well as the method used.

EViews also produces a graph of each of the individual forecasts, and the calculated average, allowing a quick visual comparison.

Exponential Smoothing

Exponential smoothing is a simple method of adaptive forecasting. It is an effective way of forecasting when you have only a few observations on which to base your forecast. Unlike forecasts from regression models which use fixed coefficients, forecasts from exponential

smoothing methods adjust based upon past forecast errors. For additional discussion, see Bowerman and O'Connell (1979).

EViews offers two methods of performing exponential smoothing: simple exponential smoothing based on older *ad hoc* methods, and exponential smoothing based on the Error-Trend-Seasonal likelihood framework of Hyndman, Koehler, *et al.* (2002).

Simple Exponential Smoothing

To obtain forecasts based on exponential smoothing methods, choose **Proc/Exponential Smoothing/Simple Exponential Smoothing...**. The **Exponential Smoothing** dialog box appears:

You need to provide the following information:

- **Smoothing Method**. You have the option to choose one of the five methods listed.

- **Smoothing Parameters**. You can either specify the values of the smoothing parameters or let EViews estimate them.

To estimate the parameter, type the letter e (for estimate) in the edit field. EViews estimates the parameters by minimizing the sum of squared errors. Don't be surprised if the estimated damping parameters are close to one—it is a sign that the series is close to a random walk, where the most recent value is the best estimate of future values.

To specify a number, type the number in the field corresponding to the parameter. All parameters are constrained to be between 0 and 1; if you specify a number outside the unit interval, EViews will estimate the parameter.

- **Smoothed Series Name**. You should provide a name for the smoothed series. By default, EViews will generate a name by appending SM to the original series name, but you can enter any valid EViews name.

- **Estimation Sample**. You must specify the sample period upon which to base your forecasts (whether you choose to estimate the parameters or not). The default is the current workfile sample. EViews will calculate forecasts starting from the first observation after the end of the estimation sample.

- **Cycle for Seasonal**. You can change the number of seasons per year from the default of 12 for monthly or 4 for quarterly series. This option allows you to forecast from unusual data such as an undated workfile. Enter a number for the cycle in this field.

Single Smoothing (one parameter)

This single exponential smoothing method is appropriate for series that move randomly above and below a constant mean with no trend nor seasonal patterns. The smoothed series $\hat{y}_t$ of y_t is computed recursively, by evaluating:

$$\hat{y}_t = \alpha y_t + (1 - \alpha)\hat{y}_{t-1} \tag{11.51}$$

where $0 < \alpha \leq 1$ is the *damping* (or *smoothing*) factor. The smaller is the α, the smoother is the $\hat{y}_t$ series. By repeated substitution, we can rewrite the recursion as

$$\hat{y}_t = \alpha \sum_{s=0}^{t-1} (1 - \alpha)^s y_{t-s} \tag{11.52}$$

This shows why this method is called exponential smoothing—the forecast of y_t is a weighted average of the past values of y_t, where the weights decline exponentially with time.

The forecasts from single smoothing are constant for all future observations. This constant is given by:

$$\hat{y}_{T+k} = \hat{y}_T \qquad \text{for all } k > 0 \tag{11.53}$$

where T is the end of the estimation sample.

To start the recursion, we need an initial value for $\hat{y}_t$ and a value for α. EViews uses the mean of the initial $(T+1)/2$ observations of y_t to start the recursion (where T is the number of observations in the sample). Bowerman and O'Connell (1979) suggest that values of α around 0.01 to 0.30 work quite well. You can also let EViews estimate α to minimize the sum of squares of one-step forecast errors.

Double Smoothing (one parameter)

This method applies the single smoothing method twice (using the same parameter) and is appropriate for series with a linear trend. Double smoothing of a series y is defined by the recursions:

$$\begin{aligned} S_t &= \alpha y_t + (1 - \alpha)S_{t-1} \\ D_t &= \alpha S_t + (1 - \alpha)D_{t-1} \end{aligned} \tag{11.54}$$

where S is the single smoothed series and D is the double smoothed series. Note that double smoothing is a single parameter smoothing method with damping factor $0 < \alpha \leq 1$.

Forecasts from double smoothing are computed as:

$$y_{T+k} = \left(2 + \frac{\alpha k}{1-\alpha}\right) S_T - \left(1 + \frac{\alpha k}{1-\alpha}\right) D_T$$

$$= \left(2S_T - D_T + \frac{\alpha}{1-\alpha}(S_T - D_T)k\right)$$

(11.55)

The last expression shows that forecasts from double smoothing lie on a linear trend with intercept $2S_T - D_T$ and slope $\alpha(S_T - D_T)/(1-\alpha)$.

Holt-Winters—Multiplicative (three parameters)

This method is appropriate for series with a linear time trend and multiplicative seasonal variation. The smoothed series y_t is given by,

$$y_{t+k} = (a + bk)c_{t+k} \tag{11.56}$$

where

$$\begin{array}{ll} a & \text{permanent component (intercept)} \\ b & \text{trend} \\ c_t & \text{multiplicative seasonal factor} \end{array} \tag{11.57}$$

These three coefficients are defined by the following recursions:

$$\begin{aligned} a(t) &= \alpha \frac{y_t}{c_t(t-s)} + (1-\alpha)(a(t-1) + b(t-1)) \\ b(t) &= \beta(a(t) - a(t-1)) + (1-\beta)b(t-1) \\ c_t(t) &= \gamma \frac{y_t}{a(t)} + (1-\gamma)c_t(t-s) \end{aligned} \tag{11.58}$$

where $0 < \alpha, \beta, \gamma < 1$ are the damping factors and s is the seasonal frequency specified in the **Cycle for Seasonal** field box.

Forecasts are computed by:

$$\hat{y}_{t+k} = (a(T) + b(T)k)c_{T+k-s} \tag{11.59}$$

where the seasonal factors are used from the last s estimates.

Holt-Winters—Additive (three parameter)

This method is appropriate for series with a linear time trend and additive seasonal variation. The smoothed series y_t is given by:

$$y_{t+k} = a + bk + c_{t+k} \tag{11.60}$$

where a and b are the permanent component and trend as defined above in Equation (11.57) and c are the additive seasonal factors. The three coefficients are defined by the following recursions:

$$
\begin{aligned}
a(t) &= \alpha(y_t - c_t(t-s)) + (1-\alpha)(a(t-1) + b(t-1)) \\
b(t) &= \beta(a(t) - a(t-1)) + 1 - \beta b(t-1) \\
c_t(t) &= \gamma(y_t - a(t+1)) - \gamma c_t(t-s)
\end{aligned}
\tag{11.61}
$$

where $0 < \alpha, \beta, \gamma < 1$ are the damping factors and s is the seasonal frequency specified in the **Cycle for Seasonal** field box.

Forecasts are computed by:

$$
\hat{y}_{T+k} = a(T) + b(T)k + c_{T+k-s}
\tag{11.62}
$$

where the seasonal factors are used from the last s estimates.

Holt-Winters—No Seasonal (two parameters)

This method is appropriate for series with a linear time trend and no seasonal variation. This method is similar to the double smoothing method in that both generate forecasts with a linear trend and no seasonal component. The double smoothing method is more parsimonious since it uses only one parameter, while this method is a two parameter method. The smoothed series $\hat{y}_t$ is given by:

$$
\hat{y}_{t+k} = a + bk
\tag{11.63}
$$

where a and b are the permanent component and trend as defined above in Equation (11.57).

These two coefficients are defined by the following recursions:;

$$
\begin{aligned}
a(t) &= \alpha y_t + (1-\alpha)(a(t-1) + b(t-1)) \\
b(t) &= \beta(a(t) - a(t-1)) + 1 - \beta b(t-1)
\end{aligned}
\tag{11.64}
$$

where $0 < \alpha, \beta, \gamma < 1$ are the damping factors. This is an exponential smoothing method with two parameters.

Forecasts are computed by:

$$
\hat{y}_{T+k} = a(T) + b(T)k
\tag{11.65}
$$

These forecasts lie on a linear trend with intercept $a(T)$ and slope $b(T)$.

It is worth noting that Holt-Winters—No Seasonal, is *not* the same as additive or multiplicative with $\gamma = 0$. The condition $\gamma = 0$ only restricts the seasonal factors from changing over time so there are still (fixed) nonzero seasonal factors in the forecasts.

Illustration

As an illustration of forecasting using exponential smoothing we forecast data on monthly housing starts (HS) for the period 1985m01–1988m12 using the DRI Basics data for the period 1959m01–1984m12. These data are provided in the workfile "Hs.WF1". Load the workfile, highlight the HS series, double click, select **Proc/Exponential Smoothing/Simple Exponential Smoothing…**. We use the **Holt-Winters - Multiplicative** method to account for seasonality, name the smoothed forecasts HSSM, and estimate all parameters over the period 1959m1–1984m12. Leave the remaining settings at their default values.

When you click **OK**, EViews displays the results of the smoothing procedure. The first part displays the estimated (or specified) parameter values, the sum of squared residuals, the root mean squared error of the forecast. The zero values for Beta and Gamma in this example mean that the trend and seasonal components are estimated as fixed and not changing.

```
Date: 08/18/09   Time: 17:03
Sample: 1959M01 1984M12
Included observations: 312
Method: Holt-Winters Multiplicative Seasonal
Original Series: HS
Forecast Series: HSSM
```

Parameters:	Alpha		0.7100
	Beta		0.0000
	Gamma		0.0000
Sum of Squared Residuals			40365.69
Root Mean Squared Error			11.37441

The second part of the table displays the mean (α), and trend (β) at the end of the estimation sample that are used for post-sample smoothed forecasts.

End of Period Levels:	Mean		134.6584
	Trend		0.064556
	Seasonals:	1984M01	0.680745
		1984M02	0.711559
		1984M03	0.992958
		1984M04	1.158501
		1984M05	1.210279
		1984M06	1.187010
		1984M07	1.127546
		1984M08	1.121792
		1984M09	1.050131
		1984M10	1.099288
		1984M11	0.918354
		1984M12	0.741837

For seasonal methods, the seasonal factors (γ) used in the forecasts are also displayed. The smoothed series in the workfile contains data from the beginning of the estimation sample to the end of the workfile range; all values after the estimation period are forecasts.

When we plot the actual values (HS) and the smoothed forecasts (HSSM) on a single graph, we get:
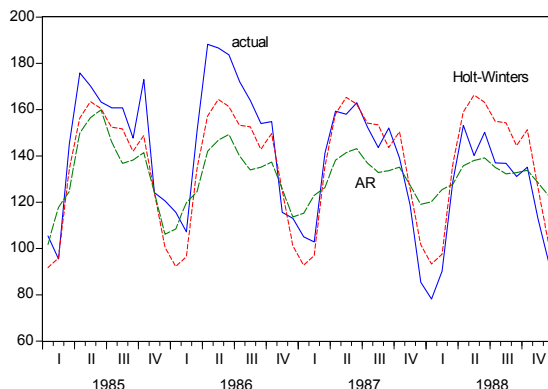
We have included the forecasted values HSF from estimation of an AR model with an AR(1) and a SAR(12) term for comparison purposes. The forecasts from the multiplicative exponential smoothing method do a better job of tracking the seasonal movements in the actual series.

ETS Exponential Smoothing

Although *ad hoc* exponential smoothing (ES) methods have been employed for many decades, recent methodological developments have embedded these models in a modern dynamic nonlinear model framework (see Chatfield, *et al.* 2001 for a brief historical overview).

Hyndman, Koehler, *et al.* (2002) outline the ETS (*E*rror-*T*rend-*S*easonal or *E*xponen*T*ial *S*moothing) framework which defines an extended class of ES methods and offers a theoretical foundation for analysis of these models using state-space based likelihood calculations, with support for model selection and calculation of forecast standard errors.

Notably, the ETS framework encompasses the standard ES models (*e.g.*, Holt and Holt–Winters additive and multiplicative methods), so that it provides a theoretical foundation for what was previously a collection of *ad hoc* approaches.

The remainder of this section outlines briefly the basics of ETS exponential smoothing models and describes their estimation in EViews. Users who desire additional detail should consult Ord, Koehler, and Snyder (1997), Hyndman, Koehler, Snyder, and Grose (2002), and in particular, the book-length treatment by Hyndman, Koehler, Ord, and Snyder (2008).

ETS Decomposition

The time series Y may be decomposed into three components, trend (T), seasonal (S), and error (E), where the trend term characterizes the long-term movement of Y, the seasonal term corresponds to a pattern with known periodicity, and the error term is the irregular, unpredictable component of the series.

These three components may be combined in various additive and multiplicative combinations to produce Y. We may have a purely additive model, say $Y = T + S + E$ or $Y = S + E$, a pure multiplicative model like $Y = T \cdot S \cdot E$ or $Y = T \cdot E$, or models which mix the two, as in $Y = (T \cdot S) + E$ or $Y = (T + S)(1 + E)$

The individual components of an ETS specification may be specified as being of the form:

$$
\begin{array}{ll}
E & \{\text{A, M}\} \\
T & \{\text{N, A, M, AD, MD}\} \\
S & \{\text{N, A, M}\}
\end{array}
\tag{11.66}
$$

where N = none, A = additive, M = multiplicative, AD = additive dampened, and MD = multiplicative dampened (damping uses an extra parameter to reduce the impact of the trend over time). The are a total of $30 = 2 \cdot 5 \cdot 3$ possible ETS models associated with these choices.

The trend specification requires a bit of additional discussion. The forecasted trend term T may itself be decomposed into a level term (l) and a growth term (b) in several ways. Following Hyndman, *et al.* (2008) we let T be the trend forecast h periods out, and let $0 < \phi < 1$ be a damping parameter. The five different trend types corresponding to different assumptions about the growth term are given by:

None	$T_h = l$
Additive	$T_h = l + bh$
Additive damped	$T_h = l + b\phi_h$
Multiplicative	$T_h = l \cdot b^h$
Multiplicative damped	$T_h = l \cdot b^{\phi_h}$

for $\phi_h \equiv \sum_{s=1}^{h} \phi^s$.

ETS Representation

We have seen that the trend is comprised of a level term (l) which is always present, and a growth term (b) which may or may be present, depending on the trend specification. If we

let s represent included seasonal terms, we may define the general following p-dimensional state vector:

$$x_t \equiv (l_t,\, b_t,\, s_t,\, s_{t-1},\, \ldots,\, s_{t-m})' \tag{11.67}$$

Following Ord, Koehler, and Snyder (1997), we may write a nonlinear dynamic model representation of the exponential smoothing equations using a state space model with a common error term:

$$\begin{aligned} y_t &= h(x_{t-1}, \theta) + k(x_{t-1}, \theta)e_t \\ x_t &= f(x_{t-1}, \theta) + g(x_{t-1}, \theta)e_t \end{aligned} \tag{11.68}$$

where the h and k are known continuous scalar functions, f and g are known continuously functions with continuous derivatives from $\Re^p \to \Re^p$ and $e_t \sim iid(0, \sigma^2)$ where e_t independent of past realizations of y and x.

Conceptually, the y_t equation represents how the various state variable components $(l_{t-1}, b_{t-1}, s_{t-m})$ are combined to express the series in terms of a smoothed forecast $\hat{y}_t = h(x_{t-1}, \theta)$ and the prediction error e_t. The multiple functions are a notational device for writing the additive and multiplicative errors in compact form. With additive errors we have $k \equiv 1$ so that

$$y_t = h(x_{t-1}, \theta) + e_t \tag{11.69}$$

With multiplicative errors we set $k \equiv h$, yielding

$$y_t = h(x_{t-1}, \theta)(1 + e_t) \tag{11.70}$$

The x_t equations outline the process by which the component estimates are updated using the previous period's estimates and the current prediction error e_t. For the ETS models considered here, we may think of the updating smoothing equations as being weighted averages of a term which depends on the current prediction error (and prior states), and one which depends on the prior states. The resulting state equations are extended versions of those outlined by Hyndman, *et al.* (2002), and take the general form:

$$\begin{aligned} l_t &= \alpha P(x_{t-1}, e_t) + (1 - \alpha)Q(x_{t-1}) \\ b_t &= \beta R(x_{t-1}, e_t) + (1 - \beta)\phi_1 b_{t-1}^{\phi_2} \\ s_t &= \gamma T(x_{t-1}, e_t) + (1 - \gamma)s_{t-m} \end{aligned} \tag{11.71}$$

where $P_t \equiv P(x_{t-1}, e_t)$, R_t and T_t are functions of the prediction error and lagged states, and $Q_t = Q(x_{t-1})$ is a function of the lagged states. ϕ_1 is the damping parameter for linear trend models and ϕ_2 is the damping parameter for multiplicative trend models. In the absence of a damping, the parameters are set to 1.

The exact forms of all of these equations depend on the specific ETS specification. The expressions corresponding to the 30 possible specifications are listed in Hyndman, *et al.*

(2008, Tables 2.2 and 2.3, p. 21–22). We do not reproduce the results here, but instead offer examples to show how familiar models fit into the framework.

Simple Exponential Smoothing (A, N, N)

To take the simplest example, consider the simple exponential smoothing (A,N,N) specification. For this model, the contemporaneous estimate of the level may be written as the weighted average of the current value of the variable and its forecasted value:

$$l_t = \alpha y_t + (1 - \alpha) \hat{y}_t \tag{11.72}$$

where the forecast is the previous value of the state, $\hat{y}_t = l_{t-1}$. For this linear error specification, obtain the prediction error equation

$$
\begin{aligned}
y_t &= \hat{y}_t + e_t \\
&= l_{t-1} + e_t
\end{aligned}
\tag{11.73}
$$

Substituting for y_t and $\hat{y}_t$ in Equation (11.72), we have an updating equation in error correction form:

$$
\begin{aligned}
l_t &= \alpha(l_{t-1} + e_t) + (1 - \alpha) l_{t-1} \\
&= l_{t-1} + \alpha e_t
\end{aligned}
\tag{11.74}
$$

The full model may be written in ETS form as (Hyndman, *et al.,* 2008, Table 2.2, p. 21):

$$
\begin{aligned}
y_t &= l_{t-1} + e_t \\
l_t &= l_{t-1} + \alpha e_t
\end{aligned}
\tag{11.75}
$$

This specification is an ETS model defined by Equation (11.69) and Equation (11.71) with state $x_t = l_t$, and functions $h(x_{t-1}, \theta) = l_{t-1}$, $P_t = l_{t-1} + e_t$, and $Q_t = l_{t-1}$.

Holt's Method with Multiplicative Errors (M, A, N)

Next, consider Holt's linear trend method with multiplicative errors (M, A, N). Define the state vector $x_t = (l_t, b_t)'$ and let $\hat{y}_t = l_{t-1} + b_{t-1}$ be the one-period ahead forecast of y_t since we have an additive trend specification with no seasonal component.

Given this forecast, the multiplicative error assumption implies that:

$$
\begin{aligned}
y_t &= \hat{y}_t(1 + e_t) \\
&= (l_{t-1} + b_{t-1})(1 + e_t)
\end{aligned}
\tag{11.76}
$$

The updating equation for the level state variable is obtained by forming a weighted average of the previous period component and a term which depends on the prediction error. We have

$$
\begin{aligned}
l_t &= \alpha y_t + (1 - \alpha)\hat{y}_t \\
&= \alpha(l_{t-1} + b_{t-1})(1 + e_t) + (1 - \alpha)(l_{t-1} + b_{t-1}) \\
&= (l_{t-1} + b_{t-1})(1 + \alpha e_t)
\end{aligned}
\tag{11.77}
$$

so that $P_t = (l_{t-1} + b_{t-1})(1 + e_t)$ and $Q_t = (l_{t-1} + b_{t-1})$.

Similarly, for the growth component, we have

$$
\begin{aligned}
b_t &= \beta(y_t - l_{t-1}) + (1 - \beta)b_{t-1} \\
&= \beta((l_{t-1} + b_{t-1})(1 + e_t) - l_{t-1}) + (1 - \beta)b_{t-1} \\
&= \beta(l_{t-1} + b_{t-1})e_t + b_{t-1}
\end{aligned}
\tag{11.78}
$$

so that $R_t = (l_{t-1} + b_{t-1})(1 + e_t) - l_{t-1}$ and $\phi_1 = \phi_2 = 1$.

The ETS representation for this model may be written as:

$$
\begin{aligned}
y_t &= (l_{t-1} + b_{t-1})(1 + e_t) \\
l_t &= (l_{t-1} + b_{t-1})(1 + \alpha e_t) \\
b_t &= b_{t-1} + \beta(l_{t-1} + b_{t-1})e_t
\end{aligned}
\tag{11.79}
$$

(Hyndman, et al., 2008, Table 2.3, p. 22.)

Holt-Winters Method with Multiplicative Errors and Seasonals (M, A, M)

Consider the Holt-Winters (M, A, M) linear trend model with multiplicative errors and multiplicative seasonal (m). Define the state vector $x_t \equiv (l_t, b_t, s_t, s_{t-1}, \ldots, s_{t-m})'$. The one-period ahead forecast is

$$
\hat{y}_t = (l_{t-1} + b_{t-1})s_{t-m}
\tag{11.80}
$$

and the prediction error decomposition is

$$
y_t = (l_{t-1} + b_{t-1})s_{t-m} \cdot (1 + e_t)
\tag{11.81}
$$

The level state equation is given by the weighted average of a term involving the prediction error and the previous state forecast:

$$
\begin{aligned}
l_t &= \alpha(y_t / s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \\
&= \alpha(l_{t-1} + b_{t-1})(1 + e_t) + (1 - \alpha)(l_{t-1} + b_{t-1}) \\
&= (l_{t-1} + b_{t-1})(1 + \alpha e_t)
\end{aligned}
\tag{11.82}
$$

Similarly, the growth equation is

$$
\begin{aligned}
b_t &= \beta(y_t / s_{t-m} - l_{t-1}) + (1 - \beta)b_{t-1} \\
&= \beta(l_{t-1} + b_{t-1})e_t + b_{t-1}
\end{aligned}
\tag{11.83}
$$

and the seasonal equation is

$$s_t = \gamma(y_t/(l_{t-1} + b_{t-1})) + (1 - \gamma)s_{t-m}$$
$$= \gamma s_{t-m}(1 + e_t) + (1 - \gamma)s_{t-m} \tag{11.84}$$
$$= s_{t-m}(1 + \gamma e_t)$$

The ETS representation for the (M, A, M) model is then:

$$y_t = (l_{t-1} + b_{t-1})s_{t-m} \cdot (1 + e_t)$$
$$l_t = (l_{t-1} + b_{t-1})(1 + \alpha e_t)$$
$$b_t = \beta(l_{t-1} + b_{t-1})e_t + b_{t-1} \tag{11.85}$$
$$s_t = s_{t-m}(1 + \gamma e_t)$$

(Hyndman, *et al.* 2008, Table 2.3, p. 22).

Fully Multiplicative Method with Damping (M, Md, M)

Consider an damped exponential trend model with multiplicative seasonal and error components (M, Md, M). Define the state vector $x_t \equiv (l_t, b_t, s_t, s_{t-1}, \ldots, s_{t-m})'$ and let $\hat{y}_t = l_{t-1}b_{t-1}^\phi s_{t-m}$ be the one-period ahead forecast of y_t. Then the prediction error decomposition is

$$y_t = l_{t-1}b_{t-1}^\phi s_{t-m} \cdot (1 + e_t) \tag{11.86}$$

The level state equation is given by:

$$l_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)(l_{t-1}b_{t-1}^\phi)$$
$$= \alpha(l_{t-1}b_{t-1}^\phi)(1 + e_t) + (1 - \alpha)(l_{t-1}b_{t-1}^\phi) \tag{11.87}$$
$$= (l_{t-1}b_{t-1}^\phi)(1 + \alpha e_t)$$

The growth equation is

$$b_t = \beta(y_t/l_{t-1}s_{t-m}) + (1 - \beta)b_{t-1}^\phi$$
$$= \beta(b_{t-1}^\phi)(1 + e_t) + (1 - \beta)b_{t-1}^\phi \tag{11.88}$$
$$= b_{t-1}^\phi(1 + \beta e_t)$$

Notice that this state equation is of the form in Equation (11.71) with $\phi_1 = 1$ (since we have a multiplicative damped trend) and $R_t = b_{t-1}^\phi(1 + e_t)$.

The seasonal equation is

$$s_t = \gamma(y_t/(l_{t-1}b_{t-1}^\phi)) + (1 - \gamma)s_{t-m}$$
$$= \gamma s_{t-m}(1 + e_t) + (1 - \gamma)s_{t-m} \tag{11.89}$$
$$= s_{t-m}(1 + \gamma e_t)$$

$$\tag{11.90}$$

Then the ETS representation for the (M, Md, M) model is:

$$y_t = l_{t-1} b_{t-1}^{\phi} s_{t-m} \cdot (1 + e_t)$$

$$l_t = (l_{t-1} b_{t-1}^{\phi})(1 + \alpha e_t)$$

$$b_t = b_{t-1}^{\phi}(1 + \beta e_t) \tag{11.91}$$

$$s_t = s_{t-m}(1 + \gamma e_t)$$

(Hyndman, *et al.* 2008, Table 2.3, p. 22).

Parameters and Initial States

Given any ETS specification, parameters $\theta^* = (\alpha, \beta, \gamma, \phi)$, and values for the initial states $x_0^* = (l_0, b_0, s_0, s_{-1}, \ldots, s_{-m+1})$, we may use the ETS state and prediction equations to obtain smoothed estimates of the unobserved components and the underlying series. Typically the parameters are estimated. The initial states are either estimated or assigned *ad hoc* values.

Some ETS models require only a subset of the parameters and states. Models without a seasonal component, for example, use neither the seasonal parameter γ, nor the initial seasonal states $(s_0, s_{-1}, \ldots, s_{-m+1})$. We will use θ and x_0 to refer to the relevant subsets for a given model which are to be estimated.

Parameters

The parameter values are usually unknown and must be estimated. There have been many suggestions for reducing the size of the potential parameter space for θ. The traditional approach has assigned the following restrictions on the parameters to ensure model stability:

$$0 \leq \alpha \leq 1$$

$$0 \leq \beta \leq \alpha$$

$$0 \leq \gamma \leq (1 - \alpha) \tag{11.92}$$

$$0 \leq \phi \leq 1$$

Initial states

Traditionally, the initial state values are specified using *ad hoc* values. The ETS approach allows for either the use of these experimental values, or estimation of the initial states as part of the optimization problem. Even when the initial states are estimated, it is useful to use *ad hoc* values as a starting point for optimization.

Following Hyndman, *et al. (*2002) EViews uses the following *ad hoc* method for computing the initial states or starting values for the estimated initial states:

- *Initial level components*: For nonseasonal data, compute a linear trend on the first ten observations against a time series $t = 1, \ldots, 10$. For seasonal data, compute a linear trend using linear regression on the first ten seasonally adjusted values against a time series. And then set the initial level l_0 to be the intercept of the trend.

- *Initial growth components*: For additive trends, set b_0 to be the slope of the trend. For multiplicative trends, set $b_0 = 1 + b/a$ where a and b denote the intercept and slope of the fitted trend, respectively.

- *Initial seasonal component*: Compute a $2 \times m$ moving average through the first few years of data. Denote this set of values $\{f_t\}$, $t = m/2 + 1, m/2 + 2, \ldots$. For additive seasonality, detrend the data to obtain $y_t - f_t$. For multiplicative seasonality, detrend the data to obtain y_t/f_t. Compute initial seasonal indices $s_{-m+1}, \ldots, s_0$ by averaging the detrended data for each season. Normalize these seasonal indices so that they add to zero for additive seasonality, and add to m for multiplicative seasonality.

Estimation

EViews offers both maximum likelihood (ML) and average mean square error minimization (AMSE) methods of estimating the unknown parameters θ and, optionally, the initial states x_0.

Maximum Likelihood Estimation

The maximum likelihood estimator finds the parameters and initial states that maximize the log-likelihood function. The Gaussian log-likelihood for ETS specifications can be written in terms of the prediction errors as

$$\log L(\theta, x_0, \sigma^2 | y) = -\frac{2}{T}\log(2\pi\sigma^2) - \frac{1}{2}\sum_{t=1}^{n}\frac{e_t^2}{\sigma^2} \tag{11.93}$$

For a given set of parameters and initial states and ETS model, the likelihood is evaluated recursively using the state equations and solving for the prediction error using the appropriate prediction decomposition equation.

The parameters and initial conditions are obtained by maximizing the likelihood in Equation (11.93) with respect to θ, x_0, and σ^2 using the Broyden, Fletcher, Goldfarb and Shanno (BFGS) algorithm.

Average Mean Square Error Minimization

The average mean square error estimator finds the parameter values and initial state values that minimize the average mean square error of the h-step forecasts of the specified ETS model.

Recall that our prediction error is simply the one-step ahead forecast error for the series: $e_t = y_t - \hat{y}_t$. The h-step ahead forecast error is defined as

$$e_{t+h|t} = y_{t+h} - \hat{y}_{t+h|t} \tag{11.94}$$

where the components of $\hat{y}_{t+h|t}$ are obtained by recursively solving the state equations assuming $e_{t+h} = 0$ in each period.

The average mean square error of the h-step forecasts is defined as

$$AMSE = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{h} \sum_{k=1}^{h} e_{t+k|t}^2 \right) \tag{11.95}$$

The AMSE is then minimized with respect to the θ and x_0 using BFGS.

For example, for the (M, A, N) specification described earlier (), and $h = 3$ we have

$$\begin{aligned}
b_{t+1} &= b_t \\
b_{t+2} &= b_{t+1} \\
b_{t+3} &= b_{t+2}
\end{aligned} \tag{11.96}$$

so that $b_{t+h} = b_t$, and

$$\begin{aligned}
l_{t+1} &= l_t + b_t \\
l_{t+2} &= l_{t+1} + b_{t+1} \\
l_{t+3} &= l_{t+2} + b_{t+2}
\end{aligned} \tag{11.97}$$

so that $l_{t+h} = l_t + hb_t$. Then the h-step ahead forecasts are

$$\begin{aligned}
\hat{y}_{t+1|t} &= l_t + b_t \\
\hat{y}_{t+2|t} &= l_t + 2b_t \\
\hat{y}_{t+3|t} &= l_t + 3b_t
\end{aligned} \tag{11.98}$$

The AMSE is then given by

$$AMSE = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{h} \sum_{k=1}^{h} (y_{t+k} - (l_t + kb_t)) \right) \tag{11.99}$$

Model Selection

The ETS smoothing framework lends itself to automatic model selection. Since both a likelihood and a forecast error can be calculated for each ETS model it is possible to compare a likelihood based information criterion or out-of-sample AMSE across models to determine which one more closely fits the data (using the likelihood), or forecasts more accurately (using AMSE).

Likelihood based comparisons can be performed using the standard likelihood based criteria: Akaike Information Criterion (AIC), Schwarz Information Criterion (BIC), or the Hannan-Quinn Criterion (HQ). Specifically,

$$
\begin{aligned}
AIC &= -2\log L(\boldsymbol{\theta}, \hat{x}_0) + 2p \\
BIC &= -2\log L(\boldsymbol{\theta}, \hat{x}_0) + \log(T)p \\
HQ &= -2\log L(\boldsymbol{\theta}, \hat{x}_0) + 2\log(\log T)p
\end{aligned}
\tag{11.100}
$$

where $(\hat{\boldsymbol{\theta}}, \hat{x}_0)$ are the maximized values and p is the number of parameters in $\hat{\boldsymbol{\theta}}$ plus the number of the estimated initial states in $\hat{x}_0$. The model that minimizes the AIC (BIC, or HQ) across all available models is adopted.

The out-of-sample average mean squared error (AMSE) is calculated as before:

$$
AMSE = \frac{1}{T^*} \sum_{t=T+1}^{T+T^*} \left(\frac{1}{h} \sum_{k=1}^{h} e_{t+h|t}^2 \right)
\tag{11.101}
$$

but using out-of-sample data that were not used for model estimation.

Note that two complications can arise when performing model selection:

- Division by zero: some multiplicative models can run into division by zero issues during estimation. One way to alleviate this problem is to only compare models with additive trend and seasonal components.

- Failure to converge: some models may fail to find an optimal value for the parameters, or fail to converge. These models may report likelihood and forecast errors values even though they were not truly optimized.

Smoothing/Forecasting

Once an ETS model has been chosen and the parameters have been estimated, the in-sample smoothed series can be calculated by recursively substituting the estimated parameters into the one-step ahead forecast function $h(x_{t-1}, \theta)$ and using the prediction errors to update the state estimates.

The smoothing procedure can be extended beyond the range of the original data to form an out-of-sample smoothing forecast of the series. The out-of-sample smoothed values are dynamic forecasts of the series using the smoothing model, estimated parameters, and in-sample data.

Using ETS Smoothing in EViews

To perform exponential smoothing in EViews, open the series and choose **Proc/Exponential Smoothing/ETS Exponential Smoothing…** to bring up the **ETS Smoothing** dialog:
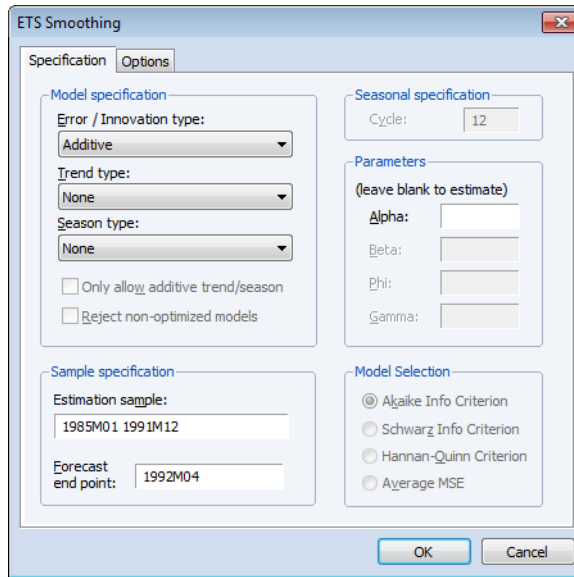
ETS specification

You may use the **Model specification** section of the dialog to specify the type of ETS model you wish to use for smoothing. There is one dropdown menu for each part of the decomposition. By default, EViews employs the additive error, no trend, and no seasonal (A, N, N) model.

In addition to the component choices outlined in "ETS Decomposition" on page 517 (N, A, AD, *etc.*), EViews offers you an **Auto** choice so you may specify that the component specification will be chosen using model selection techniques.

For the calculational efficiency, EViews provides two options to reduce the number of available ETS smoothing models. The **Only allow additive trend/season only** checkbox lets you choose from models with additive trend and seasonal components only. The **Reject non-optimized models** checkbox lets you exclude models for which the estimation procedure did not converge from model comparison.

Seasonal specification

If you have specified a seasonal component or elected to select the seasonal specification using model selection, the **Seasonal specifica-**
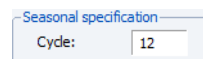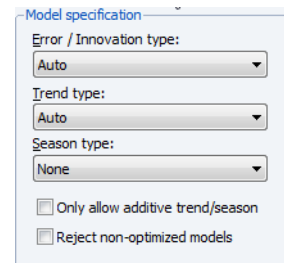
tion edit field lets you specify the m length of seasonality. The default cycle setting will depend on the workfile structure (4 for quarterly, 12 for monthly, 2 for all others).

Model selection

When you choose **Auto** for any of the component types, EViews displays **Model Selection** options which let you specify the model selection procedure. You may choose between minimizing one of the information criteria (**Akaike Info Criterion**, **Schwarz Info Criterion**, **Hannan-Quinn Criterion**) or finding the model with the best within sample one-step predictive forecast by minimizing the **Average MSE**.

Parameters

The **Parameters** section of the dialog lets you specify fixed values for the smoothing parameters in your model. If you enter a numeric value in one of the edit fields, EViews will fix the parameter at that value and remove it from the set of parameters to be estimated. Leaving a field blank instructs EViews to estimate the parameter.

Note that any user-specified parameter values should fall within the parameter bounds outlined in Equation (11.92).

Estimation Sample

The **Sample specification** section consists of two settings: the estimation sample, and the forecast end point.

The **Estimation sample** edit field identifies the contiguous sample that will be used for estimation of the smoothing model. You should enter a single date pair. Note that the ETS procedure does not allow for missing values in the source series.

The **Forecast end point** lets you specify the end point of your forecast sample. Note that the forecast start point is fixed as the first observation after the estimation sample. For example, if the estimation sample is given as "1985m01 1991m12" and the forecast end point is given as "1992m04", then EViews will perform exponential smoothing analysis for the estimation sample and forecast the remaining periods (*i.e.* "1992m01 1992m04").

Estimation Control

To specify the options for controlling estimation, click on the **Option** tab.

The **Model optimization** section allows you to specify the optimization objective function, maximum iterations and convergence, and starting values, and to indicate whether you wish

to treat the initial state values as parameters or to use the *ad hoc* method for computing the initial states.

- **Objective**: you can choose the objective function which needs to be optimized over, either likelihood based (**Log-likelihood**) or forecast error based (**Average MSE**).

- **AMSE Length**: If the average MSE objective function is chosen as the object, the AMSE length h described in Equation (11.95) should be specified in the edit field.

- **Max Iterations** and **Convergence**: Lets you control the iterative process by specifying the maximum number of iterations and convergence criterion.

- By default, starting parameters are set to EViews supplied values, but you may select the **User starting values** checkbox to use the existing values in the C coefficient vector.

- Select the **Do not optimize initial states** box to use the *ad hoc* method for computing the initial states.

Display

The **Display** section provides options for showing the results of the ETS procedure in graph or table form.

The top portion of the section lets you specify the display of the component results in single or multiple graphs. If you select **Multiple graphs**, EViews will display the selected **Forecast**, **Level**, **Trend**, and **Season** components in separate graphs. If you select **Single graph**, all of the selected components will be included in a single graph.
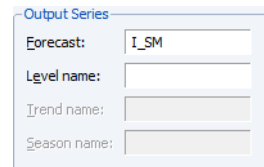
If the ETS specification in the **Specification** page of the dialog ("ETS specification" on page 527) contains any automatically chosen components, EViews will offer you the option of displaying a graph or table of the forecast and likelihood values used to forecast and/or likelihood values used in the model selection procedure. Simply select the appropriate Graph or Table checkbox to include a component. The results will be displayed along with the estimation results.

Output Series

The **Output Series** section of the **Options** tab lets you specify the smoothing process output series names. By default EViews will create a new series in the workfile with the final smoothed (and possibly forecasted) series. The name of the created series will be the same as the original series name, with an "_SM" appended to the end. You can change this name using the **Forecast name** edit field.
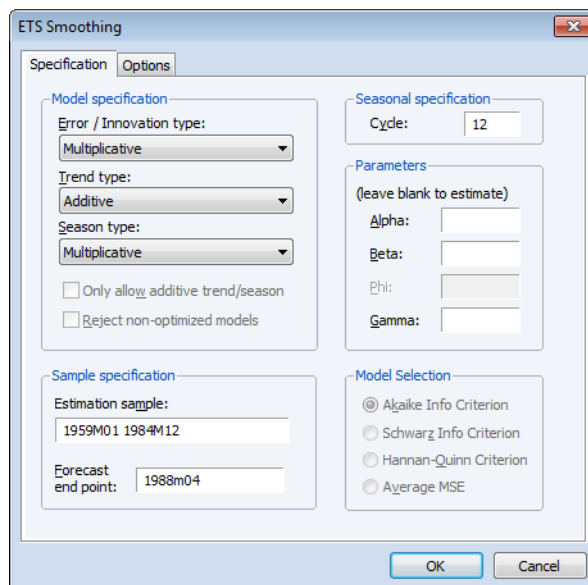
You may elect to save the individual level, trend and seasonal components of the smoothed series output to your workfile by entering a name for those output series in the **Level name**, **Trend name**, and **Season name** fields.

Illustration

To illustrate estimation and smoothing using an ETS model, we forecast monthly housing starts (HS) for the period 1985m01–1988m12. These data are provided in the workfile "Hs.WF1".

We will use the multiplicative error, additive trend, and multiplicative seasonal (M, A, M) model to estimate parameters using data from 1959m01–1984m12 and to smooth and forecast for 1985m1–1988m12.

First, load the workfile, open the HS series, and select **Proc/Exponential Smoothing/ETS Exponential Smoothing**....

Change the **Model specification** drop-down menus to (M, A, M), set the **Estimation sample** to "1959 1984" or "1959m01 1984m12", set the **Forecast end poin**t to "1988m04", and leave the remaining settings at their default values. When you click **OK**, EViews estimates the ETS model, displays the results, and saves the smoothed results in the HS_SM series in the work-file.

The results are divided into four parts. The first part of the table shows the settings employed in the ETS procedure, including the sample employed for estimation and the esti-mation status.

```
ETS Smoothing
Original series: HS
Date: 12/05/12   Time: 17:30
Sample: 1959M01 1984M12
Included observations: 312
Model: M,A,M - Multiplicative Error, Additive
        Trend, Multiplicative Season
Convergence achieved on boundaries.
```

Here we see that we have estimated an (M, A, M) model using data from 1959 to 1984, and that the estimator converged, but with some parameters at boundary values.

The next section of the table shows the smoothing parameters (α, β, γ) and initial states $x_0 = (l_0, b_0, s_0, s_{-1}, \ldots, s_{-11})$. Note the presence of the boundary zero values for β and γ, which indicate that the seasonal and trend components do not change from their initial values.

Parameters	
Alpha:	0.731105
Beta:	0.000000
Gamma:	0.000000
Initial Parameters	
Initial level:	139.0989
Initial trend:	0.540034
Initial state 1:	0.734455
Initial state 2:	0.910607
Initial state 3:	1.092262
Initial state 4:	1.050499
Initial state 5:	1.122191
Initial state 6:	1.132674
Initial state 7:	1.191977
Initial state 8:	1.217559
Initial state 9:	1.161483
Initial state 10:	1.003810
Initial state 11:	0.708928
Initial state 12:	0.673557

The bottom portion of the table output contains summary statistics for the estimation procedure:

Compact Log-likelihood	-1651.782
Log-likelihood	-1198.583
Akaike Information Criterion	3335.565
Schwarz Criterion	3395.453
Hannan-Quinn Criterion	3359.500
Sum of Squared Residuals	2.595320
Root Mean Squared Error	0.091205
Average Mean Squared Error	202.6130

Most of these statistics are self-explanatory. The reported "Compact Log-likelihood" is simply the log-likelihood value absent inessential constants, and is provided to facilitate comparison with results obtained from other sources.

For comparison purposes, it may be useful to consider the ETS model obtained using model selection. To perform model selection, fill out the dialog as before, but set each of the **Model** specification drop-down menus to **Auto**.

Note that at the default settings, the best model will be selected using the Akaike Information Criterion.
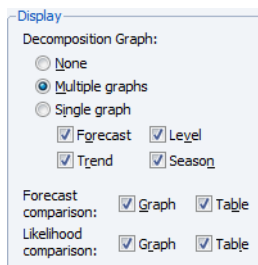
Next, click on the **Options** tab and set the **Display** options to show the forecast and all of the elements of the decomposition in Multiple graphs, and to produce graphs and tables for the forecast and likelihood comparisons of all of the models considered by the model selection procedure.

Click on **OK** to perform the smoothing. Since EViews will be producing several types of output for the procedure, the results will be displayed in a spool:
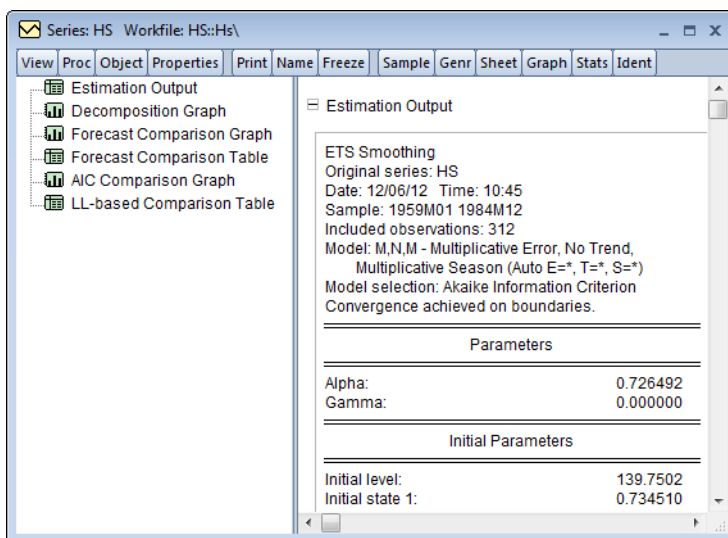
The left output pane allows you to select the output you wish to display. Simply click on the output you wish to display or use the scroll bar on the right side of the window to move from output to output.

The **Estimation Output** contains the specification, estimated smoothing and initial parameters, and summary statistics. The top portion of the output,

ETS Smoothing
Original series: HS
Date: 12/06/12   Time: 11:14
Sample: 1959M01 1984M12
Included observations: 312
Model: M,N,M - Multiplicative Error, No Trend,
     Multiplicative Season (Auto E=*, T=*, S=*)
Model selection: Akaike Information Criterion
Convergence achieved on boundaries.

Parameters	
Alpha:	0.726492
Gamma:	0.000000

shows that the Akaike information criterion selected ETS model is an (M, N, M) specification, with level smoothing parameter estimate $\hat{\alpha} = 0.72$, and the seasonal parameter $\hat{\gamma} = 0$ estimated on the boundary.

The summary statistics indicate that this specification is superior to the earlier (M, A, M) model,

Compact Log-likelihood	-1652.500
Log-likelihood	-1199.301
Akaike Information Criterion	3333.001
Schwarz Criterion	3385.403
Hannan-Quinn Criterion	3353.944
Sum of Squared Residuals	2.638713
Root Mean Squared Error	0.091964
Average Mean Squared Error	200.6665

on the basis of all three of the information criteria and the average mean squared error, though the likelihood is lower and the SSR and RMSE are both slightly higher in the selected model.

Clicking on the AIC comparison graph in the spool, we see the results for all candidate models:
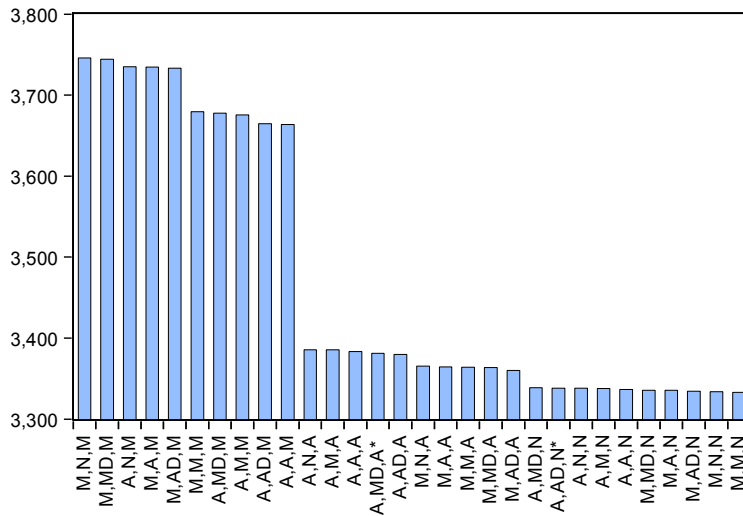
Note that the selected (M, N, M) and the original (M, A, M) model are among the five specifications with relatively high AIC values.

The forecast comparison graph shows the forecasts for the candidate models:
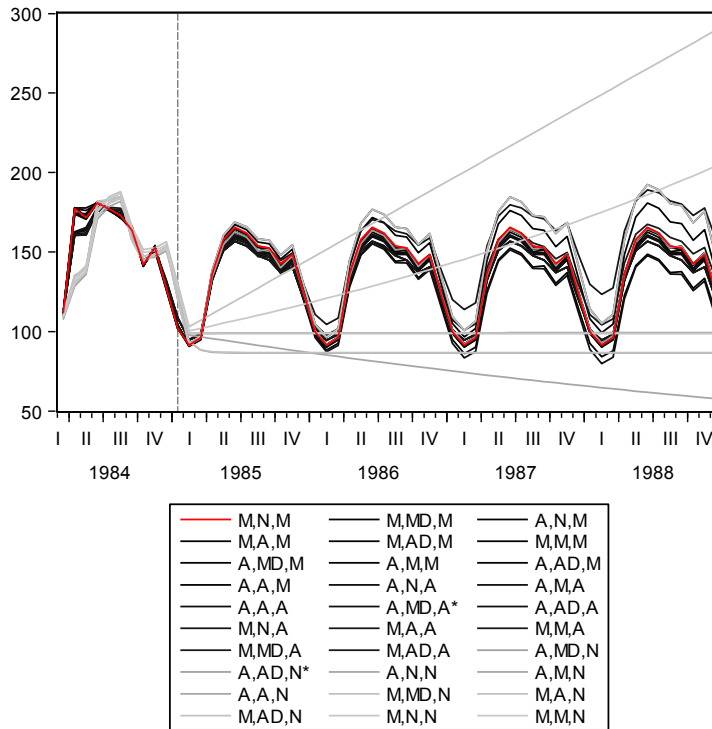
The graph shows both the last few observations of in-sample forecasts and the out-of-sample forecasts for each of the possible ETS specifications.

In addition, our chosen ETS display settings produced both the likelihood table which contains the actual likelihood and Akaike values for each specification, and the forecast comparison table, which presents a subset of the values displayed in the graph. For example, the likelihood table consists of

Model	Compact LL	Likelihood	AIC*	BIC	HQ	AMSE
M,N,M	-1652.50	-1199.30	3333.00	3385.40	3353.94	200.666
M,MD,M	-1650.02	-1196.82	3334.03	3397.66	3359.46	202.538
A,N,M	-1653.26	-1200.06	3334.52	3386.92	3355.47	198.379
M,A,M	-1651.78	-1198.58	3335.56	3395.45	3359.50	202.613
M,AD,M	-1650.80	-1197.60	3335.60	3399.23	3361.03	NA
M,M,M	-1652.34	-1199.14	3336.67	3396.56	3360.61	202.400
A,MD,M	-1651.93	-1198.74	3337.87	3401.50	3363.30	195.904
A,M,M	-1653.10	-1199.90	3338.20	3398.09	3362.14	198.013
A,AD,M	-1652.13	-1198.93	3338.25	3401.89	3363.69	NA
A,A,M	-1653.26	-1200.06	3338.51	3398.40	3362.45	198.348
A,N,A	-1665.98	-1212.78	3359.96	3412.36	3380.90	220.707
A,M,A	-1665.78	-1212.58	3363.56	3423.45	3387.50	220.163
A,A,A	-1665.98	-1212.78	3363.96	3423.85	3387.90	220.709
A,MD,A*	-1665.34	-1212.14	3364.68	3428.32	3390.12	219.225
A,AD,A	-1665.67	-1212.47	3365.34	3428.97	3390.77	NA
M,N,A	-1675.82	-1222.62	3379.65	3432.05	3400.59	231.326
M,A,A	-1674.66	-1221.46	3381.33	3441.21	3405.26	232.101
M,M,A	-1675.73	-1222.53	3383.45	3443.34	3407.39	231.930
M,MD,A	-1675.69	-1222.49	3385.38	3449.01	3410.81	230.702
M,AD,A	-1675.82	-1222.62	3385.65	3449.28	3411.08	NA
A,MD,N	-1826.80	-1373.60	3663.61	3682.32	3671.09	1036.08
A,AD,N*	-1827.39	-1374.19	3664.78	3683.50	3672.26	NA
A,N,N	-1835.67	-1382.47	3675.34	3682.83	3678.33	1018.94
A,M,N	-1834.78	-1381.58	3677.56	3692.53	3683.54	1002.86
A,A,N	-1835.67	-1382.47	3679.34	3694.31	3685.32	1018.93
M,MD,N	-1861.57	-1408.37	3733.14	3751.85	3740.62	1035.91
M,A,N	-1863.25	-1410.05	3734.51	3749.48	3740.49	1092.82
M,AD,N	-1862.53	-1409.33	3735.06	3753.78	3742.54	NA
M,N,N	-1870.03	-1416.83	3744.06	3751.55	3747.05	1019.19
M,M,N	-1868.93	-1415.73	3745.86	3760.83	3751.85	1077.12

*2 models failed to converge

Lastly, the spool contains a multiple graph containing the actual and forecasted values of HS over the estimation and forecast period, along with the decomposition of the series into the level and seasonal components.
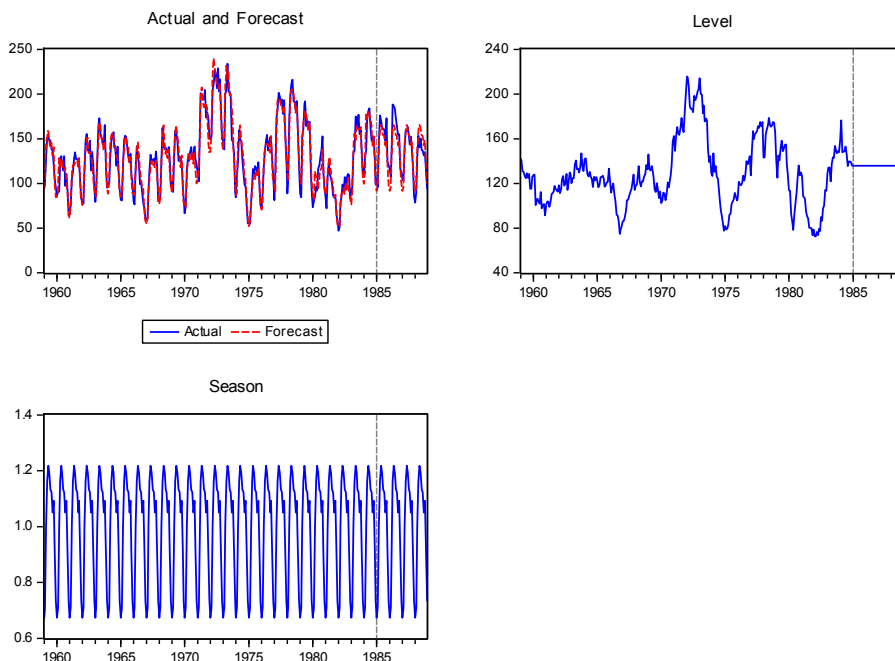
Hodrick-Prescott Filter

The Hodrick-Prescott Filter is a smoothing method that is widely used among macroeconomists to obtain a smooth estimate of the long-term trend component of a series. The method was first used in a working paper (circulated in the early 1980's and published in 1997) by Hodrick and Prescott to analyze postwar U.S. business cycles.

Technically, the Hodrick-Prescott (HP) filter is a two-sided linear filter that computes the smoothed series s of y by minimizing the variance of y around s, subject to a penalty that constrains the second difference of s. That is, the HP filter chooses s to minimize:

$$\sum_{t=1}^{T} (y_t - s_t)^2 + \lambda \sum_{t=2}^{T-1} ((s_{t+1} - s_t) - (s_t - s_{t-1}))^2 . \tag{11.102}$$

The penalty parameter λ controls the smoothness of the series σ. The larger the λ, the smoother the σ. As $\lambda = \infty$, s approaches a linear trend.

To smooth the series using the Hodrick-Prescott filter, choose **Proc/Hodrick-Prescott Filter…**:
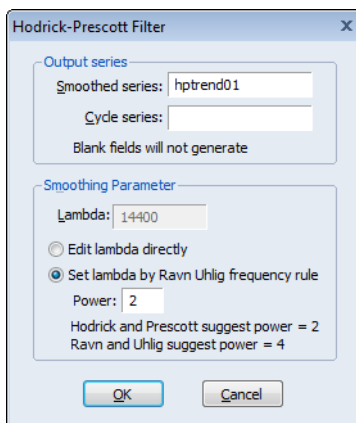
First, provide a name for the smoothed series. EViews will suggest a name, but you can always enter a name of your choosing. Next, specify an integer value for the smoothing parameter, λ. You may specify the parameter using the frequency power rule of Ravn and Uhlig (2002) (the number of periods per year divided by 4, raised to a power, and multiplied by 1600), or you may specify λ directly. The default is to use a power rule of 2, yielding the original Hodrick and Prescott values for λ:

$$
\lambda = \left\{
\begin{array}{ll}
100 & \text{for annual data} \\
1,600 & \text{for quarterly data} \\
14,400 & \text{for monthly data}
\end{array}
\right.
\tag{11.103}
$$

Ravn and Uhlig recommend using a power value of 4. EViews will round any non-integer values that you enter. When you click **OK**, EViews displays a graph of the filtered series together with the original series. Note that only data in the current workfile sample are filtered. Data for the smoothed series outside the current sample are filled with NAs.

Frequency (Band-Pass) Filter

EViews computes several forms of band-pass (frequency) filters. These filters are used to isolate the cyclical component of a time series by specifying a range for its duration. Roughly speaking, the band-pass filter is a linear filter that takes a two-sided weighted moving average of the data where cycles in a "band", given by a specified lower and upper bound, are "passed" through, or extracted, and the remaining cycles are "filtered" out.

To employ a band-pass filter, the user must first choose the range of durations (*periodicities*) to pass through. The range is described by a pair of numbers (P_L, P_U), specified in units of the workfile frequency. Suppose, for example, that you believe that the business cycle lasts somewhere from 1.5 to 8 years so that you wish to extract the cycles in this range. If you are

working with quarterly data, this range corresponds to a low duration of 6, and an upper duration of 32 quarters. Thus, you should set $P_L = 6$ and $P_U = 32$.

In some contexts, it will be useful to think in terms of frequencies which describe the number of cycles in a given period (obviously, periodicities and frequencies are inversely related). By convention, we will say that periodicities in the range (P_L, P_U) correspond to frequencies in the range $(2\pi/P_U, 2\pi/P_L)$.

Note that since saying that we have a cycle with a period of 1 is meaningless, we require that $2 \leq P_L < P_U$. Setting P_L to the lower-bound value of 2 yields a *high-pass filter* in which all frequencies above $2\pi/P_U$ are passed through.

The various band-pass filters differ in the way that they compute the moving average:

- The fixed length symmetric filters employ a fixed lead/lag length. Here, the user must specify the fixed number of lead and lag terms to be used when computing the weighted moving average. The symmetric filters are time-invariant since the moving average weights depend only on the specified frequency band, and do not use the data. EViews computes two variants of this filter, the first due to Baxter-King (1999) (BK), and the second to Christiano-Fitzgerald (2003) (CF). The two forms differ in the choice of objective function used to select the moving average weights.

- Full sample asymmetric – this is the most general filter, where the weights on the leads and lags are allowed to differ. The asymmetric filter is time-varying with the weights both depending on the data and changing for each observation. EViews computes the Christiano-Fitzgerald (CF) form of this filter.

In choosing between the two methods, bear in mind that the fixed length filters require that we use same number of lead and lag terms for every weighted moving average. Thus, a filtered series computed using q leads and lags observations will lose q observations from both the beginning and end of the original sample. In contrast, the asymmetric filtered series do not have this requirement and can be computed to the ends of the original sample.

Computing a Band-Pass Filter in EViews

The band-pass filter is available as a series Proc in EViews. To display the band-pass filter dialog, select **Proc/Frequency Filter...** from the main series menu.

The first thing you will do is to select a filter type. There are three types: **Fixed length symmetric (Baxter-King)**, **Fixed length symmetric (Christiano-Fitzgerald)**, or **Full length asymmetric (Christiano-Fitzgerald)**. By default, the EViews will compute the Baxter-King fixed length symmetric filter.

For the Baxter-King filter, there are only a few options that require your attention. First, you must select a frequency length (lead/lags) for the moving average, and the low and high values for the cycle period (P_L, P_U) to be filtered. By default, these fields will be filled in with reasonable default values that are based on the type of your workfile.

Lastly, you may enter the names of objects to contain saved output for the cyclical and non-cyclical components. The **Cycle series** will be a series object containing the filtered series (cyclical component), while the **Non-cyclical series** is simply the difference between the actual and the filtered series. The user may also retrieve the moving average weights used in the filter. These weights, which will be placed in a matrix object, may be used to plot customized frequency response functions. Details are provided below in "The Weight Matrix" on page 542.

Both of the CF filters (symmetric and asymmetric) provide you with additional options for handling trending data.

The first setting involves the **Stationarity assumption**. For both of the CF, you will need to specify whether the series is assumed to be an I(0) covariance stationary process or an I(1) unit root process.

Lastly, you will select a **Detrending method** using the dropdown. For a covariance stationary series, you may choose to demean or detrend the data prior to applying the filters. Alternatively, for a unit root process, you may choose to demean, detrend, or remove drift using the adjustment suggested by Christiano and Fitzgerald (2003).

Note that, as the name suggests, the full sample filter uses all of the observations in the sample, so that the **Lead/Lags** option is not relevant. Similarly, detrending the data is not an option when using the BK fixed length symmetric filter. The BK filter removes up to two unit roots (a quadratic deterministic trend) in the data so that detrending has no effect on the filtered series.

The Filter Output

Here, we depict the output from the Baxter-King filter using the series LOG(GDP) in the workfile "Bpf.WF1". The results were produced by selecting **Proc/Frequency Filter...** from the main series menu and clicking on **OK** to accept the default settings. (We have added grid lines to the graph to make it easier to see the cyclical behavior). The left panel depicts the original series, filtered series, and the non-cyclical component (difference between the original and the filtered).
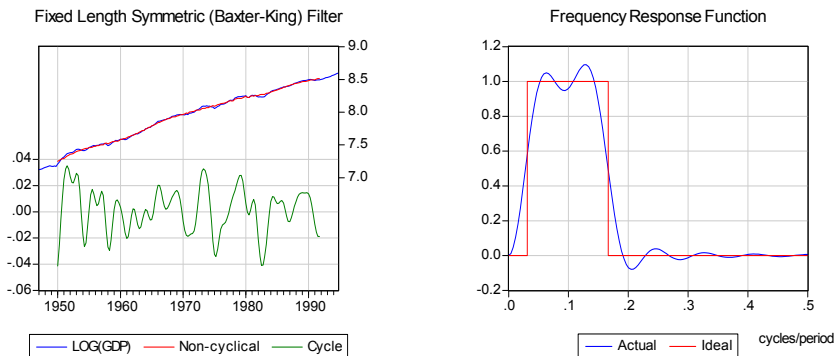
For the BK and CF fixed length symmetric filters, EViews plots the frequency response function $\alpha(\omega)$ representing the extent to which the filtered series "responds" to the original series at frequency ω. At a given frequency ω, $|\alpha(\omega)|^2$ indicates the extent to which a moving average raises or lowers the variance of the filtered series relative to that of the original series. The right panel of the graph depicts the function. Note that the horizontal axis of a frequency response function is always in the range 0 to 0.5, in units of cycles per duration. Thus, as depicted in the graph, the frequency response function of the ideal band-pass filter for periodicities (P_L, P_U) will be one in the range $(1/P_U, 1/P_L)$.

The frequency response function is not drawn for the CF time-varying filter since these filters vary with the data and observation number. If you wish to plot the frequency response function for a particular observation, you will have to save the weight matrix and then evaluate the frequency response in a separate step. The example program BFP02.PRG and subroutine FREQRESP.PRG illustrate the steps in computing of gain functions for time-varying filters at particular observations.

The Weight Matrix

For time-invariant (fixed-length symmetric) filters, the weight matrix is of dimension $1 \times (q + 1)$ where q is the user-specified lag length order. For these filters, the weights on the leads and the lags are the same, so the returned matrix contains only the one-sided weights. The filtered series can be computed as:

$$z_t = \sum_{c=1}^{q+1} w(1, c) y_{t+1-c} + \sum_{c=2}^{q+1} w(1, c) y_{t+c-1} \qquad t = q+1, \ldots, n-q$$

For time-varying filters, the weight matrix is of dimension $n \times n$ where n is the number of non-missing observations in the current sample. Row r of the matrix contains the weighting vector used to generate the r-th observation of the filtered series where column c contains the weight on the c-th observation of the original series:

$$z_t = \sum_{c=1}^{n} w(t, c) y_c \qquad t = 1, \ldots, n \qquad (11.104)$$

where z_t is the filtered series, y_t is the original series and $w(r, c)$ is the (r, c) element of the weighting matrix. By construction, the first and last rows of the weight matrix will be filled with missing values for the symmetric filter.

Whiten Data

EViews offers easy-to-use tools for whitening the data in a series. To access these tools, simply select **Make Whitened...**. from the series toolbar or the main EViews menu. See "Make Whitened," on page 613 for a discussion of the use of these tools in a group object setting.

Distribution Plot Data

EViews offers the ability to display a variety of what we term *analytical graphs*—graphs created by first performing data reduction or statistical analysis on the series, then displaying the results visually ("Analytical Graph Types" on page 685). The central feature of these graphs is that they do not show data for each observation, but instead display a summary of the original data. One important class of analytical graphs are *distribution graphs*, which show various aspects of the distribution of the data in a series. Included among these graph types are histograms, kernel density plots, theoretical distribution plots, and theoretical quantile-quantile plots.

Typically, you will access the distribution graph tools by selecting **Graph** from the series window and then making the appropriate choice (see "Graphing a Series" on page 621). These graph tools allow you to display the constructed data graphically, but not to examine them directly.

In some cases, you may wish to save the data depicted in the graph, possibly for further analysis. To do so, simply select **Proc/Make Distribution Plot Data...** to bring up a dialog that is quite similar to the main graph dialog:
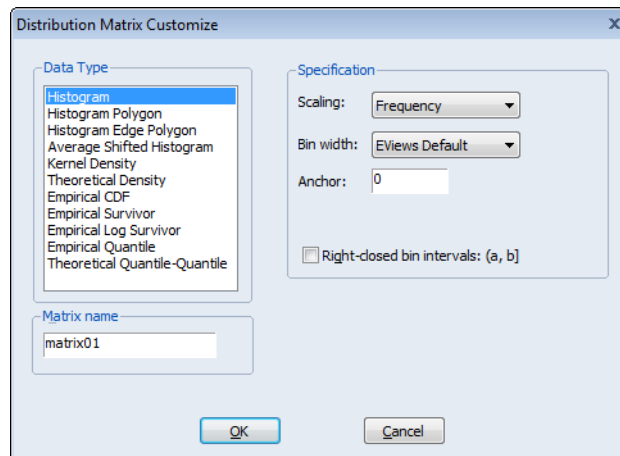
On the left-hand side of the dialog is a list box showing the types of distribution graph data that may be saved, along with the name of a matrix in which to save the results. As you select entries on the left, the right-hand side of the dialog will change to reflect the different options that are available for your selection. Specify your options as desired and click on **OK** to save your results.

References

Aiolfi, M. and Timmermann, A. (2006), "Persistence of forecasting performance and combination strategies", *Journal of Forecasting,* 24, 233–254.

Anderson, T. W. and D. A. Darling (1952). "Asymptotic Theory of Certain Goodness of Fit Criteria Based on Stochastic Processes," *Annals of Mathematical Statistics*, 23, 193-212.

Anderson, T. W. and D. A. Darling (1954), "A Test of Goodness of Fit," *Journal of the American Statistical Association*, 49, 765-769.

Baxter, Marianne and Robert G. King (1999). "Measuring Business Cycles: Approximate Band-Pass Filters For Economic Time Series," *Review of Economics and Statistics*, 81, 575–593.

Bergmann, Reinhard, John Ludbrook, and Will P. J. M. Spooren (2000). "Different Outcomes of the Wilcoxon-Mann-Whitney Test From Different Statistical Packages," *The American Statistician*, 45(1), 72–77.

Bowerman, Bruce L. and Richard T. O'Connell (1979). *Time Series and Forecasting: An Applied Approach,* New York: Duxbury Press.

Box, George E. P. and Gwilym M. Jenkins (1976). *Time Series Analysis: Forecasting and Control*, Revised Edition, Oakland, CA: Holden-Day.

Brown, M. B. and A. B. Forsythe (1974a). "Robust Tests for the Equality of Variances," *Journal of the American Statistical Association*, 69, 364–367.

Brown, M. B. and A. B. Forsythe (1974b). "The Small Sample Behavior of Some Test Statistics which Test the Equality of Several Means," *Technometrics*, 16, 129–132.

Chatfield, Chris, Koehler, Anne B., Ord, J. Keith, and Ralph D. Snyder (2001). "A New Look at Models for Exponential Smoothing," *The Statistician*, 50, Part 2, 147–159.

Chong, Y. Y. and Hendry, D. F. (2006), "Econometric evaluation of linear macro-economic models", *Review of Economic Studies*, 53, 671-690.

Christiano, Lawrence J. and Terry J. Fitzgerald (2003). "The Band Pass Filter," *International Economic Review*, 44(2), 435-465.

Clemen, R. T. (1989) "Combining forecasts: A review and annotated bibliography". *International Journal of Forecasting* 5, 559–581.

Cleveland, R., Cleveland, W., McRae, J., and Terpenning, I. (1990), "STL: A Seasonal-Trend Decomposition Procedure based on Loess", *Journal of Official Statistics*, 6(1), 3-73.

Cochran, W. G. (1937). "Problems Arising in the Analysis of a Series of Similar Experiments," *Supplement to the Journal of the Royal Statistical Society*, 4(1), 102-118.

Conover, W. J., M. E. Johnson and M. M. Johnson (1981). "A Comparative Study of Tests for Homogeneity of Variance with Applications to the Outer Continental Shelf Bidding Data," *Technometrics*, 23, 351–361.

Csörgö, Sandor and Julian Faraway (1996). "The Exact and Asymptotic Distributions of Cramer-von Mises Statistics," *Journal of the Royal Statistical Society, Series B*, 58, 221-234.

D'Agostino and Michael A. Stephens, (*eds.*) (1986). *Goodness-of-Fit Techniques*. New York: Marcel A. Deckker.

Dallal, Gerard E. and Leland Wilkinson (1986). "An Analytic Approximation to the Distribution of Lilliefor's Test Statistic For Normality," *The American Statistician*, 40(4), 294-296.

Davis, Charles S., and Michael A. Stephens (1989). "Empirical Distribution Function Goodness-of-Fit Tests," *Applied Statistics*, 38(3), 535-582.

Dezhbaksh, Hashem (1990). "The Inappropriate Use of Serial Correlation Tests in Dynamic Linear Models," *Review of Economics and Statistics*, 72, 126–132.

Diebold, Francis X., and Robert S. Mariano (2012). "Comparing Predictive Accuracy," *Journal of Business & Economic Statistics*, 13, 253–263.

Durbin, J. (1970). *Distribution Theory for Tests Based on the Sample Distribution Function*. SIAM: Philadelphia.

Findley, David F., Brian C. Monsell, William R. Bell, Mark C. Otto, Bor-Chung Chen (1998). "New Capabilities and Methods of the X-12-ARIMA Seasonal-Adjustment Program," *Journal of Business & Economic Statistics*, 16, 127–152.

Hansen, B. (2008). "Least Squares Forecast Averaging", *Journal of Econometrics*, 146(2), 342-350

Harvey, Andrew C. (1990). *The Econometric Analysis of Time Series*, 2nd edition, Cambridge, MA: MIT Press.

Harvey, Andrew C. (1993). *Time Series Models*, 2nd edition, Cambridge, MA: MIT Press.

Harvey, David, Stephen Leybourne, and Paul Newbold (1997). "Testing the Equality of Prediction Mean Squared Errors," *International Journal of Forecasting*, 13, 281–291.

Hodrick, R. J. and E. C. Prescott (1997). "Postwar U.S. Business Cycles: An Empirical Investigation," *Journal of Money, Credit, and Banking*, 29, 1–16.

Hyndman, R, and Khandakar, Y. (2008), "Automatic Time-Series Forecasting" *Journal of Statistical Software*, 27, 1–22.

Hyndman, Rob J., Koehler, Anne B., Snyder, Ralph D., and Simone Grose (2002). "A State Space Framework For Automatic Forecasting Using Exponential Smoothing Methods," *International Journal of Forecasting*, 18, 439–454.

Hyndman, Rob J., Koehler, Anne B., Ord, J. Keith, and Simone Grose (2008). *Forecasting with Exponential Smoothing: The State Space Approach*. Berlin: Springer-Verlag.

Judge, George G., W. E. Griffiths, R. Carter Hill, Helmut Lütkepohl, and Tsoung-Chao Lee (1985). *The Theory and Practice of Econometrics, 2nd edition*, New York: John Wiley & Sons.

Levene, H. (1960). "Robust Tests for the Equality of Variances," in I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow, and H. B. Mann (*eds.*), *Contribution to Probability and Statistics*, Palo Alto, CA: Stanford University Press.

Lewis, Peter A. W. (1961). "Distribution of the Anderson-Darling Statistic," *Annals of Mathematical Statistics*, 32, 1118-1124.

Ljung, G. and G. Box (1979). "On a Measure of Lack of Fit in Time Series Models," *Biometrika*, 66, 265–270.

Makridakis, S., Hibon, M. (2000). "The M3-competition: Results, conclusions and implications". *International Journal of Forecasting* 16, 451–476.

Neter, John, Michael H. Kutner, Christopher J. Nachtsheim, and William Wasserman (1996). *Applied Linear Statistical Models*, 4th Edition. Chicago: Times Mirror Higher Education Group, Inc. and Richard D. Irwin, Inc.

Ord, J. K., Koehler, A. B., and R. D. Snyder (1997). "Estimation and Prediction for a Class of Dynamic Nonlinear Statistical Models". *Journal of the American Statistical Association*, 92, 1621–1629.

Ravn, Morten O. and Harald Uhlig (2002). "On Adjusting the Hodrick-Prescott Filter for the Frequency of Observations," *Review of Economics and Statistics*, 84, 371-375.

Satterthwaite, F. E. (1946). "An Approximate Distribution of Estimates of Variance Components," *Biometrics Bulletin*, 2(6), 110-114.

Sheskin, David J. (1997). *Parametric and Nonparametric Statistical Procedures,* Boca Raton: CRC Press.

Smith, J, and Yadav, S. (1994) "Forecasting Costs Incurred from Unit Differencing Fractionally Integrated Processes", *International Journal of Forecasting*, 10(4), 507–514.

Sokal, Robert R. and F. James Rohlf (1995). *Biometry*. New York: W. H. Freeman and Company.

Stephens, Michael A. (1986). "Tests Based on EDF Statistics," in *Goodness-of-Fit Techniques*, Ralph B. D'Agostino and Michael A. Stephens, (*eds.*). New York: Marcel A. Deckker, 97-193.

Stock, J. H., Watson, M. (2001). "A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series". In: Engle, R.F., White, H (Eds.), Festschrift in Honour of Clive Granger. Cambridge University Press, Cambridge, 1–44.

Stock, J. H., Watson, M. (2004). "Combination forecasts of output growth in a seven-country data set". *Journal of Forecasting* 23, 405–430.

Timmermann, A. (2006) "Forecast Combinations". *Handbook of Economic Forecasting* (Chapter 4), Elsevier.

Welch, B. L. (1951). "On the Comparison of Several Mean Values: An Alternative Approach," Biometrika, 38, 330–336.

Chapter 12. Groups

This chapter describes the views and procedures of a group object. A group object is a collection of series which you wish to treat as a single unit. With a group, you can compute various statistics that describe the relationship between the series and display them in various forms such as spreadsheets, tables, and graphs.

To create a group object, you may select the desired series in the workfile, then right-mouse click and select **Open/as Group**. Alternately, you may enter the keyword group on the command line, followed by the name to be given the group object, then the list of series and series expressions to be placed in the group.

The remainder of this chapter assumes that you are already familiar with the basics of working with a group. See the documentation of EViews features beginning with Chapter 4. "Object Basics," on page 101 for relevant details on the basic operations.

Group Views Overview

The group view menu is divided into four blocks:

- The views in the first block provide various ways of looking at the actual data in the group.

- The views in the second block display various basics statistics.

- The views in the third block are for specialized statistics typically computed using time series data.

- The fourth block contains the label view, which provides information regarding the group object.

| Group Members |
| Spreadsheet |
| Dated Data Table |
| Graph... |
| Descriptive Stats ▶ |
| Covariance Analysis... |
| N-Way Tabulation... |
| Tests of Equality... |
| Principal Components... |
| Correlogram (1) ... |
| Cross Correlation (2) ... |
| Long-run Covariance... |
| Unit Root Test... |
| Cointegration Test ▶ |
| Granger Causality... |
| Label |

Group Members

This view displays the member series in the group and allows you to alter the group. To change the group, simply edit the group window. You can add other series from the workfile, include expressions involving series, or delete series from the group.
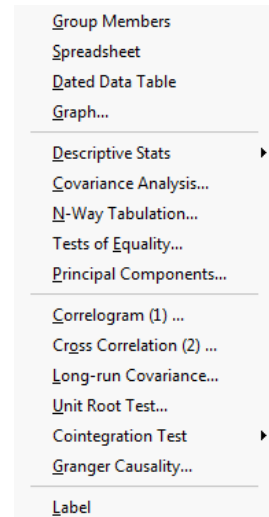
The **Group Members** view displays a list of all the names of the series currently in the group.

You may change the contents of the group by simply dragging and dropping series objects from the workfile window to the **Group Members** window or by right-clicking on the group right-click on the **Group Members** window and selecting **Add**.

The order of the series in the group can be rearranged by dragging and dropping the members into the desired position. Changes that you make to the group are finalized immediately. You may sort the members, by number or name, by clicking on the headers (**#** or **Name**) of the list window.

You may also use a text window to change the group by right-clicking on the **Group Members** window and selecting **Edit Members...** to open a new window.

This window allows you to use a text-based format to add, remove, or rearrange series in a group. To add one series to the group, simply place the edit cursor in the desired position and enter the name of the series or the series expression. Removing or rearranging members may be accomplished by cutting and pasting as desired. You may also use the clipboard to cut-and-paste lists of series from group to group or even from other applications into EViews.

Changes you make to the members view are not finalized until you click the **OK** button on the edit dialog. If you attempt to without pressing **OK**, you will be prompted to save or discard your changes.

Alternately, you may use the right-button menu in the spreadsheet view as described below ("Additional Customization" on page 549) to change the group members, or you may drag-and-drop series into the **Group Members** or **Spreadsheet** views of the group as described in "Adding to or Modifying a Group" on page 135).
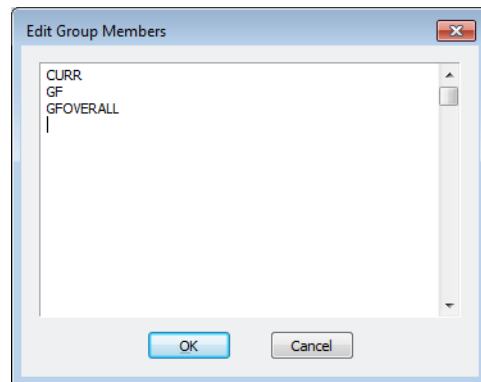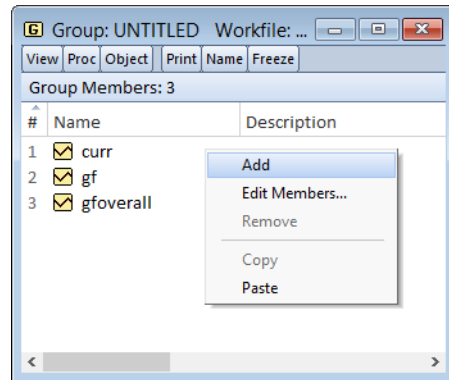
Spreadsheet

This view displays the data, in spreadsheet form, for each series in the group. If you wish, you can flip the rows and columns of the spreadsheet by pressing the **Transpose** button. In

transpose format, each row contains a series, and each column an observation or date. Pressing the **Transpose** button toggles between the two spreadsheet views.

You may change the display mode of your spreadsheet view to show various common transformations of your data using the dropdown menu in the group toolbar. By default, EViews displays the original or mapped values in the series using the formatting specified in the series (**Default**). If you wish, you can change the spreadsheet display to show any transformations defined in the individual series (**Series Spec**), the underlying series data (**Raw Data**), or various differences of the series (in levels or percent changes), with or without log transformations.

```
Series Spec
Default
Raw Data
Differenced
Year Dif
% Change
% Chg A.R.
Year % Chg
Log
Log Dif
```
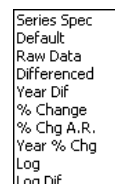
You may edit the series data in either levels or transformed values. The **Edit +/-** on the group toolbar toggles the edit mode for the group. If you are in edit mode, an edit window appears in the top of the group window and a double-box is used to indicate the cell that is being edited.

obs	GDP	M1	PR
1952Q1	87.87500	126.5370	0.197561
1952Q2	88.12500	127.5060	0.198167
1952Q3	89.62500	129.3850	0.200179
1952Q4	92.87500	128.5120	0.201246
1953Q1	94.62500	130.5870	0.201052
1953Q2	95.55000	130.3410	0.201444
1953Q3	95.42500	131.3890	0.202236
1953Q4	94.17500	129.8910	0.202723
1954Q1	94.07500	130.1730	0.203416
1954Q2	94.20000	131.3850	0.203841
1954Q3	95.45000	134.6270	0.204291
1954Q4			

Here, we are editing the data in the group in 1-period percent changes (note the label to the right of the edit field). If we change the 1952Q4 value of the percent change in GDP, from 3.626 to 5, the values of GDP from 1952Q4 to the end of the workfile will change to reflect the one-time increase in the value of GDP.

Additional Customization

EViews provides you with additional tools for altering the display of your spreadsheet. To change the display properties, select one or more series by clicking on the series names in the headers, then right-click to bring up a menu.

```
Open Selected

Copy
Paste
Paste Special...

Insert Series
Remove Series

Display format...
Clear Contents
Sort              ▶

Save table to disk...
```

Open selected will open a new group object containing the selected series in a new, untitled group.

You may use the **Insert Series** and **Remove Series** entries to change the contents of the current group.
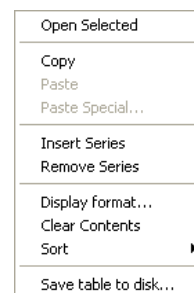
Selecting **Insert Series** brings up a dialog which allows you to insert multiple series by simply entering their names or series expressions. The new series will be inserted after the currently selected series and all subsequent series will follow the newly inserted series. **Remove Series** works slightly differently, in that no dialog will be shown and all selected series will be removed from the group. (See also "Group Members" on page 547 for an alternative method of changing the group contents.)
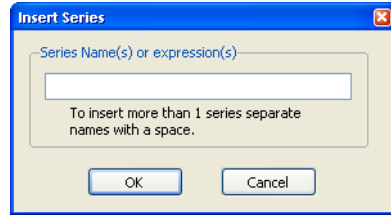
If the current ordering of the series is not desired, you may change the order by dragging a series to the desired position. Dragging is achieved by first selecting one or more series. Once the selection has been made, move the mouse to either the left edge of the first selected series or to the right edge of the last selected series. Once over the edge, the cursor should change to the cursor shown in the following image. From here, press the left mouse button and drag the series to the desired location.

If you right-click and then select **Display format...** EViews will open a format dialog that will allow you to specify group display characteristics that override the individual series display characteristics. Once you specify the desired format and click on **OK**, EViews will update the group display to reflect your specification.
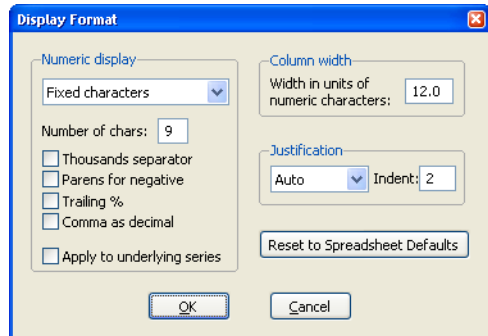
Note that, by default, changes to the group display format will apply only to the group spreadsheet and will not change the underlying series characteristics. If, for example, you elect to show series X in fixed decimal format in the group spreadsheet, but X uses significant digits in its individual series settings, the latter settings will not be modified. To update the display settings in the selected series, you must select the **Apply to underlying series** checkbox in the format dialog.

You may display the observations in the group in sorted order using the **Sort/by dialog** menu item in the right-mouse menu. EViews will open the **Sort Order** dialog, prompting you to select sort keys and orders for up to three series.

When you click on **OK**, EViews will rearrange the group spreadsheet display so that observations are displayed in the specified order.
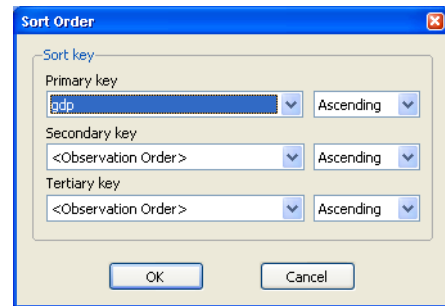
A quicker alternative, assuming you only want to sort by only one series, is to use the right-mouse button menus. Selecting **Sort/Ascending** or **Sort/Descending** sorts the observations using the first selected column of data.

Note that the underlying data in the workfile is not sorted, only the display of observations and observation identifiers in the group spreadsheet. This method of changing the spreadsheet display may prove useful if you wish to determine the identities of observations with high or low values for some series in the group.

Selecting **Clear Contents** sets the selected observations and variables to NA.

Lastly, you should note that as with series, you may write the contents of the spreadsheet view to a CSV, tab-delimited ASCII text, RTF, or HTML file by selecting **Save table to disk...** and filling out the resulting dialog.

It is worth pointing out that if you select a set of specific cells in the group spreadsheet and then right-click, you will be presented with a slightly different menu that, among other things, allows you to copy the contents, insert, or delete observations.
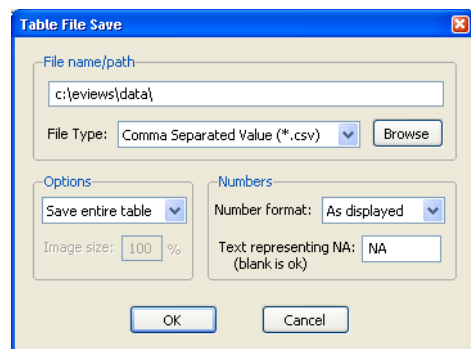
Group Comparison Tools

EViews offers tools so that you may compare the data between the series in your group. When looking at the **Spreadsheet** view of the group, simply press the **Compare +/-** button on the toolbar to enter Compare mode. Compare mode will behave differently depending upon whether there are only two series in the group, or more than two.

In both cases the main feature of compare mode is that it will highlight, in red, any observations for which the series in the group have different values. This can be useful when com-

paring revisions to series in order to quickly find for which observations any revisions or changes have been made.

By default the comparison of the series is done with zero tolerance - even the slightest difference will cause EViews to highlight the row. You can change this tolerance level by right clicking anywhere on the spreadsheet when in compare mode and selecting the **Highlighting...** item. This will bring up a dialog that allows you to change the tolerance level by changing the number in the **Highlight if difference exceeds** box.

With only two series in the group, compare mode will also show additional columns containing information on how much the data in the two series differ. The following example shows compare mode being used on two series, UNRATE and UNRATE_REV, in the group G:

We are able to quickly see that there are three observations for which the two series differ: 2005M08, 2005M12 and 2006M03. Compare mode has added two columns to the spreadsheet, one containing "Delta", which shows the raw difference between the two series (UNRATE - UNRATE_REV), and the second containing "Delta%", which is the percentage difference ((UNRATE-UNRATE_REV)/UNRATE_REV). Note that for ease of use, in cases where there is no difference between the two series, the Delta and Delta% columns are left blank instead of showing "0".

All four of these columns is fully editable. A quick way to remove any difference between the two series is simply to type a "0" into the Delta column (or simply delete a value in it). Similarly, a quick way to make UNRATE equal to 10% higher than UNRATE_REV is to type "10" in the Delta% column. Note that when editing Delta and Delta%, the change is made to the first series (UNRATE in our case), and not the second

If you change the display type of the group to a setting other than "Default" or "Raw data", rather than replace the raw series values with the transformation you selected, EViews will remove the "Delta%" column and introduce new columns containing the transformation, and a new "Delta" column containing the difference for that transformation. For example, if we switch the display type of our group to "**% Change**", the compare view will contain 6 columns of data - the original UNRATE and UNRATE_REV series, the "Delta" column for those two series, an @PCH(UNRATE) column (containing the percentage change for UNRATE), an @PCH(UNRATE_REV) column (containing the percentage change for UNRATE_REV), and finally a "Delta" column containing the difference between the two percentage change columns:

obs	UNRATE	UNRATE...	Delta	@PCH(UNRATE)	@PCH(UNRATE_REV)	Delta
2005M01	5.700000	5.700000		NA	NA	
2005M02	5.800000	5.800000		1.75	1.75	
2005M03	5.400000	5.400000		-6.90	-6.90	
2005M04	4.900000	4.900000		-9.26	-9.26	
2005M05	4.900000	4.900000		0.00	0.00	
2005M06	5.200000	5.200000		6.12	6.12	
2005M07	5.200000	5.200000		0.00	0.00	
2005M08	4.900000	5.200000	-0.300000	-5.77	0.00	-5.77
2005M09	4.800000	4.800000		-2.04	-7.69	+5.65
2005M10	4.600000	4.600000		-4.17	-4.17	
2005M11	4.800000	4.800000		4.35	4.35	
2005M12	4.600000	4.700000	-0.100000	-4.17	-2.08	-2.08
2006M01	5.100000	5.100000		10.87	8.51	+2.36
2006M02	5.100000	5.100000		0.00	0.00	
2006M03	4.800000	4.900000	-0.100000	-5.88	-3.92	-1.96
2006M04	4.500000	4.500000		-6.25	-8.16	+1.91
2006M05	4.400000	4.400000		-2.22	-2.22	
2006M06	4.800000	4.800000		9.09	9.09	
2006M07	5.000000	5.000000		4.17	4.17	
2006M08	4.600000	4.600000		-8.00	-8.00	
2006M09	4.400000	4.400000		4.35	4.35	

Again, each of these columns is editable. For example, a quick way to ensure that the two series grow at the same rate, even if they start at different values, is to enter a "0" in the "Delta" column for the @PCH columns after that date.

When your group contains more than two series, compare mode does not add any extra columns to the spreadsheet, and merely provides color highlighting for any columns which have different values. The series in the group will be compared in pairs.

See Appendix B. "Enhanced Spreadsheet Editing," on page 885 for a description of array expressions that may be used in editing your data.

Dated Data Table

The dated data table view of a group offers sophisticated tools to help you construct tables that combine original data along with transformations, frequency conversions, and summary statistics of the data.

Time series data, forecasts, and simulation results are often presented in complex tables that are constructed to highlight various aspects of the data. Suppose, for example, that you wish to display values of the quarterly GDP and PR series in the years 1994–1996 (in the workfile "Demo.wf1"), along with the annual average, arranged by year:

	I	II	III	IV	Year
		1994			1994
GDP	1,698.6	1,727.9	1,746.7	1,774.0	1,736.8
PR	1.04	1.05	1.05	1.06	1.05
		1995			1995
GDP	1,792.3	1,802.4	1,825.3	1,845.5	1,816.4
PR	1.07	1.07	1.08	1.09	1.08
		1996			1996
GDP	1,866.9	1,902.0	1,919.1	1,948.2	1,909.0
PR	1.09	1.10	1.11	1.11	1.10

Or perhaps you wish to display the original data along with the one-year percent changes with more descriptive display labels taken from the series:

	I	II	III	IV	Year
		1994			1994
Gross Domestic Product	1,698.6	1,727.9	1,746.7	1,774.0	1,736.8
(year % ch.)	*5.43*	*6.18*	*6.27*	*5.84*	*5.93*
Price Label	1.04	1.05	1.05	1.06	1.05
(year % ch.)	*2.25*	*2.31*	*2.49*	*2.49*	*2.39*
		1995			1995
Gross Domestic Product	1,792.3	1,802.4	1,825.3	1,845.5	1,816.4
(year % ch.)	*5.51*	*4.31*	*4.50*	*4.03*	*4.58*
Price Label	1.07	1.07	1.08	1.09	1.08
(year % ch.)	*2.69*	*2.62*	*2.50*	*2.38*	*2.55*
		1996			1996
Gross Domestic Product	1,866.9	1,902.0	1,919.1	1,948.2	1,909.0
(year % ch.)	*4.16*	*5.52*	*5.14*	*5.57*	*5.10*
Price Label	1.09	1.10	1.11	1.11	1.10
(year % ch.)	*2.29*	*2.22*	*2.34*	*2.24*	*2.27*

Alternatively, you may wish to display annual averages for each year, followed by the last four quarterly observations in your sample:

	1994	1995	1996	1996q1	1996q2	1996q3	1996q4
GDP	1,736.8	1,816.4	1,909.0	1,866.9	1,902.0	1,919.1	1,948.2
PR	1.05	1.08	1.10	1.09	1.10	1.11	1.11

In all three of these examples, the dated data table view offers an easy-to-use set of tools for quickly specifying the form of the table, performing the required calculations, and arranging the results into attractive output.

You can even use the dated data table to input data by constructing a table that matches the format of a published tables and editing the dated data table cells to match.

Creating a Dated Data Table

To display the dated table view you must first create a group object containing the series of interest. Once you have created the group, simply select **View/Dated Data Table** to construct the table.

(*Note that due to the nature of the operations involved in constructing a dated data table, the view is only available in an annual, semi-annual, quarterly, or monthly workfile.*)

The group window will change to display a dated data table view created using the current workfile sample and the default dated data table settings:

At the default settings, each line contains a single year of data for each series, at the original (non-annual) frequency along with the annual average, or each line will show up to five years of data in an annual workfile.

Customizing a Dated Data Table

You may set options to customize an existing dated data table view through the **Dated data table options** dialog. First create a dated data table, then click on the **Table Options** button on the toolbar to open the **Dated data table options** dialog.

The dialog has four separate tabs: **Table Options**, **Data Format**, **Fonts**, and **Labels/Headers**. We will discuss, in turn, the use of each of these settings for customizing the contents and look of your dated data table.

Note that once you modify the dated data table options for a group, EViews will remember these settings the next time you open the view for the group.

Table Options

The first tab in the dialog, **Table Options**, provides options to control the general style of the table.
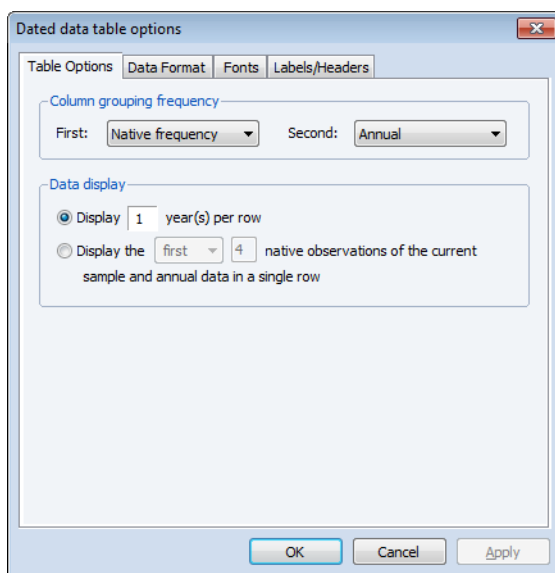
You may use these settings to specify whether you want the annual data arranged on the left-hand side of the table with a year's worth of monthly data on the right, the native data on the left and the quarterly data on the right, any other combination of native and lower frequency data, or data at a single frequency.

Defining the Column Groupings

The dated data table view displays data in one or two groups of columns. The contents of the groups are defined in the **Column grouping frequency** section of the page. The **First** and **Second** drop-down menus determine the contents of the two groups. You may choose from a subset of the following:

- **Native frequency** (the frequency of the workfile)
- **Annual**

- **Quarterly**

- **Monthly**

- **None** (only available for the second set of columns)

with the available choices depending on the native frequency of the workfile (only native or lower frequencies are allowed) and on the column grouping.

By default, the **First** drop-down will be set to **Native frequency**. Given this setting, the first group of columns will contain data at the original workfile frequency. For example, if your workfile contains monthly data, then the first group will contain monthly data; if your workfile is quarterly, the group will contain quarterly data.

You may instruct EViews to display the underlying data at a lower frequency in the first group of columns. You may, for example, choose to display your monthly workfile data at a quarterly or an annual frequency. EViews will perform a frequency conversion to the desired lower frequency using the conversion method specified in "Data Format" on page 559, and will display those results in the first group.

By default, the **Second** menu is set to **Annual** so that the second group of columns is actually a single column of annual data. EViews will convert the data from the native frequency to an annual frequency using the conversion method specified in "Data Format" on page 559.

You may use the drop-down menu to select **Native frequency** or any lower frequency, or you may select **None** to remove the second block of columns.
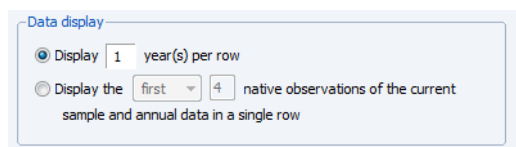
If the specified settings for the first and second groups produce duplicate data, EViews will remove groups of columns to eliminate redundancy. Thus, if you have an quarterly workfile and select **Native frequency** and **Quarterly**, respectively, duplicate quarterly results will not be shown.

Specifying the Data in a Line

The **Data display** area allows you to control the amount of data contained in each line of table output. You may choose between displaying multiple years of data in a row or displaying a single year of data with a subset of observations from that year.

To select multiple years of data, select the first radio button and enter the number of years per row you would like to display. The data displayed will be for the years included in the current workfile sample only (note this means

that if your sample starts half way through a year, EViews will display all periods of that year, including those prior to the start of the sample, if they exist in the workfile).

The impact of selecting the second radio button depends upon the selection in the **Column grouping frequency** area. If you have chosen to show annual data in one of the column groupings, selecting the second radio button will tell EViews to display a row with a single entry for each year in your current workfile sample for that column grouping.

If you have chosen a non-annual frequency for one of the column groupings, selecting the second radio will tell EViews to only display observations from either the start (**first**) or end (**last**) of the first year of your sample in that grouping. You can enter the number of periods/observations you would like to show.

The effects of these choices on the table display are best described by the following examples. For purposes of illustration, note that the current workfile is quarterly, with a current sample of 1993q1–1996q4.

Now suppose that you choose the first **Data display** style with 2 years of data, with the first columns set to the native frequency, and the second columns set to annual frequency.

Each row of the resulting table will contain eight quarters of data (two years of the native frequency data) followed by two annual observations (two years of the annual frequency data):

	I	II	III	IV		I	II	III	IV		Year	
			1993					1994			1993	1994
GDP	1,611.1	1,627.3	1,643.6	1,676.0		1,698.6	1,727.9	1,746.7	1,774.0		1,639.5	1,736.8
PR	1.02	1.02	1.03	1.04		1.04	1.05	1.05	1.06		1.03	1.05
			1995					1996			1995	1996
GDP	1,792.3	1,802.4	1,825.3	1,845.5		1,866.9	1,902.0	1,919.1	1,948.2		1,816.4	1,909.0
PR	1.07	1.07	1.08	1.09		1.09	1.10	1.11	1.11		1.08	1.10

EViews automatically performs the frequency conversion to annual data using the specified method (see "Transformation Method" on page 560).

If you reverse the ordering of data types in the first and second columns so that the first columns display the annual data, and the second columns display the native frequency,

the resulting dated data table will contain:

	1993	1994	I	II	III	IV	I	II	III	IV
				1993				1994		
GDP	1,639.5	1,736.8	1,611.1	1,627.3	1,643.6	1,676.0	1,698.6	1,727.9	1,746.7	1,774.0
PR	1.03	1.05	1.02	1.02	1.03	1.04	1.04	1.05	1.05	1.06
	1995	1996		1995				1996		
GDP	1,816.4	1,909.0	1,792.3	1,802.4	1,825.3	1,845.5	1,866.9	1,902.0	1,919.1	1,948.2
PR	1.08	1.10	1.07	1.07	1.08	1.09	1.09	1.10	1.11	1.11

Lastly, specify **Annual** for the first columns and **Native frequency** for the second columns. Then choose the second display style, select **last**, and enter "4" in the edit box:

EViews will display the annual data for all four years of the current sample, followed by the last four native (quarterly) observations:

	1993	1994	1995	1996	1996q1	1996q2	1996q3	1996q4
GDP	1,639.5	1,736.8	1,816.4	1,909.0	1,866.9	1,902.0	1,919.1	1,948.2
PR	1.03	1.05	1.08	1.10	1.09	1.10	1.11	1.11

Data Format

Where the **Table Options** tab ("Table Options" on page 556) determines the basic layout of the table and the frequencies at which you will display the data, the **Data Format** tab of the dialog determines the actual values and display formats of the data in the table.

The settings in this tab control:

- whether to display one or two lines of values for each series in the group
- whether to display the original data or transformations of the data

- the method for performing any frequency conversion
- the format for displaying the numbers

Select Data to Customize

The first step in setting the data format is to select the table elements for which you wish to change settings. You will find the controls for selecting data in the **Data to customize** section of the dialog.

- The **Series** dropdown allows you to specify the data format for all of the series in the table, or for a specific series.
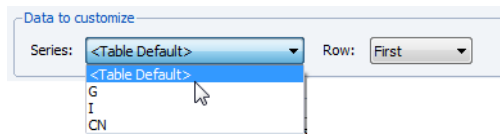
Select the **< Table Default >** entry to specify default settings for all of the series in the table. Changes to the display options and formatting of the table defaults will affect all series which employ default settings.

To modify the settings for a specific series, select the name of the series in the drop-down menu.

- The **Row** dropdown lets you choose whether to change the data format settings for the first or for the second display row of the element selected in **Series**.

EViews allows you to display either one or two rows for each series. You may, for example, display the levels of the series in one row and the 1-period percent change in a second row. Changing the **Row** drop-down between **First** or **Second** in allows you to choose whether to modify settings for the first or the second row of the element selected in **Series**.

As you use the **Series** and **Row** menus to select a table data element, the remaining sections of the page will change show the data format settings corresponding to the selections.

Specify Transformation and Frequency Conversion

Once you used **Data to customize** to select a data table element to customize, you may use the **Display options** section of the page to transform the data prior to display, and if relevant, to select a frequency conversion method to use.

Transformation Method

The **Transformation** menu offers quick access to the most commonly employed transformations:

Level (no transformation)	No transformation
1 Period Difference	$(y - y(-1))$
1 Year Difference	$y - y(-f)$, where $f = \begin{cases} 1 & \text{for annual} \\ 2 & \text{for semi-annual} \\ 4 & \text{for quarterly} \\ 12 & \text{for monthly} \end{cases}$
1 Period % Change	$100 \times (y - y(-1)) / y(-1)$
1 Period % Change at Annual Rate	Computes R such that: $$(1 + r / 100)^f$$ where f is defined above and r is the 1 period % change.
1 Year % Change	$100 \times (y - y(-f)) / y(-f)$, where f is defined above.

In addition, if the **Data to customize** setting for **Row** is set to **Second**, the dropdown will also offer the choice of **No second row** so that you may eliminate the second row display entirely.

We emphasize that the transformation menu includes only the methods most commonly employed in constructing tables. If you wish to display the data with other transformations, you may always add an appropriate auto-series to the original group.

Frequency Conversion

The **Frequency conversion** menu allows you to select the desired frequency conversion for the specified data:

Average then Transform	First convert by taking the average, then transform the average.
Transform then Average	First transform the series, then take the average of the transformed series.
Sum then Transform	First convert by taking the sum, then transform the sum.
First Period	Convert by taking the first quarter of each year or first month of each quarter/year.
Last Period	Convert by taking the last quarter of each year or last month of each quarter/year.

The choice between **Average then Transform** and **Transform then Average** changes the ordering of the transformation and frequency conversion operations. The methods produce different results only for the nonlinear "% change" methods.

Set Data Labelling and Formatting

EViews lets you control formatting of the display of the selected table element using the controls in the **Data units & label format** section.

If you are modifying the table defaults, the dialog allows you to define the default settings:

As you can see, you may specify format, units, label, separator and delimiter settings. Changes in settings for the table default will affect formatting for all rows in the table which employ the defaults.

If you are modifying settings for an individual series row, the dialog changes to display radio buttons offering the option of using the default settings (**Use table default settings**), or overriding the default with series specific (**Manual**) formatting:

If you select **Manual**, you will be prompted to specify format, units, label, separator and delimiter settings.

For both table default and individual series row settings, the basic numerical display format may be set using the **Format** dropdown. The basic format options are:

Auto format	EViews chooses the format depending on the data.
Fixed decimal	Specify how many digits to display after the decimal point. This option aligns all numbers at the decimal point.
Fixed chars	Specify how many total characters to display for each number.

Generally, the default auto formatting will produce appropriate output formatting, but if not, simply select the desired fixed decimal or character method and enter the appropriate integer value in the edit field.

You may also specify the **Units** in which to display your data. The menu allows you to choose between the default **Native**, and **Percent: .01**, **Thousands: 1e3**, **Millions: 1e6**, **Billions: 1e9**, **Trillions: 1e12**).

If you select an entry other than **Native**, EViews will transform the data into the units selected. For example, if you have a data point containing the value 1234567, setting the units to thousands changes the display of that data point to 1234.567, and setting the units to millions changes displays 1.234567. Again this change in units is for display purposes only and does not alter the underlying data.

Note that EViews will round and transform your data prior to display in order to fit the specified format. We emphasize, however, that this data adjustment is for display purposes and does not alter the original data.

The remaining settings allow you to assign a custom label prefix or suffix to the number (*i.e.*, a currency or percentage symbol), to specify the separator to denote thousands and decimals (comma or period), and to determine whether to use surrounding parentheses to indicate negative numbers.

Font Settings

The **Fonts** tab of the dated data table dialog lets you change the font styles used in the table.
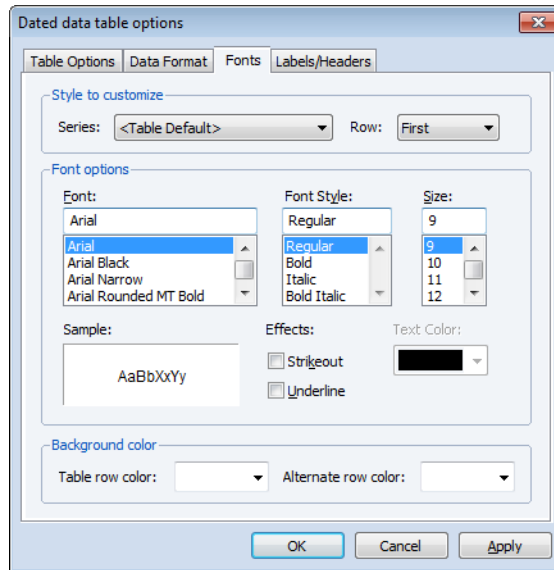
As in the **Data Format** tab, the top portion of the dialog is used to select the element to customize. The **Series** dropdown allows you to set the table default or change the font of a specific series in the table. If you chose to have two rows of data per series on the **Data Format** tab, the **Row** dropdown lets you choose whether to change the data format settings for the first display row, for the second display row, or for the custom header.

The **Font options** section of the dialog may be used to change the text format of the selected table elements. You should use the settings to specify the font type, style, size, color, and effects of the element selected at the top of the dialog.

If you are modifying the fonts of an individual series, the **Font options** section will contain a **Reset to default** button in the bottom right-hand corner. The button will be enabled if the current settings differ from the table defaults. Clicking on the button clears the individual modifications.
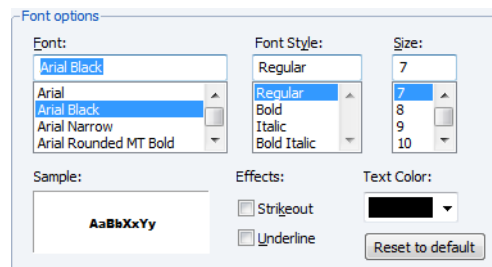
The **Background color** area allows you to assign background color highlighting to rows in your table:

- If you are modifying the table defaults, the **Background color** area will contain two color selectors labeled **Table row color** and **Alternate row color** (*note that these settings are not row-specific so they do not depend on the selected row*). By default, both

selectors are set to the color white. To change the color, simply click on the color to bring up the picker dialog and choose the desired color.

Modifying the **Table row color** will change the color of the *odd num-bered rows* in your table. Changing the **Alternate row color** dropdown will change the color in the *even numbered rows* of the table.

- If you are modifying the settings for an individual series, the **Background color** area will display settings for the specific series row.

By default, the **Use table default color** checkbox is selected. Unse-lecting this option enables the **Series row color** option. To change the series background color, click on the color box to bring up the picker and make your selection. Note that you may select the **Use table default color** option to restore the series color to the default table setting.

Labels and Headers

You may use the **Labels/Headers** tab to add custom series headers, apply italic or bold for-matting to the column (variable) and row (date label) headers, set date formatting, and specify the text to represent missing values:

- The **Series options** section allows you to insert a blank line or custom header for a data series. If you set this option, EViews will insert a header row above each row of data for the specified series.

  You may select the desired series in the **Series** drop-down and check the **Insert a blank line or custom header** box.

  You may enter the custom header text in the edit field, or leave the field blank. The custom header text may also be edited directly in the dated data table by clicking on the **Edit +/-** button on the table button bar and entering text in a blank header cell.

- You may use the **Header options** section to add bold or italic formatting to the row or column headers. Simple select **Row** or **Column** in the **Header type** drop-down menu and check the **Bold** and **Italic** checkboxes as desired to apply formatting.

- The **Label options** section offers controls for value and series labeling.

  The **Label for NA** edit field contains the text representation for missing values in the dated data table. You may replace the default "--" with the desired characters.

  If your series has display names, you may use the display name as the label for the series by selecting the **Use display names as default labels** option. (See "Naming Objects" on page 115 and "Labeling Objects" on page 116 for discussion of display names and the label view.)

- The **Date formats** section controls the representation of dates in the table output. You may use the three drop-down menus to set the format used for representing **Year**, **Quarter** and **Month**. You can also specify whether date periods will be labelled using the last date of the period rather than the first date of the period by selecting the **Use end of period date labels** check box.

Editing a Dated Data Table

Along with customizing your dated data table using the options dialog, you may edit the individual cells of the table, add a title, or change the sample of observations used to construct the table.

To edit individual cells, you must first click on the **Edit +/–** button in the group toolbar to toggle edit mode. You may then edit series (row) labels, custom headers, and data values in the table.

It is important to note that *if you edit a data cell, the underlying series data will change.* This is an important feature allows you to use dated data tables for data entry. Moreover, editing transformed data will modify the original series data in an appropriate fashion. Thus, you may edit data in period differences or one-year percent changes or any other supported transformation method (see "Transformation Method" on page 560), and have the original data change to match your edits in the transformed data.

Note however, that any changes that you make to row labels will not apply to the underlying series, only to the dated data table view.

Lastly, the dated data table view offers two toolbar buttons which allow for quick modification of the table:

- **Title** allows you to add a title to the table
- **Sample** allows you to change the sample used to construct the table

Dated Data Table Templates

You may store and load all of your dated data table settings using dated data templates.

To create a template from an existing dated data table, click on **TableOptions**, then select the **Templates** tab.
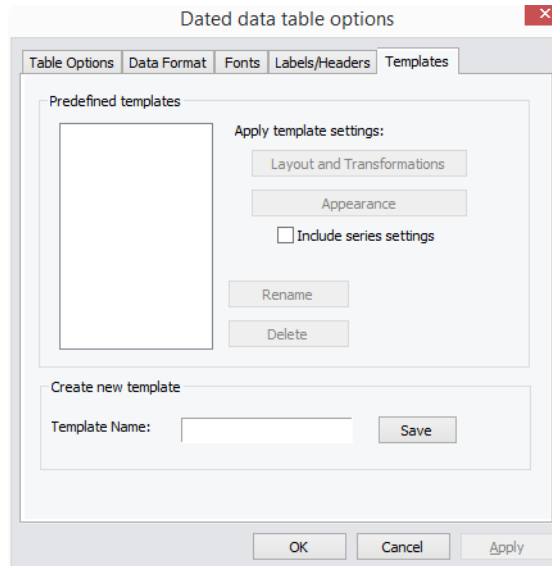
In the bottom portion of the dialog page you will see a **Template Name** edit field. Enter a valid EViews name for the template, and click on **Save**.

To apply the template to a dated data table, open the group and display the dated data table. Click on the **TableOptions**, select the **Templates** tab, then select the template you wish to use.
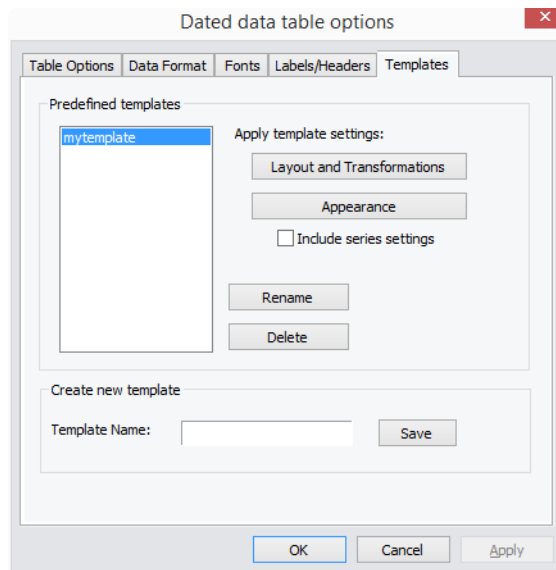
Clicking on the **Layout and Transformations** and **Appearance** button updates the current dated data table settings to match those of the template.

- The **Layout and Transformations** correspond to the basic table layout options including the frequency specifications and other options found on the **Table Options** tab, along with the table default row transformations.

- The **Appearance** button updates the data label formatting, fonts, and labels and headers.

- You should use the **Include series settings** checkbox in conjunction with the update buttons if you wish to import the series specific data format, font, or custom header overrides.

You may use the **Rename** and the **Delete** button to manage your existing templates.

An Illustration

As an illustration of these settings we use the "Macromod.wf1" workfile containing US macro economic data. We set the workfile sample to "1960 2005" and open a group containing the series G, the series I, and the series CN.

By default the dated-data table view for this group looks like the following:

The first column grouping shows quarterly data for each series for each year. The second grouping shows an annual summary. Each series has a single line of data per year.

We will use the dated data table options dialog to customize this table. First, click on the **TableOptions** button, go to **Data Format** tab, and change the table default **Data units & label format** to have "$" as the prefix and "b" (for billions) as the suffix:

Next, we specify a second row with transformation for the G series. Specify the data table element to modify by selecting **G** and **Second** in the **Data to customize section**, and set the **Transformation** drop-down menu to **1 Year % Change** with **Frequency Conversion** set to **Average then Transform**:
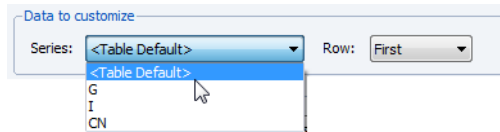
Note that transformation for the first row of G should still be set to the table default.

Lastly, to display a blank header line above every row of the G series, click on the **Label/Headers** tab, select the **Insert a blank line** checkbox, and leave the **Custom header** edit field blank.
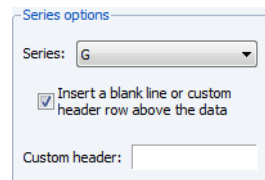
Click on **OK** to accept the changes. The modified dated data table now looks as follows:

Notice the prefix and suffixes which have been added to the data values for all of the series, and the extra header line separating the year headers and the data for the G series.

We will make two final changes. First, click on the **TableOptions** button and go to the **Fonts** tab. With the **Series** drop-down set to " < Table default > ", we may set the **Table row color** field to show a gray and the **Alternate row color** to yellow.

Clicking on **OK** yields the following table:

in which odd rows have a gray background and even rows have a yellow background.

Lastly, we override the default background color settings for the G series. Click on the **Fonts** tab in the options dialog and select "G" and **Second** in the **Series** and **Row** drop-down menus.

Deselect the **Use table default color** checkbox and select a color using the **Series row color** picker. Here we have elected to color the annual percentage change row of our Government spending series (G) in orange.

Click on **OK** to accept the changes. Our final table is depicted below:

Graph

The **Graph...** menu item brings up the **Graph Options** dialog, which allows you to select various types of graphical display of the group. You can create graph objects by freezing these views. See Chapter 13. "Graphing Data," beginning on page 617 for a discussion of techniques for creating and customizing the graphical display.

Descriptive Statistics

These views display the summary statistics of each series in the group. Details for each statistic are provided in "Descriptive Statistics & Tests" on page 402.

Common Sample
Individual Samples

- **Common Sample** computes the statistics using observations for which there are no missing values in any of the series in the group (casewise deletion of observations).

- **Individual Samples** computes the statistics using all nonmissing observations for each series (listwise deletion).

The two views produce identical results if there are no missing values, or if every series has missing observations for the same set of observations.

	GDP	PR		
Mean	632.4190	0.514106		
Median	374.3000	0.383802		
Maximum	1948.225	1.110511		
Minimum	87.87500	0.197561		
Std. Dev.	564.2441	0.303483		
Skewness	0.845880	0.592712		
Kurtosis	2.345008	1.829239		
Jarque-Bera	24.68300	20.81933		
Probability	0.000004	0.000030		
Sum	113835.4	92.53909		
Sum Sq. Dev.	56988478	16.48625		
Observations	180	180		

Covariance Analysis

The covariance analysis view may be used to obtain different measures of association (covariances and correlations) and associated test statistics for the series in a group. You may compute measures of association from the following general classes:

- ordinary (Pearson product moment)
- ordinary uncentered
- Spearman rank-order
- Kendall's tau-a and tau-b

EViews allows you to calculate partial covariances and correlations for each of these general classes, to compute using balanced or pairwise designs, and to weight individual observa-

tions. In addition, you may display your results in a variety of formats and save results to the workfile for further analysis.

Performing Covariance Analysis

We consider the stock price example from Johnson and Wichern (1992, p. 397) in which 100 observations on weekly rates of return for Allied Chemical, DuPont, Union Carbide, Exxon, and Texaco were examined over the period from January 1975 to December 1976 ("Stocks.WF1"). These data are in the group object G2 containing the series ALLIED, DUPONT, UNION.

To proceed, simply open the group object and select **View/Covariance Analysis...** to display the covariance dialog:

We will consider the various options in detail below. For now, note that by default, EViews will compute the unweighted ordinary (Pearson product moment) covariance for the data in the group, and display the result in a spreadsheet view.

The current sample of observations in the workfile, "1 100", will be used by default, and EViews will perform listwise exclusion of cases with missing values to balance the sample if necessary.

Click on **OK** to accept the defaults, and the group display changes to show the covariances between the variables in the group. The sheet header clearly shows that we have computed the covariances for the data. Each cell of the table shows the variances and covariances for the corresponding variables. We see that the rates of return are positively related, though it is difficult to tell at a glance the relative strengths of the relationships.

Covariance			
	ALLIED	DUPONT	EXXON
ALLIED	0.001614	0.000809	0.000438
DUPONT	0.000809	0.001217	0.000383
EXXON	0.000438	0.000383	0.000794

Statistics

Let us now consider some of the options in the dialog in greater detail. The first section of the dialog, labeled **Statistics**, controls the statistics to be calculated, and the method of displaying our results.

First you may use the **Method** dropdown to specify the type of calculations you wish to perform. You may choose between computing ordinary Pearson covariances (**Ordinary**), uncentered covariances (**Ordinary (uncentered)**), Spearman rank-order covariances (**Spearman rank-order**), and Kendall's tau measures of association (**Kendall's tau**).

The checkboxes below the dropdown menu identify the statistics to be computed. Most of the statistics are self-explanatory, but a few deserve a quick mention

The statistic labeled **SSCP** refers to the "sum-of-squared cross-products." The **Number of cases** is the number of *rows* of data used in computing the statistics, while the N**umber of obs** is the obviously the number of *observations* employed. These two values will differ only if frequency weights are employed. The **Sum of weights** will differ from the number of cases only if weighting is employed, and it will differ from the number of observations only if weights are non-frequency weights.

If you select **Kendall's tau** from the **Method** dropdown, the checkbox area changes to provide you with choices for a different set of covariance statistics. In addition to the previously offered number of cases and obs, and the sum of weights, EViews allows you to display Kendall's tau-a and tau-b, the raw concordances and discordances, Kendall's score statistic, and the probability value for the score statistic.

Turning to the layout options, EViews provides you with up to four display options: **Spreadsheet**, **Single table**, **Multiple tables**, and **List**. We have already seen the spreadsheet view of the statistics. As the names suggest, the two table views lay out the statistics in a table view; the single table will stack multiple statistics in a single "cell" of the table, while the multiple tables will place the table for the second statistic under the table for the first statistic, *etc.* The list view displays each of the statistics in a separate column of a table, with the rows corresponding to pairs of variables. Note that the spreadsheet view is not available if you select multiple statistics for display.
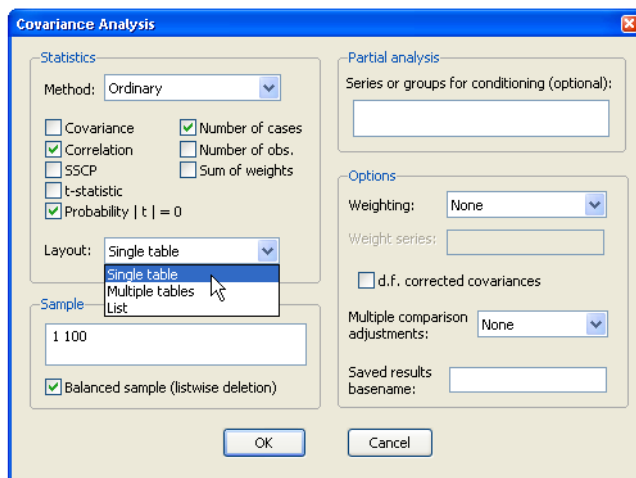
In this example, we perform ordinary covariance analysis and display multiple statistics in a single table. We see that all of the correlations are positive, and significantly different from zero at conventional levels. Displaying the **Correlation** instead of **Covariance** makes it easier to see that the two chemical companies, Allied and DuPont are more highly correlated with each other than they are with the oil company Exxon.

Covariance Analysis: Ordinary
Date: 08/15/09   Time: 17:39
Sample: 1 100
Included observations: 100

Correlation Probability Cases	ALLIED	DUPONT	EXXON
ALLIED	1.000000 ----- 100		
DUPONT	0.576924 0.0000 100	1.000000 ----- 100	
EXXON	0.386721 0.0001 100	0.389519 0.0001 100	1.000000 ----- 100

To compute Spearman rank-order correlations, simply select **Spearman rank-order** in the **Method** dropdown and choose the statistics you wish to compute. Spearman rank-order covariances is a nonparametric measure of correlation that may be thought of as ordinary covariances applied to rank transformed data. Here we display three Spearman results, **Correlation**, **Probability**, and **Number of cases**, arranged in multiple tables:

Note that the multiple table results make it easier to compare correlations across variables, but more difficult to relate a given correlation to the corresponding probability and number of cases.

A third major class of measures of association is based on Kendall's tau (see "Kendall's Tau" on page 581). Briefly, Kendall's tau for two variables is based on the number of concordances and discordances between the orderings of the variables for all possible comparisons of observations. If the number of concordances and discordances are roughly the same, there is no association between the variables, relatively large numbers of concordances suggest a positive relationship between the variables, and conversely for relatively large numbers of discordances.

Here, we display output in list format, showing **Kendall's tau-b**, **Kendall's D & C**, and **Probability**:

The results are similar to those obtained from ordinary and Spearman correlations, though the tau-b measures of association are somewhat lower than their counterparts.

Sample

EViews will initialize the edit field with the current workfile sample, but you may modify the entry as desired.

By default, EViews will perform listwise deletion when it encounters missing values so that all statistics are calculated using the same observations. To perform pairwise deletion of missing values, simply uncheck the **Balanced sample (listwise deletion)** checkbox. Pairwise calculations will use the maximum number of observations for each calculation.

Note that this option will be ignored when performing partial analysis since the latter is only performed on balanced samples

Partial Analysis

A partial covariance is the covariance between two variables while controlling for a set of conditioning variables.

To perform partial covariance analysis in EViews, simply enter a list of conditioning variables in the partial analysis edit field. EViews will automatically balance the sample, compute the statistics and display the results. Partial covariances or correlations will be computed for each pair of analysis variables, controlling for all of the variables in the conditioning set.

Consider the example from Matthews (2000) in which we consider the Pearson correlation between the number of stork breeding pairs and the number of births in 17 European countries. The data are provided in the workfile "Storks.WF1".

The unconditional correlation coefficient of 0.62 for the STORKS and BIRTH_RATE variables is statistically significant, with a p-value of about 0.008, indicating that the numbers of storks and the numbers of babies are correlated.

While some stork lovers may wish to view this correlation as indicative of a real relationship, others might argue that the positive correlation is spurious. One possible explanation lies in the existence of confounding variables or factors which are related to both the stork population and the number of births. Two possible factors are the population and area of the country.

To perform the analysis conditioning on the country area, select the statistics you wish to display, enter AREA in the partial analysis edit field, and press **OK**. The partial correlation falls to 0.27, with a statistically insignificant p-value of about 0.31.

Options

EViews provides a variety of other options for calculating your measures of association and for saving results.

Weighting

When you specify weighting, you will be prompted to enter a different weighting method and the name of a weight series. There are five different weight choices: frequency, variance, standard deviation, scaled variance, and scaled standard deviation.

EViews will compute weighted means and variances using the specified series and weight method. In each case, observations are weighted using the weight series; the different weight choices correspond to different functions of the series and different numbers of observations.

See "Weighting," beginning on page 582 for details.

Degrees-of-freedom Correction

You may choose to compute covariances using the maximum likelihood estimator or using the unbiased (degree-of-freedom corrected) formula. By default, EViews computes the ML estimates of the covariances.

When you check **d.f. corrected covariances**, EViews will compute the covariances by dividing the sums-of-squared cross-products by the number of observations n less the number of conditioning elements k, where k equals the number of conditioning variables, including the mean adjustment term, if present. For example, if you compute ordinary covariances conditional on Z_1, Z_2, Z_3, the divisor will be $n - 4$.

Multiple Comparison Adjustments

You may adjust your probability values for planned multiple comparisons using Bonferroni or Dunn-Sidak adjustments. In both of these approaches you employ a conservative approach to testing by adjusting the level of significance for each comparison so that the overall error does not exceed the nominal size.

The Bonferroni adjustment sets the elementwise size to

$$\alpha'' = \alpha / m \tag{12.1}$$

where α is the specified (overall) size of the tests, and m is the number of tests performed. For the Dunn-Sidak adjustment, we set the elementwise size to

$$\alpha' = 1 - (1 - \alpha)^{1/m} \tag{12.2}$$

For details, see Sokal and Rohlf (1995, p. 228–240).

Saved Results

You may place the results of the covariance analysis into symmetric matrices in the workfile by specifying a **Saved results basename**. For each requested statistic, EViews will save the results in a sym matrix named by appending an identifier ("cov," "corr," "sscp," "tstat," "prob," "taua," "taub," "score" (Kendall's score), "conc" (Kendall's concurrences), "disc" (Kendall's discordances), "cases," "obs," "wgts") to the basename.

For example, if you request ordinary correlations, probability values, and number of observations, and enter "MY" in the **Saved results basename** edit field, EViews will output three sym matrices MYCORR, MYPROB, MYOBS containing the results. If objects with the specified names already exist, you will be prompted to replace them.

Details

The following is a brief discussion of computational details. For additional discussion, see Johnson and Wichern (1992), Sheskin (1997), Conover (1980), and Kendall and Gibbons (1990).

Ordinary and Uncentered Covariances

The sums-of-squared cross-products are computed using

$$SSCP(X, Y) = \sum_i (X_i - \hat{\mu}_X)(Y_i - \hat{\mu}_Y) \tag{12.3}$$

where $\hat{\mu}_X$ and $\hat{\mu}_Y$ are the estimates of the means. For uncentered calculations, the mean estimates will be set to 0.

The covariances are computed by dividing the SSCP by the number of observations with or without a degrees-of-freedom correction:

$$\hat{\sigma}(X, Y) = \frac{\sum_i (X_i - \hat{\mu}_X)(Y_i - \hat{\mu}_Y)}{n - k} \tag{12.4}$$

where n is the number of observations associated with the observed X, Y pairs, and k is a degree-of-freedom adjustment term. By default EViews uses the ML estimator so that $k = 0$, but you may perform a degrees-of-freedom correction that sets k equal to the number of conditioning variables (including of the mean adjustment term, if present).

The correlation between the variables X and Y is computed from the following expression:

$$\hat{\rho}(X, Y) = \frac{\hat{\sigma}(X, Y)}{(\hat{\sigma}(X, X) \cdot \hat{\sigma}(Y, Y))^{1/2}} \tag{12.5}$$

It is worth reminding you that in unbalanced designs, the numbers of observations used in estimating each of the moments may not be the same.

Spearman Rank-order Covariances

Spearman covariances are a nonparametric measure of association that is obtained by computing ordinary covariances on ranked data, where ties are handled using averaging. To compute the Spearman rank-order covariances and correlations, we simply convert the data to ranks and then compute the *centered* ordinary counterparts.

Textbooks often provide simplified expressions for the rank correlation in to the case where there are no ties. In this case, Equation (12.5) simplifies to:

$$\rho(X, Y) = 1 - \frac{6\sum_i (R(X_i) - R(Y_i))^2}{n(n^2 - 1)} \tag{12.6}$$

where R returns the rank of the observation.

Kendall's Tau

Kendall's tau is a nonparametric statistic that, like Spearman's rank-order statistic, is based on the ranked data. Unlike Spearman's statistic, Kendall's tau uses only the relative orderings of ranks and not the numeric values of the ranks.

Consider the ranked data for any two observations i and j. We say that there is a *concordance* in the rankings if the relative orderings of the ranks for the two variables across observations are the same: $R(X_i) > R(X_j)$ and $R(Y_i) > R(Y_j)$ or $R(X_i) < R(X_j)$ and $R(Y_i) < R(Y_j)$. Conversely, we say that there is a *discordance* if the ordering of the X ranks differs from the ordering of the Y ranks: $R(X_i) > R(X_j)$ and $R(Y_i) < R(Y_j)$ or $R(X_i) < R(X_j)$ and $R(Y_i) > R(Y_j)$. If there are ties in the ranks of either the X or the Y pairs, we say the observation is neither concordant or discordant.

Intuitively, if X and Y are positively correlated, the number of concordances should outnumber the number of discordances. The converse should hold if X and Y are negatively related.

We may form a simple measure of the relationship between the variables by considering Kendall's score S, defined as the excess of the concordant pairs, C, over the discordant pairs, D. S which may be expressed as:

$$S(X, Y) = \sum_{i < j} \text{sgn}(R(X_i) - R(X_j)) \cdot \text{sgn}(R(Y_i) - R(Y_j)) \tag{12.7}$$

where the sign function takes the values -1, 0, and 1 depending on whether its argument is negative, zero, or positive. Kendall's tau-a is defined as the average of the excess of the con-

cordant over the discordant pairs. There are $n(n-1)/2$ unique comparisons of pairs of observations that are possible so that:

$$\tau_a(X, Y) = S(X, Y)/q \qquad (12.8)$$

In the absence of tied ranks, $-1 \leq \tau_a \leq 1$, with $\tau_a = 1$ when all pairs are concordant and $\tau_a = -1$ when all pairs are discordant.

One disadvantage of τ_a is that the endpoint values -1 and 1 are not reached in the presence of tied ranks. Kendall's tau-b τ_b rescales τ_a by adjusting the denominator of τ_a to account for the ties:

$$\tau_b(X, Y) = \frac{S(X, Y)}{\sqrt{\left[\frac{n(n-1)}{2} - \sum_s \frac{t_s(t_s-1)}{2}\right]\left[\frac{n(n-1)}{2} - \sum_v \frac{u_v(u_v-1)}{2}\right]}} \qquad (12.9)$$

where t_s are the number of observations tied at each unique rank of X, and u_v are the number of observations tied at each rank of Y. This rescaling ensures that $\tau_b(X, X) = 1$. Note that in the absence of ties, the summation terms involving t and u equal zero so that $\tau_a = \tau_b$.

It is worth noting that computation of these measures requires $n(n-1)/2$ comparisons, a number which increases rapidly with n. As a result, textbooks sometimes warn users about computing Kendall's tau for moderate to large samples. EViews uses very efficient algorithms for this computation, so for practical purposes, the warning may safely be ignored. If you find that the computation is taking too long, pressing the ESC break key may be used to stop the computation.

Weighting

Suppose that our weight series Z has m individual *cases* denoted by z_i. Then the weights and number of observations associated with each of the possible weighting methods are given by:

Method	Weights: w_i	Observations: n
Frequency	z_i	$(\sum w_i)$
Variance	z_i	m
Std. Dev.	z_i^2	m
Scaled Variance	$m z_i / (\sum z_i)$	$(\sum w_i) = m$
Scaled Std. Dev.	$(m z_i / (\sum z_i))^2$	m

Frequency weighting is the only weighting allowed for Spearman's rank-order and Kendall's tau measures of association.

The weighted SSCP is given by

$$WSSCP(X, Y) \; = \; \sum_i w_i (X_i - \tilde{\mu}_X)(Y_i - \tilde{\mu}_Y) \tag{12.10}$$

where the X_i and Y_i are the original data or ranks (respectively), the $\tilde{\mu}$ are weighted means (or zeros if computing uncentered covariances), and the w_i are weights that are functions of the specified weight series.

If estimated, the weighted means of X and Y are given by:

$$\tilde{\mu}_X \; = \; \left(\sum_i w_i X_i\right) / n, \qquad \tilde{\mu}_Y \; = \; \left(\sum_i w_i Y_i\right) / n \tag{12.11}$$

where n is the number of observations.

The weighted variances are given by

$$\tilde{\sigma}(X, Y) \; = \; \frac{\displaystyle\sum_i w_i (X_i - \tilde{\mu}_X)(Y_i - \tilde{\mu}_Y)}{n - k} \tag{12.12}$$

and the weighted correlations by

$$\tilde{\rho}_{XY} \; = \; \frac{\tilde{\sigma}(X, Y)}{\left(\tilde{\sigma}(X, X) \cdot \tilde{\sigma}(Y, Y)\right)^{1/2}} \tag{12.13}$$

The weighted Kendall's tau measures are derived by implicitly expanding the data to accommodate the repeated observations, and then evaluating the number of concordances and discordances in the usual fashion.

Testing

The test statistics and associated p-values reported by EViews are for testing the hypothesis that a single correlation coefficient is equal to zero. If specified, the p-values will be adjusted using Bonferroni or Dunn-Sidak methods (see "Multiple Comparison Adjustments," on page 579).

For ordinary Pearson and Spearman correlations, the t-statistic is computed as

$$t \; = \; \frac{r\sqrt{n - k - 1}}{\sqrt{1 - r^2}} \tag{12.14}$$

where r is the estimated correlation, and k is the number of conditioning variables, including the implicit mean adjustment term, if necessary. The p-value is obtained from a t-distribution with $n - k - 1$ degrees-of-freedom (Sheskin, 1997, p. 545, 598).

In the leading case of centered non-partial correlations, $k = 1$, so the degrees-of-freedom is $n - 2$. For centered partial correlations, $k = k' + 1$ where k' is the number of non-redundant conditioning variables, so the degrees of freedom is given by $n - k' - 2$.

The test of significance for Kendall's tau is based on a normal approximation using the continuity corrected z-statistic (Kendall and Gibbons, 1990, p. 65–66):

$$z = \frac{S - \text{sgn}(S)}{\sqrt{var(S)}} \tag{12.15}$$

where the variance is given by:

$$var(S) = S1 + S2 + S3 \tag{12.16}$$

for

$$S1 = \frac{1}{18}\left(n(n-1)(2n+5) - \sum_s t_s(t_s-1)(2t_s+5) - \sum_v u_v(u_v-1)(2u_v+5)\right)$$

$$S2 = \frac{1}{9n(n-1)(n-2)}\left(\sum_s t_s(t_s-1)(t_s-2)\right)\left(\sum_v u_v(u_v-1)(u_v-2)\right) \tag{12.17}$$

$$S3 = \frac{1}{2n(n-1)}\left(\sum_s t_s(t_s-1)\right)\left(\sum_v u_v(u_v-1)\right)$$

where t_s are the number of observations tied at each unique rank of X and u_v are the number of observations tied at each rank of Y. In the absence of ties, Equation (12.16) reduces to the expression:

$$\text{var}(S) = \frac{1}{18}n(n-1)(2n+5) \tag{12.18}$$

usually provided in textbooks (*e.g.*, Sheskin, 1997, p. 633).

Probability values are approximated by evaluating the two-tailed probability of z using the standard normal distribution. Note that this approximation may not be appropriate for small sample sizes; Kendall and Gibbons (1990) suggest that the approximation is not generally recommended for $n < 30$ (in the untied case).

Significance level values are currently not provided for partial Kendall's tau.

Partial Analysis

Let $W = (X, Y)$ be the set of analysis variables, and let Z be the set of conditioning variables.

For the ordinary and Spearman rank-order calculations, the joint sums of squares and cross-products for the two sets of variables are given by:

$$S = \begin{bmatrix} S_{ZZ} & S_{ZW} \\ S_{WZ} & S_{WW} \end{bmatrix} \tag{12.19}$$

EViews conditions on the Z variables by calculating the partial SSCP using the partitioned inverse formula:

$$S_{WW|Z} = S_{WW} - S_{WZ}S_{ZZ}^{-1}S_{ZW} \tag{12.20}$$

In the case where S_{ZZ} is not numerically positive definite, Z is replaced by a subset of Z formed by sequentially adding variables that are not linear combinations of those already included in the subset.

Partial covariances are derived by dividing the partial SSCP by $n - k$; partial correlations are derived by applying the usual correlation formula (scaling the partial covariance to unit diagonals).

For Kendall's tau computations, the partitioned inverse is applied to the corresponding matrix of joint Kendall's tau values. The partial Kendall's tau values are obtained by applying the correlation formula to the partitioned inverse.

N-Way Tabulation

This view classifies the observations in the current sample into cells defined by the series in the group. You can display the cell counts in various forms and examine statistics for independence among the series in the group. **Select View/N-Way Tabulation…** which opens the tabulation dialog.

Many of the settings will be familiar from our discussion of one-way tabulation in "One-Way Tabulation" on page 419.

Group into Bins If

If one or more of the series in the group is continuous and takes many distinct values, the number of cells becomes excessively large. This option provides you two ways to automatically bin the values of the series into subgroups.

- **Number of values** option bins the series if the series takes more than the specified number of distinct values.
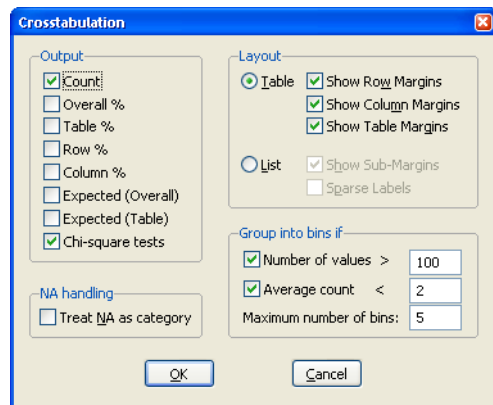
- **Average count** option bins the series if the average count for each distinct value of the series is less than the specified number.

- **Maximum number of bins** specifies the approximate maximum number of sub-groups to bin the series. The number of bins may be chosen to be smaller than this number in order to make the bins approximately the same size.

The default setting is to bin a series into approximately 5 subgroups if the series takes more than 100 distinct values or if the average count is less than 2. If you do not want to bin the series, unmark both options.

NA Handling

By default, EViews drops observations from the contingency table where any of the series in the group has a missing value. **Treat NA as category** option includes all observations and counts NAs in the contingency table as an explicit category.

Layout

This option controls the display style of the tabulation. The **Table** mode displays the categories of the first two series in $r \times c$ tables for each category of the remaining series in the group.

The **List** mode displays the table in a more compact, hierarchical form. The **Sparse Labels** option omits repeated category labels to make the list less cluttered. Note that some of the conditional χ^2 statistics are not displayed in list mode.

Output

To understand the options for output, consider a group with three series. Let (i, j, k) index the bin of the first, second, and third series, respectively. The number of observations in the (i, j, k)-th cell is denoted as n_{ijk} with a total of $N = \sum_i \sum_j \sum_k n_{ijk}$ observations.

- **Overall%** is the percentage of the total number of observations accounted for by the cell count.

- **Table%** is the percentage of the total number of observations in the conditional table accounted for by the cell count.

- **Row%** is the percentage of the number of observations in the row accounted for by the cell count.

- **Column%** is the percentage of the number of observations in the column accounted for by the cell count.

The *overall* expected count in the (i, j, k)-th cell is the number expected if *all* series in the group were independent of each other. This expectation is estimated by:

$$\hat{n}_{ijk} = \left(\sum_i n_{ijk^*}/N\right)\left(\sum_j n_{ijk^*}/N\right)\left(\sum_k n_{ijk^*}/N\right)N. \tag{12.21}$$

The *table* expected count $\tilde{n}_{ijk}$ is estimated by computing the expected count for the conditional table. For a given table, this expected value is estimated by:

$$\tilde{n}_{ijk^*} = \left(\sum_i n_{ijk^*}/N_{k^*}\right)\left(\sum_j n_{ijk^*}/N_{k^*}\right)N_{k^*} \tag{12.22}$$

where N_{k*} is the total number of observations in the $k*$ table.

Chi-square Tests

If you select the **Chi-square tests** option, EViews reports χ^2 statistics for testing the independence of the series in the group. The test statistics are based on the distance between the actual cell count and the count expected under independence.

- **Overall (unconditional) independence among all series in the group**. EViews reports the following two test statistics for overall independence among all series in the group:

$$\text{Pearson } \chi^2 = \sum_{i,j,k} \frac{(\tilde{n}_{i,j,k} - n_{i,j,k})^2}{\hat{n}_{i,j,k}}$$

$$\text{Likelihood ratio} = 2\sum_{i,j,k} n_{i,j,k} \, \log\!\left(\frac{n_{i,j,k}}{\hat{n}_{i,j,k}}\right) \tag{12.23}$$

where n_{ijk} and $\tilde{n}_{ijk}$ are the actual and overall expected count in each cell. Under the null hypothesis of independence, the two statistics are asymptotically distributed χ^2 with $IJK - (I-1) - (J-1) - (K-1) - 1$ degrees of freedom where I, J, K are the number of categories for each series.

These test statistics are reported at the top of the contingency table. For example, the top portion of the tabulation output for the group containing LWAGE, UNION, and MARRIED in the workfile "Cps88.WF1" shows:

Tabulation of LWAGE and UNION and MARRIED
Date: 08/18/09   Time: 15:15
Sample: 1 1000
Included observations: 1000

Tabulation Summary

Variable	Categories
LWAGE	5
UNION	2
MARRIED	2
Product of Categories	20

Test Statistics	df	Value	Prob
Pearson X2	13	174.5895	0.0000
Likelihood Ratio G2	13	167.4912	0.0000

WARNING: Expected value is less than 5 in 40.00% of cells (8
   of 20).

The three series LWAGE, UNION, and MARRIED, have $I = 5$, $J = 2$, and $K = 2$ categories, respectively. Note the WARNING message: if there are many cells with expected value less than 5, the small sample distribution of the test statistic under the null hypothesis may deviate considerably from the asymptotic χ^2 distribution.

- **Conditional independence between series in the group**. If you display in table mode, EViews presents measures of association for *each* conditional table. These measures are analogous to the correlation coefficient; the larger the measure, the larger the association between the row series and the column series in the table. In addition to the Pearson χ^2 for the table, the following three measures of association are reported:

$$\text{Phi coefficient} = \sqrt{\tilde{\chi}^2 / N} \tag{12.24}$$

$$\text{Cramers V} = \sqrt{\tilde{\chi}^2 / ((\min\{r, c\} - 1)N)} \tag{12.25}$$

$$\text{Contingency coefficient} = \sqrt{\tilde{\chi}^2 / (\tilde{\chi}^2 + N)} \tag{12.26}$$

where $\min(r, c)$ is the smaller of the number of row categories r or column categories c of the table, and $\tilde{N}$ is the number of observations in the table. Note that all three measures are bounded between 0 and 1, a higher number indicating a stronger relation between the two series in the table. While the correlation coefficient only measures the linear association between two series, these nonparametric measures are robust to departures from linearity.

Table 1: Conditional table for MARRIED=0:

		UNION		
Count		0	1	Total
	[0, 1)	0	0	0
	[1, 2)	167	8	175
LWAGE	[2, 3)	121	44	165
	[3, 4)	17	2	19
	[4, 5)	0	0	0
	Total	305	54	359

Measures of Association	Value
Phi Coefficient	0.302101
Cramer's V	0.302101
Contingency Coefficient	0.289193

Table Statistics	df	Value	Prob
Pearson X2	2	32.76419	0.0000
Likelihood Ratio G2	2	34.87208	0.0000

Note: Expected value is less than 5 in 16.67% of cells (1 of 6).

Bear in mind that these measures of association are computed for each two-way table. The conditional tables are presented at the top, and the unconditional tables are reported at the bottom of the view.

Tests of Equality

This view tests the null hypothesis that all series in the group have the same mean, median (distribution), or variance. All of these tests are described in detail in "Equality Tests by Classification" on page 411.

The **Common sample** option uses only observations for which none of the series in the group has missing values.

As an illustration, we demonstrate the use of this view to test for groupwise heteroskedasticity. Suppose we use data for seven countries over the period 1950–1992 and estimate a pooled OLS model (see Chapter 43. "Pooled Time Series, Cross-Section Data," on page 843 of *User's Guide II*). To test whether the residuals from this pooled regression are groupwise heteroskedastic, we test the equality of the variances of the residuals for each country.

First, save the residuals from the pooled OLS regression and make a group of the residuals corresponding to each country. This is most easily done by estimating the pooled OLS regression using a pool object and saving the residuals by selecting **Proc/Make Residuals** in the pool object menu or toolbar.
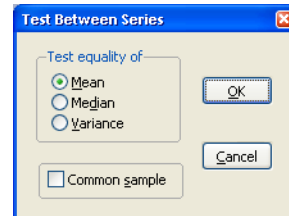
Next, open a group containing the residual series. One method is to highlight each residual series with the right mouse button, double click in the highlighted area and select **Open Group**. Alternatively, you can type show, followed by the names of the residual series, in the command window.

Select **View/Tests of Equality…**, and choose the **Variance** option in the **Test Between Series** dialog box.

Test for Equality of Variances Between Series
Date: 08/18/09   Time: 15:56
Sample: 1935 1954
Included observations: 20

Method	df	Value	Probability
Bartlett	9	496.0973	0.0000
Levene	(9, 190)	14.47574	0.0000
Brown-Forsythe	(9, 190)	9.566464	0.0000

Category Statistics

Variable	Count	Std. Dev.	Mean Abs. Mean Diff.	Mean Abs. Median Diff.
IAR	20	15.16693	12.02575	11.98550
ICH	20	42.72556	32.89755	29.64650
IDM	20	1.718661	1.474400	1.335500
IGE	20	48.58450	40.96800	39.70000
IGM	20	309.5746	212.1080	192.1700
IGY	20	14.88117	13.04590	12.91000
IIB	20	34.94722	27.96830	25.88800
IUO	20	18.31569	14.13405	13.80250
IUS	20	125.3994	96.11750	95.28500
IWH	20	19.11019	14.92465	14.56250
All	200	216.8753	46.56641	43.72855

Bartlett weighted standard deviation: 108.6848

Here we see results for testing the equality of variances for investment by various firms using data in the workfile "Grunfeld_baltagi_pool.WF1". The test statistics provide strong evidence of the presence of firmwise heteroskedasticity, decisively rejecting the null hypothesis of equal variances.

Principal Components

Principal components analysis models the variance structure of a set of observed variables using linear combinations of the variables. These linear combinations, or *components,* may be used in subsequent analysis, and the combination coefficients, or *loadings,* may be used in interpreting the components. While we generally require as many components as vari-

ables to reproduce the original variance structure, we usually hope to account for most of the original variability using a relatively small number of components.

We may, for example, have a very large number of variables describing individual health status that we wish to reduce to a manageable set. By forming linear combinations of the observed variables we may achieve data reduction by creating a handful of measures that describe overall health (*e.g.*, "strength," "fitness," "disabilities"). The coefficients in these linear combinations may be used to provide interpretation to the newly constructed health measures.

The principal components of a set of variables are obtained by computing the eigenvalue decomposition of the observed variance matrix. The first principal component is the unit-length linear combination of the original variables with maximum variance. Subsequent principal components maximize variance among unit-length linear combinations that are orthogonal to the previous components.

For additional details see Johnson and Wichtern (1992).

Performing Principal Components

EViews allows you to compute the principal components of the estimated correlation or covariance matrix of a group of series, and to display your results in a variety of ways. You may display the table of eigenvalues and eigenvectors, display line graphs of the ordered eigenvalues, and examine scatterplots of the loadings and component scores. Furthermore you may save the component scores and corresponding loadings to the workfile.

As an illustration, we again consider the stock price example from Johnson and Wichtern (1992) in which 100 observations on weekly rates of return for Allied Chemical, DuPont, Union Carbide, Exxon, and Texaco were examined over the period from January 1975 to December 1976 ("Stocks.WF1").

To perform principal components on these data, we open the group G1 containing the series and select **View/Principal Components...** to open the dialog:

The principal components dialog has two tabs. Here, we have selected the first tab, labeled **Components**. The second tab, labeled **Calculation,** controls the computation of the dispersion matrix from the series in the group. By default, EViews will perform principal components on the ordinary (Pearson) *correlation* matrix, but you may use the settings on this tab to modify the preliminary calculation. We will examine this tab in greater detail in "Covariance Calculation" on page 598.
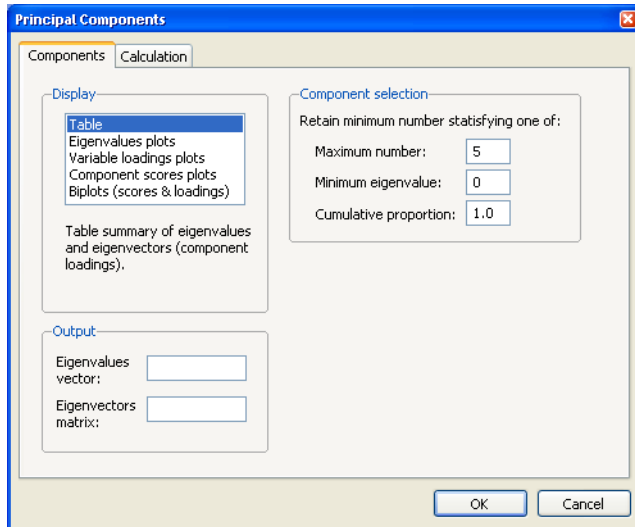
Viewing the Components

The **Components** tab is used to specify options for displaying the components or saving the eigenvalues and eigenvectors of the variances.

The **Display** box allows you to choose between showing the eigenvalues and eigenvectors in a tabular form, or displaying line graphs of the ordered eigenvalues, or scatterplots of the loadings, scores, or both (biplot). As you select different display methods, the remainder of the dialog will change to provide you with different settings.
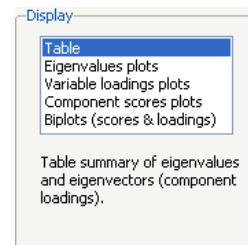
Table

In the figure above, the **Table** display setting is chosen. There are two sets of fields that you may wish to modify.

First, EViews provides you with three settings for controlling the number of components to be displayed; the number displayed will be the minimum number satisfying any of the criteria. The **Maximum number** setting should be self-explanatory. The **Minimum eigenvalue** instructs EViews to only show results for components where the eigenvalue (variance) exceeds a threshold. The **Cumulative proportion target** tells EViews to retain the first m components such that the sum of their proportion of the variances
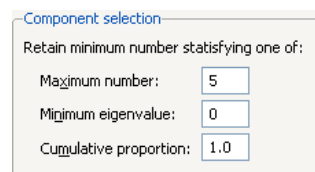
meets or exceeds the target proportion of the total variance. By default, the settings are chosen so that all components will be retained.

The **Output** fields allow you to save the eigenvalues and eigenvectors to the workfile. Simply enter a valid name in the corresponding field if you wish EViews to save your results.

If we leave the default settings as is and click **OK**, EViews will display a table of results.

Principal Components Analysis
Date: 08/18/09   Time: 15:58
Sample: 1 100
Included observations: 100
Computed using: Ordinary correlations
Extracting 5 of 5 possible components

Eigenvalues: (Sum = 5, Average = 1)

Number	Value	Difference	Proportion	Cumulative Value	Cumulative Proportion
1	2.856487	2.047368	0.5713	2.856487	0.5713
2	0.809118	0.269075	0.1618	3.665605	0.7331
3	0.540044	0.088697	0.1080	4.205649	0.8411
4	0.451347	0.108343	0.0903	4.656996	0.9314
5	0.343004	---	0.0686	5.000000	1.0000

Eigenvectors (loadings):

Variable	PC 1	PC 2	PC 3	PC 4	PC 5
ALLIED	0.463541	-0.240850	-0.613357	0.381373	-0.453288
DUPONT	0.457076	-0.509100	0.177900	0.211307	0.674981
UNION	0.469980	-0.260577	0.337036	-0.664098	-0.395725
EXXON	0.421677	0.525265	0.539018	0.472804	-0.179448
TEXACO	0.421329	0.582242	-0.433603	-0.381227	0.387467

Here we show the top two sections of the table. The header describes the sample of observations, the method used to compute the dispersion matrix, and information about the number of components retained (in this case, all five).

The next section summarizes the eigenvalues, showing the values, the forward difference in the eigenvalues, the proportion of total variance explained, *etc*. Since we are performing principal components on a correlation matrix, the sum of the scaled variances for the five variables is equal to 5. The first principal component accounts for 57% of the total variance (2.856/5.00 = 0.5713), while the second accounts for 16% (0.809/5.00 = 0.1618) of the total. The first two components account for over 73% of the total variation.

The second section describes the linear combination coefficients. We see that the first principal component (labeled "PC1") is a roughly-equal linear combination of all five of the stock returns; it might reasonably be interpreted as a general stock return index. The second principal component (labeled "PC2") has negative loadings for the three chemical firms (Allied,

du Pont and Union Carbide), and positive loadings for the oil firms (Exxon and Texaco). This loading appears to represent an industry specific component.

The third section of the output displays the calculated correlation matrix:

Ordinary correlations:

	ALLIED	DUPONT	UNION	EXXON	TEXACO
ALLIED	1.000000				
DUPONT	0.576924	1.000000			
UNION	0.508656	0.598384	1.000000		
EXXON	0.386721	0.389519	0.436101	1.000000	
TEXACO	0.462178	0.321953	0.425627	0.523529	1.000000

Eigenvalues Plots

You may elect to display line graphs of the ordered eigenvalues by selecting **Eigenvalues plots** in the **Display** portion of the main dialog. The dialog will change to offer you the choice of displaying plots of any of: the eigenvalues (scree plot), the eigenvalues difference, the cumulative proportion of variance explained. By default, EViews will only display the scree plot of ordered eigenvalues.

For the stock data, displaying the scree and cumulative proportion graphs yields the graph depicted here. The scree plot in the upper portion of the view shows the sharp decline between the first and second eigenvalues. Also depicted in the graph is a horizontal line marking the mean value of the eigenvalues (which is always 1 for eigenvalue analysis conducted on correlation matrices).
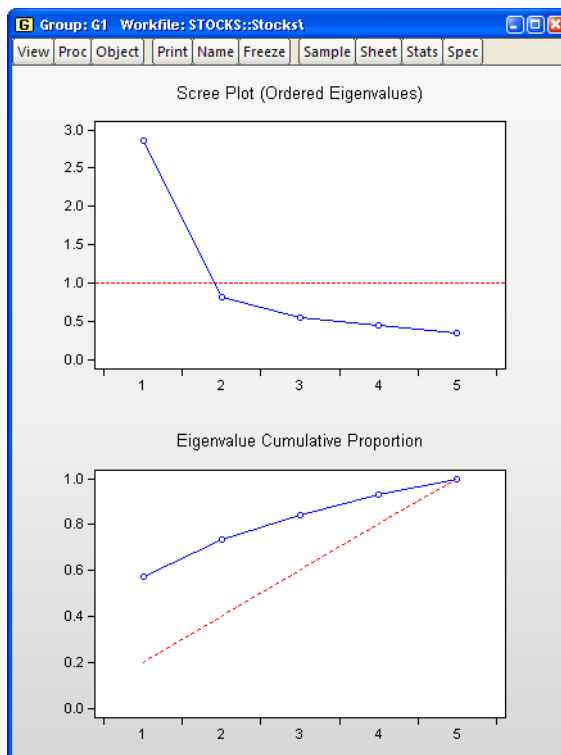
The lower portion of the graph shows the cumulative proportion of the total variance. As we saw in the table, the first two components account for about 73 % of the total variation. The diagonal reference line offers an alternative method of evaluating the size of the eigenvalues. The slope of the reference line may be compared with the slope of the cumulative propor-

tion; segments of the latter that are steeper than the reference line have eigenvalues that exceed the mean.

Other Graphs (Variable Loadings, Component Scores, Biplots)

The remaining three graphs selections produce graphs of the loadings (variables) and scores (observations): the variable loadings plots (**Variable loadings plot**) produce component-wise plots of the eigenvectors (factor loading coefficients), allowing you to visualize the composition of the components in terms of the original variables; the scores plot (**Component scores plot**) shows the actual values of the components for the observations in the sample; the biplot (**Biplots (scores & loadings)**) combines the loadings and scores plots in one display.

We continue our example by displaying the biplot graph since it includes the options for both the loadings and scores plots. If we select the **Biplots (scores and loadings)** entry, the right side of the dialog changes to provide additional plot options.

Components to Plot

The top right portion of the dialog, labeled **Components to plot,** is where you will provide the basic specification for the graphs that you want to display.

First, you must provide a list of components to plot. Here, the default setting "1 2" instructs EViews to place the first component on the x-axis and the second component on the y-axis. You may reverse the order of the axes by reversing the indices.
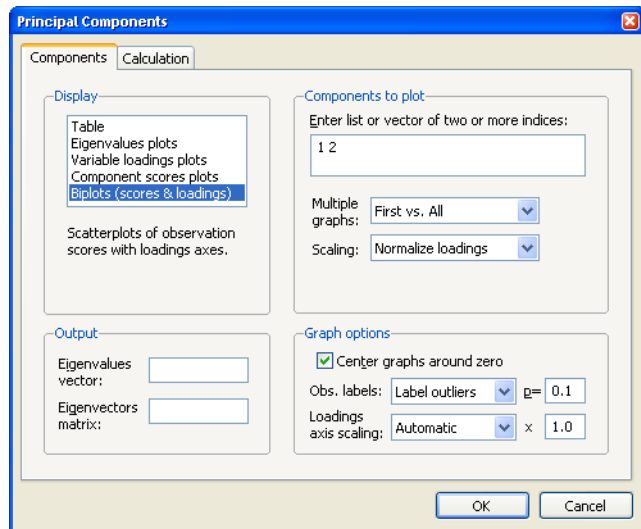
You may add indices for additional components. When more than two indices are provided, the **Multiple graphs** setting provides choices for how you wish to process the indices. You may elect to plot the first listed component against the remaining components (**First vs. All**), to use successive pairs of indices to form plots (**XY pairs**), or to plot each component against the others (**Lower triangular matrix**).

The **Scaling** options determine the weights to be applied to eigenvalues in the scores and the loadings (see "Technical Discussion," beginning on page 599 for details). By default, the

loadings are normalized so the observation scores have norms proportional to the eigenvalues (**Normalize loadings**). You may instead choose to normalize the scores instead of the loadings (**Normalize scores**) so that the observation score norms are proportional to unity, to apply symmetric weighting (**Symmetric weights**), or to specify a user-supplied loading weight (**User loading weight**).

In the latter three cases, you will be prompted to indicate whether you wish to adjust the results account for the sample size (**Adjust scores & loadings for sample size**). By default, EViews uses this setting and scales the loadings and scores so that the variances of the scores (instead of the norms) have the desired structure (see "Observation Scaling" on page 602). Setting this option may improve the interpretability of the plot. For example, when normalizing scores, the weight adjustment scales the results so that the Euclidean distances between observations are the Mahalanobis distances and the cosines of the angles between variables are the covariances.

Using the default settings and clicking on **OK**, EViews produces the view:

The component scores are displayed as circles and the variable loadings are displayed as lines from the origin with variable labels. The biplot clearly shows us that the first component has positive loadings for all five variables (the general stock return index interpretation). The second component has positive variable loadings for the energy stocks, and negative loadings for the chemical stocks; when the energy stocks do well relative to the chemical stocks, the second

specific component will be positive, and vice versa.
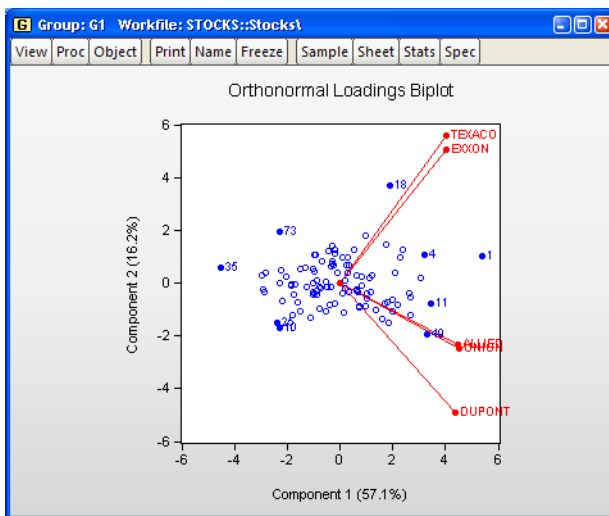
The scores labels show us that observation 3 is an outlier, with a high value for the general stock market index, and a relatively neutral value for the sector index. Observation 37 shows a poor return for the general market but is relatively sector neutral. In contrast, observation 20 is a period in which the overall market return was positive, with high returns to the energy sector relative to the chemical sector.

Graph Options

There are three additional options provided under **Graph options**. The first option is to **Center graphs around zero**. Unchecking this box will generally enlarge the graph within the frame at the expense of making it somewhat more difficult to quickly discern the signs of scores and loadings in a given dimension.
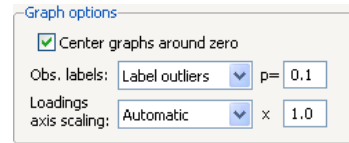
The **Obs. labels** dropdown allows you to choose the style of text labeling for observations. By default, EViews will **Label outliers**, but you may instead choose to **Label all obs.** or to display **Symbols only**. If you choose to label outliers, EViews will use a cutoff based on the specified probability value for the Mahalanobis distance of the observation from 0. The default is 0.1 so that labeled observations differ from the 0 with probability less than 0.1.

The last option, **Loadings axis scaling**, is available only for biplot graphs. Note that the observations and variables in a biplot will generally have very different data scales. Instead of displaying biplots with dual scales, EViews applies a constant scaling factor to the loadings axes to make the graph easier-to-read. **Loadings axis scaling** allows you to override the EViews default scale for the loadings in two distinct ways.

First, you may instruct EViews to apply a scale factor to the automatically chosen factor. This method is useful if you would like to stretch or shrink the EViews default axes. With the **Loadings axis scaling** set to **Automatic**, simply enter your desired adjustment factor. The automatically determined loadings will be scaled by this factor.

Alternatively, if you wish to assign an *absolute* scaling factor, select **User-specified** for the axis scaling, and enter your scale factor. The original loadings will be scaled by this factor.

Saving Component Scores

EViews provides easy-to-use tools for saving the principal components scores and scaled loadings matrices in the workfile. Simply select **Proc/Make Principal Components...** from the main group menu to display the dialog.

As with the main principal components view, the dialog has two tabs. The second tab controls the calculation of the dispersion matrix. The first describes the results that you wish to save.

The first option, **Scaling**, specifies the weights to be applied to eigenvalues in the scores and the loadings (see "Technical Discussion," beginning on page 599 for details). By default, EViews will save the scores associated with normalized loadings (**Normalize loadings**), but you may elect to save normalized scores (**Normalize scores**), equal weighted scores and loadings (**Symmetric weights**), or user weighted loadings (**User loading weight**). The default normalized loadings scores will have variances equal to the corresponding eigenvalues; the normalized scores will have unit variance.
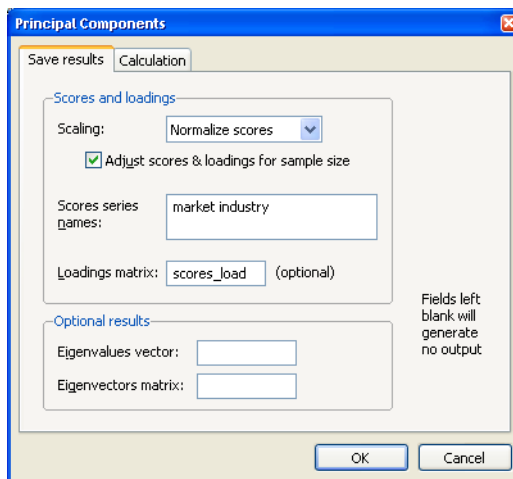
For the latter three selections, you are also given the option of adjusting the scores and loadings for the sample size. If **Adjust scores & loadings for sample size** is selected, the scores are scaled so that their variance rather than the sums-of-squares (norms) match the desired value. In this example, the sample variances of the component scores will equal 1.

Next, you should enter names for the score series, one name per component score you wish to save. Here we enter two component names, "Market" and "Industry," corresponding to the interpretation of our components given above. You may optionally save the loadings corresponding to the saved scores, eigenvalues, and eigenvectors to the workfile.

Covariance Calculation

The EViews routines for principal components allow you to compute the dispersion matrix for the series in a group in a number of ways. Simply click on the **Calculation** tab to display the preliminary calculation settings.

The **Type** dropdown allows you to choose between computing a **Correlation** or a **Covariance** matrix.

The **Method** dropdown specifies computation of **Ordinary**, **Ordinary (uncentered)**, **Spearman rank-order** or **Kendall's tau-a**, or **Kendall's tau-b** measures. The **Type** selection dropdown is not applicable if you select **Kendall's tau-a** or **Kendall's tau-b** as your method.

The remaining settings should be familiar from the covariance analysis view (). You may, for example, specify the sample of observations to be used and perform listwise exclusion of cases with missing values to balance the sample if necessary. Or you can perform partial and/or weighted analysis.
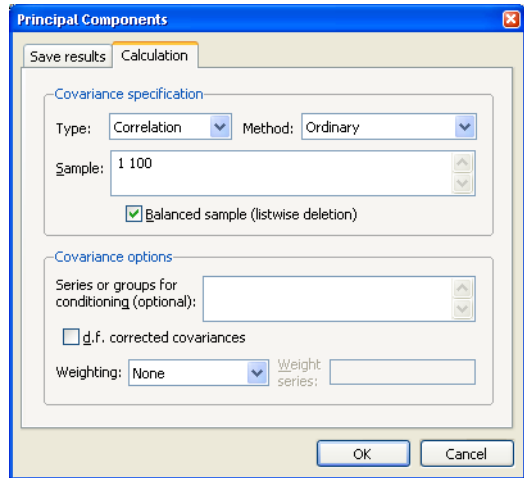
Note that component scores may not be computed for dispersion matrices estimated using Kendall's tau-a and tau-b.

Technical Discussion

From the singular value decomposition, we may represent a $(n \times p)$ data matrix Y of rank r as:

$$Y = UDV'$$ (12.27)

where U and V are orthonormal matrices of the left and right singular vectors, and D is a diagonal matrix containing the singular values.

More generally, we may write:

$$Y = AB'$$ (12.28)

where A is an $n \times r$, and B is a $p \times r$ matrix, both of rank r, and

$$
\begin{aligned}
A &= n^{\beta/2} UD^{1-\alpha} \\
B &= n^{-\beta/2} VD^{\alpha}
\end{aligned}
$$ (12.29)

so that $0 \le \alpha \le 1$ is a factor which adjusts the relative weighting of the left (observations) and right (variables) singular vectors, and the terms involving β are scaling factors where $\beta \in \{0, \alpha\}$. The basic options in computing the scores A and the corresponding loadings

B involve the choice of (loading) weight parameter α and (observation) scaling parameter β.

In the principal components context, let Σ be the cross-product moment (dispersion) matrix of Y, and perform the eigenvalue decomposition:

$$\Sigma = L\Lambda L' \tag{12.30}$$

where L is the $p \times p$ matrix of eigenvectors and Λ is the diagonal matrix with eigenvalues on the diagonal. The eigenvectors, which are given by the columns of L, are identified up to the choice of sign. Note that since the eigenvectors are by construction orthogonal, $L'L = LL' = I_m$.

We may set $U = YLD^{-1}$, $V = L$, and $D = (n\Lambda)^{1/2}$, so that:

$$
\begin{aligned}
A &= n^{\beta/2} YLD^{-\alpha} \\
B &= n^{-\beta/2} LD^{\alpha}
\end{aligned}
\tag{12.31}
$$

A may be interpreted as the *weighted principal components scores*, and B as the *weighted principal components loadings*. Then the scores and loadings have the following properties:

$$
\begin{aligned}
A'A &= n^{\beta} D^{-\alpha} L' Y' YLD^{-\alpha} = n^{\beta}(n\Lambda)^{-\alpha/2}(n\Lambda)(n\Lambda)^{-\alpha/2} = n^{\beta}(n\Lambda)^{1-\alpha} \\
B'B &= n^{-\beta} D^{\alpha} L'LD^{\alpha} = n^{-\beta}(n\Lambda)^{\alpha} \\
BB' &= n^{-\beta} LD^{2\alpha} L' = n^{-\beta} L(n\Lambda)^{\alpha} L'
\end{aligned}
\tag{12.32}
$$

Through appropriate choice of the weight parameter α and the scaling parameter β, you may construct scores and loadings with various properties (see "Loading Weights" on page 600 and "Observation Scaling" on page 602). EViews provides you with the opportunity to choose appropriate values for these parameters when displaying graphs of principal component scores and loadings and when saving scores and loadings to the workfile.

Note that when computing scores using Equation (12.33), EViews will transform the Y to match the data used in the original computation. For example, the data will be scaled for analysis of correlation matrices, and partialing will remove means and any conditioning variables. Similarly, if the preliminary analysis involves Spearman rank-order correlations, the data are transformed to ranks prior to partialing. Scores may not be computed for dispersion matrices estimated using Kendall's tau.

Loading Weights

At one extreme, we define the *normalized loadings* (also termed the *form,* or *JK*) decomposition where $\alpha = \beta = 0$. The scores formed from the normalized loadings decomposition will have variances equal to the corresponding eigenvalues. To see this, substituting into Equation (12.31), and using Equation (12.28) we have $Y = JK'$, where:

$$J = YL$$
$$K = L \tag{12.33}$$

From Equation (12.32), the scores J and loadings K have the norms:

$$J'J = n\Lambda$$
$$K'K = I_p \tag{12.34}$$

The rows of J are said to be in *principal coordinates*, since the norm of J is the diagonal matrix with the eigenvalues on the diagonal. The columns of K are in *standard coordinates* since K is orthonormal (Aitchison and Greenwood, 2002, p. 378). The JK specification has a *row preserving metric* (RPM) since the observations in J retain their original scale.

At the other extreme, we define the *normalized scores* (also referred to as the *covariance* or *GH)* decomposition where $\alpha = 1$. Then we may write $Y = GH'$ where:

$$G = n^{\beta/2} YLD^{-1}$$
$$H = n^{-\beta/2} LD \tag{12.35}$$

Evaluating the norms using Equation (12.32), we have:

$$G'G = n^{\beta} I_p$$
$$H'H = n^{-\beta}(n\Lambda) \tag{12.36}$$
$$HH' = n^{-\beta} L(n\Lambda)L' = n^{1-\beta}\Sigma$$

For this factorization, G is orthonormal (up to a scale factor) and the norm of H is proportional to the diagonal matrix with the n times the eigenvalues on the diagonal. Thus, the specification is said to favor display of the variables since the H loadings are in principal coordinates and the scores G are in standard coordinates (so that their variances are identical). The GH specification is sometimes referred to as the *column metric preserving* (CMP) specification.

In interpreting results for the GH decomposition, bear in mind that the Euclidean distances between observations are proportional to Mahalanobis distances. Furthermore, the norms of the columns of H are proportional to the factor covariances, and the cosines of the angles between the vectors approximate the correlations between variables.

Obviously, there are an infinite number of alternative scalings lying between the extremes. One popular alternative is to weight the scores and the loadings equally by setting $\alpha = 0.5$: This specification is the *SQ* or *symmetric biplot,* where $Y = SQ'$:

$$S = n^{\beta/2} YLD^{-1/2}$$
$$Q = n^{-\beta/2} LD^{1/2} \tag{12.37}$$

Evaluating the norms of the scores S and loadings Q, we have:

$$S'S = n^{\beta}(n\Lambda)^{1/2}$$
$$Q'Q = n^{-\beta}(n\Lambda)^{1/2}$$

(12.38)

so that the norms of both the observations and the variables are proportional to the square roots of the eigenvalues.

Observation Scaling

In the decompositions above, we allow for observation scaling of the scores and loadings parameterized by β. There are two obvious choices for the scaling parameter β.

First, we could ignore sample size by setting $\beta = 0$ so that:

$$A'A = (n\Lambda)^{1-\alpha}$$
$$B'B = (n\Lambda)^{\alpha}$$

(12.39)

With no observation adjustment, the norm of the scores equals $(n\Lambda)^{1-\alpha}$, the variance of the scores equals $\Lambda^{1-\alpha}/n^{\alpha}$, and the norm of the variables equals n^{α} times the eigenvalues raised to the α power. Note that the observed *variance* of the scores is not equal to, but is instead proportional to $\Lambda^{1-\alpha}$, and that the norm of the loadings is only proportional to Λ^{α}.

Alternately, we may set $\beta = \alpha$, yielding:

$$A'A = n^{\alpha}(n\Lambda)^{1-\alpha} = n\Lambda^{1-\alpha}$$
$$B'B = n^{-\alpha}(n\Lambda)^{\alpha} = \Lambda^{\alpha}$$

(12.40)

With this sample size adjustment, the variance of the scores equals $\Lambda^{1-\alpha}$ and the norm of the variables equals Λ^{α}.

Gabriel (1971), for example, recommends employing a *principal components decomposition* for biplots that sets $\beta = \alpha = 1$. From Equation (12.32) the relevant norms are given by:

$$G'G = nI_p$$
$$H'H = \Lambda$$
$$HH' = \Sigma$$

(12.41)

By performing observation scaling, the scores are normalized so that their variances (instead of their norms) are equal to 1. Furthermore the Euclidean distances between points are equal to the Mahalanobis distances (using Σ^{-1}), the norms of the columns of H are equal to the eigenvalues, and the cosines of the angles between the vectors equal the correlations between variables. Without observation scaling, these results only hold up to a constant of proportionality.

By default, EViews performs observation scaling, setting $\beta = \alpha$. To remove this adjustment, simply uncheck the **Adjust scores & loadings for sample size** checkbox. Note that when EViews performs this adjustment, it employs the denominator from the original dis-

person calculation which will differ from n if any degrees-of-freedom adjustment has been applied.

Correlograms

Correlogram displays the autocorrelations and partial autocorrelations of the first series in the group. See "Correlogram" on page 420, for a description of the correlogram view.

Cross Correlations and Correlograms

This view displays the cross correlations of the *first two series in the group*. The cross correlations between the two series x and y are given by,

$$r_{xy}(l) = \frac{c_{xy}(l)}{\sqrt{c_{xx}(0)} \cdot \sqrt{c_{yy}(0)}}, \quad \text{where } l = 0, \pm 1, \pm 2, \ldots \quad (12.42)$$

and,

$$c_{xy}(l) = \begin{cases} \sum_{t=1}^{T-l} ((x_t - \bar{x})(y_{t+l} - \bar{y}))/T & l = 0, 1, 2, \ldots \\ \sum_{t=1}^{T+l} ((y_t - \bar{y})(x_{t-l} - \bar{x}))/T & l = 0, -1, -2, \ldots \end{cases} \quad (12.43)$$

As with the correlograms ("Correlogram" on page 420), the computed cross correlation is an asymptotically consistent approximation as the means used in the computation are the overall means.

Note that, unlike autocorrelations, cross correlations are not necessarily symmetric around lag 0.

The dotted lines in the cross correlograms are the approximate two standard error bounds computed as $\pm 2 / (\sqrt{T})$.

Long-run Covariance

You may compute estimates of the long-run covariance matrix of a group of series.

	Cross Correlogram of GDP and PR				

Date: 08/15/09   Time: 18:38
Sample: 1952Q1 1996Q4
Included observations: 180
Correlations are asymptotically consistent approximations

GDP,PR(-i)	GDP,PR(+i)	i	lag	lead
		0	0.9925	0.9925
		1	0.9796	0.9736
		2	0.9666	0.9549
		3	0.9534	0.9362
		4	0.9401	0.9177
		5	0.9266	0.8992
		6	0.9130	0.8807
		7	0.8992	0.8622
		8	0.8852	0.8435
		9	0.8711	0.8248
		10	0.8569	0.8062
		11	0.8426	0.7876
		12	0.8281	0.7692
		13	0.8134	0.7508
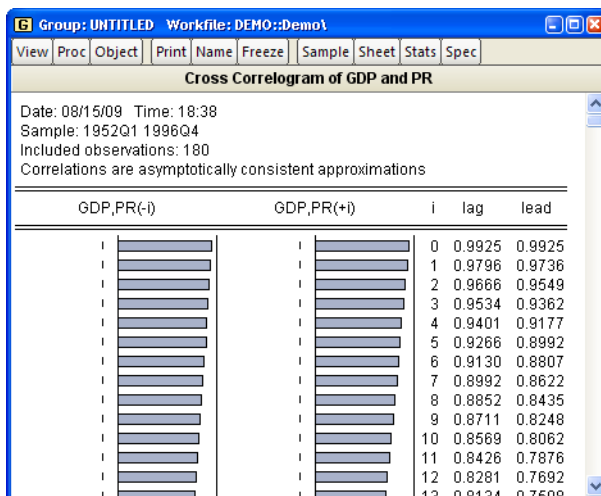
The discussion that follows assumes that you are familiar with the various issues and choices involved in computing a long-run covariance matrix. The technical description of the tools are given in Appendix F. "Long-run Covariance Estimation," on page 1115 of *User's Guide II*.

To compute the (possibly row-weighted) long-run covariance matrix of the series in a group, open the Group object and select **View/Long-run Covariance...** from the toolbar or main menu. If you wish to compute the results for a particular sample, you should set the workfile sample accordingly.

By default, EViews will estimate the symmetric long-run covariance matrix using a non-parametric kernel estimator with a Bartlett kernel and a real-valued bandwidth determined solely using the number of observations. The data will be centered (by subtracting off means) prior to computing the kernel covariance estimator, but no other pre-whitening will be performed. The results will only be displayed in the series or group window.

You may use the dialog to change these settings. The dialog (here we show the dialog for a group object) is divided into three sections. The **Basic options** section is used to describe the covariance that you wish to compute and to specify an output matrix, if desired. The **Whitening options** section is used to define pre-whitening or VARHAC options. The **Kernel options** section describes the non-parametric kernel settings.
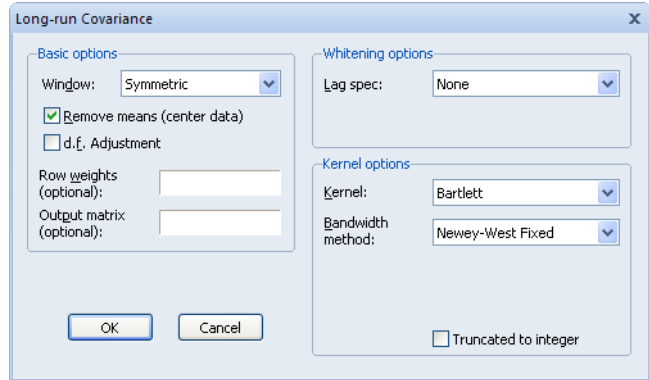
Basic Options

You may instruct EViews to compute a one-sided covariance by changing the **Window** dropdown from the **Symmetric** default to one of **Lower (lags)**, **Strict lower (lags)**, **Upper (leads)**, **Strict upper (leads)**. The "lower" methods only use covariances in which the rows correspond to contemporaneous values and the columns correspond to current and lagged data; the "upper" methods employ covariances where the columns correspond to current values and leads of the data. The "strict" long-run covariances exclude the contemporaneous covariance from the computation.
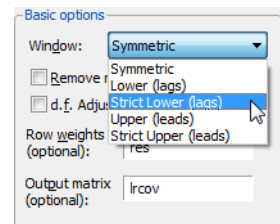
You may use the **Remove means** checkbox to indicate whether you wish to subtract off means (center your data) prior to computing the kernel covariance estimator. By default, EViews will remove the mean from each series in the group. To compute an uncentered long-run covariance, you should uncheck this box. If you do elect to center the data, you will be prompted for whether the covariance estimates should employ a **d.f. Adjustment** that accounts for the estimation of the mean values.
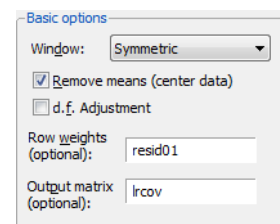
In addition, you may provide weights by entering a series expression in the **Row weights** edit field. Row weights are a convenient way of instructing EViews to compute the long-run covariance on data where the series in the group are weighted by a common element. A leading application occurs in the computation of White or Newey-West regression coefficient covariances, where the group contains the regressor data and the weights are the residuals.

For example, if you have the regressors series X1, X2, and X3 in your group, and the residuals are in the series RES, entering "RES" in the **Row weights** edit field instructs EViews to compute the long-run covariance of "X1*RES", "X2*RES", "X3*RES".

Lastly, you may elect to save your results in an EViews object. Simply enter a valid EViews object name in the **Output matrix** edit field. EViews will save symmetric long-run covariances in a sym object and one-sided covariances in a matrix object.

In panel workfiles, EViews will compute the Phillips and Moon (1999) *long-run average covariance matrix* obtained by averaging the long-run covariances across cross-sections. You may provide a name in the **Panel matrix** edit field to EViews to save a matrix containing the individual covariance estimates. Each row will contain the vec or vech of the results matrix for the corresponding cross-section.

Whitening Options

The **Whitening options** section is used to define options for computing the VAR used in computing VARHAC or pre-whitened kernel estimation.

By default, EViews does not whiten the data before computing the long-run covariance. To enable whitening, you should change the lag specification dropdown menu from **None** to one of **Fixed**, **Auto - AIC**, **Auto - Schwarz**, or **Auto - Hannan-Quinn**.
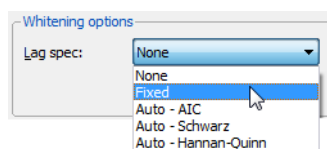
If you specify one of the automatic methods (**Auto - AIC**, **Auto - Schwarz**, **Auto - Hannan-Quinn**), you must specify the maximum number of lags in the **Maximum lags** edit field. You may provide the actual maximum value, or you may enter "*" to instruct EViews to use an observation-based maximum given by the integer portion of ($T^{1/3}$) as suggested by Den Haan and Levin (1997).
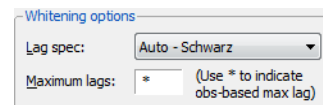
If you specify a **Fixed** lag specification, you should enter the value in the **Number of lags** edit field.
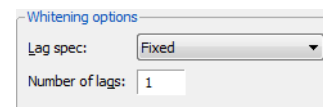
Kernel Options

The **Kernel options** section is used to define settings for estimation of the long-run covariance of the original or pre-whitened data.

Kernel Shape

The **Kernel** dropdown is used to define the method for computing the long-run covariance. Choosing **None** tells EViews to compute the long-run covariance using only the contemporaneous covariance. (The VARHAC methodology, for example, uses a combination of pre-whitening and the **None** setting.) The remaining choices specify the shape of the kernel function. There are a large number of pre-defined kernel shapes from which you may choose, as well as the option to provide the name of a **User-specified** custom kernel shape vector.
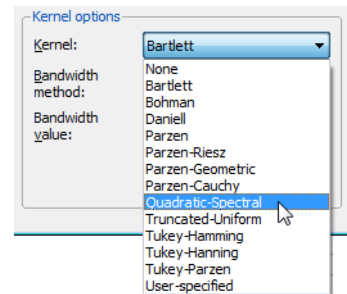
If you choose to use a user-specified kernel, EViews will prompt you for the name of a vector object containing the kernel weights. The vector should contain the weights for the (auto)covariances for lags from 0 to the lag truncation value. For example, to specify user-weights that match a Bartlett kernel with bandwidth value of 4.0, you should provide a vector of length 4 with values (1.0, 0.75, 0.5, 0.25) or equivalently a vector of length 5 with values (1.0, 0.75, 0.5, 0.25, 0.0).
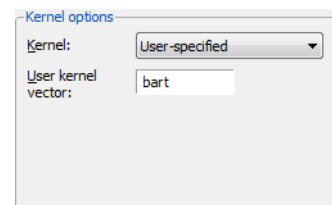
Bandwidth Specification

If you elect to use one of the pre-defined kernel shapes, EViews will prompt you to provide bandwidth information. By default, the kernel bandwidth is determined by the arbitrary observation-based **Newey-West Fixed** bandwidth rule. You may override the default choice by selecting either the **Andrews Automatic**, **Newey-West Automatic** settings for automatic optimal bandwidth selection, or the **User-specified** setting if you wish to provide a bandwidth value.
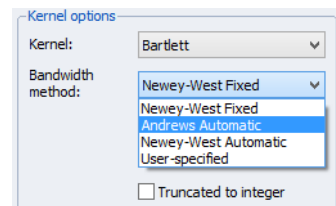
The **Newey-West Automatic** bandwidth selection method requires specification of a **Lag selection parameter**. You may enter a value for the number of lags in the edit field, or you may specify "*" to use an observation-based value given by the integer portion of $(4(T/100)^{r_n})$ where r_n depends on the properties of the selected kernel shape as given in "Kernel Function Properties" on page 1125 of *User's Guide II* (see "Newey-West Automatic Selection" on page 1121 of *User's Guide II* and Newey-West (1994) for discussion).
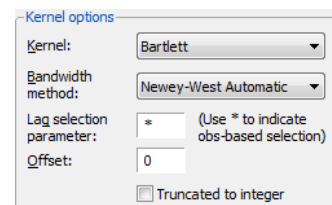
Both the **Newey-West Automatic** and the **Andrews Automatic** bandwidth selection methods allow you to adjust the chosen bandwidth by an offset, by entering a value in the **Offset** edit field. By default the offset is set to zero, which implies no adjustment will take place, however you may enter any positive or negative number to make an adjustment. This can be useful when trying to align the result given by EViews with results which use a different interpretation of the bandwidth. In such cases the most common required offset is a value of "1".

The **User-specified** bandwidth selection requires that you provide a bandwidth value. Simply enter the value as prompted in the **Bandwidth value** edit field.

Lastly, EViews does not restrict bandwidth values to be integers. Notably, the bandwidths obtained from the Newey-West fixed and the Andrews and Newey-West automatic methods are likely to be real valued. You may instruct EViews to use the integer portions of these bandwidths by checking the **Truncate to integer** checkbox.
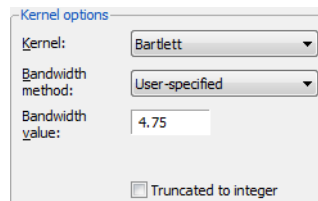
Examples

To illustrate the use of the long-run covariance view of a group, we employ the HAC covariance matrix example taken from Stock and Watson (2007, p. 620, Column 1). In this example Stock and Watson estimate the dynamic effect of cold weather on the price of orange juice by regressing the percent change in prices on a distributed lag of the number of freezing degree days and a constant, and obtain a HAC estimator for the coefficient covariance matrix using a Bartlett kernel estimator with lag truncation of 7. While the HAC covariances may be obtained directly from the EViews equation object (see "HAC Consistent Covariances (Newey-West)" on page 45 of *User's Guide II*), here we will estimate the long-run covariance estimate of $X_t\hat{\epsilon}_t$ manually using the group long-run covariance view, and will use this estimate to derive the HAC coefficient covariance matrix.

First, open the workfile "Stockwat_620.WF1", and notice that we have the equation object EQ_OLS which contains an estimated OLS equation (with conventional coefficient covariance and standard errors), and the group object XVARS which contains the regressors for the equation. We wish to compute the HAC coefficient covariance associated with the coefficient estimates in EQ_OLS using the group XVARS.

First, we need to obtain the residuals. Open EQ_OLS and click on **Proc/Make Residual Series** to display the **Make Residuals** dialog. Enter "RES" for the name of the residuals, then click on **OK** to save the regression residuals in the workfile.

Next, we will compute the symmetric long-run covariance of $X_t\hat{\epsilon}_t$. Double click on XVARS to open the group containing the regressors, then select **View/Long-run Covariance...** to display the dialog.

We must make a few changes to the default settings. To weight the regressors using the residuals, you must enter "RES" in the **Row weights** edit field. Uncheck the **Remove means** checkbox, since the $X_t \hat{\epsilon}_t$ should already have mean zero. Then, enter "LRCOV" in the **Output matrix** edit fields so that EViews will save the results in a sym matrix object.
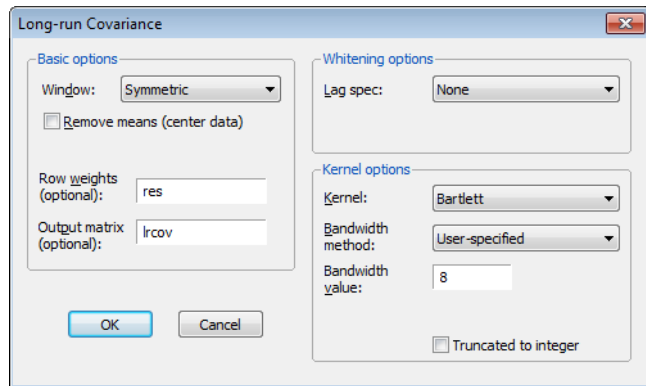
The Stock-Watson example uses a Bartlett kernel with no pre-whitening, so we leave the **Lag spec** at the default **None** setting and the **Kernel** shape choice at the default **Bartlett**. We should, however, change the **Bandwidth** method to **User-specified** and enter "8" in the edit field to match the 7 lags employed by Stock and Watson. Click on **OK** to compute and save the results.

The long-run covariance estimates are displayed in the group window and saved in the LRCOV matrix in the workfile. To obtain the d.f. corrected HAC coefficient covariances (see, for example, Hamilton (1994, p. 282)), we issue the commands:

```
sym xxinv = @inverse(@inner(xvars))
scalar obs = eq_ols.@regobs
scalar df = eq_ols.@regobs - eq_ols.@ncoef
sym hac_direct = (obs / df) * obs * xxinv * lrcov * xxinv
```

The first line computes $(X'X)^{-1}$. The next two lines get the number of observations and the number of degrees-of-freedom. The last line computes the d.f. corrected sandwich estimator using the long-run covariance in LRCOV as the estimator of $E(X'\hat{\epsilon}\hat{\epsilon}'X/T)$.

You may wish to verify that the estimate HAC_DIRECT matches the coefficient covariance matrix obtained using equation EQ_HAC (also provided in the workfile) estimated using the same HAC settings:

```
equation eq_hac.ls(cov=hac, covbw=8) 100*dlog(poj) xvars
sym hac_eq = eq_hac.@coefcov
```

Unit Root Test

This view carries out the Augmented Dickey-Fuller (ADF), GLS transformed Dickey-Fuller (DFGLS), Phillips-Perron (PP), Kwiatkowski, *et. al.* (KPSS), Elliot, Richardson and Stock (ERS) Point Optimal, and Ng and Perron (NP) unit root tests for whether the series in the group (or the first or second differences of the series) are stationary.

See "Panel Unit Root Testing" on page 617 of *User's Guide II* for additional discussion.

Cointegration Test

This view carries out the Johansen, Engle-Granger, or Phillips-Ouliaris test for whether the series in the group are cointegrated or not. See Chapter 48. "Cointegration Testing," on page 1023 of *User's Guide II*.

Granger Causality

Correlation does not necessarily imply causation in any meaningful sense of that word. The econometric graveyard is full of magnificent correlations, which are simply spurious or meaningless. Interesting examples include a positive correlation between teachers' salaries and the consumption of alcohol and a superb positive correlation between the death rate in the UK and the proportion of marriages solemnized in the Church of England. Economists debate correlations which are less obviously meaningless.

The Granger (1969) approach to the question of whether x causes y is to see how much of the current y can be explained by past values of y and then to see whether adding lagged values of x can improve the explanation. y is said to be Granger-caused by x if x helps in the prediction of y, or equivalently if the coefficients on the lagged x's are statistically significant. Note that two-way causation is frequently the case; x Granger causes y and y Granger causes x.

It is important to note that the statement "x Granger causes y" does not imply that y is the effect or the result of x. Granger causality measures precedence and information content but does not by itself indicate causality in the more common use of the term.

When you select the **Granger Causality** view, you will first see a dialog box asking for the number of lags to use in the test regressions. In general, it is better to use more rather than fewer lags, since the theory is couched in terms of the relevance of all past information. You should pick a lag length, l, that corresponds to reasonable beliefs about the longest time over which one of the variables could help predict the other.
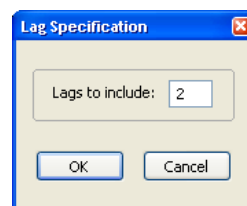
EViews runs bivariate regressions of the form:

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + \ldots + \alpha_l y_{t-l} + \beta_1 x_{t-1} + \ldots + \beta_l x_{-l} + \epsilon_t$$
$$x_t = \alpha_0 + \alpha_1 x_{t-1} + \ldots + \alpha_l x_{t-l} + \beta_1 y_{t-1} + \ldots + \beta_l y_{-l} + u_t$$

(12.44)

for all possible pairs of (x, y) series in the group. The reported F-statistics are the Wald statistics for the joint hypothesis:

$$\beta_1 = \beta_2 = \ldots = \beta_l = 0$$

(12.45)

for each equation. The null hypothesis is that x does *not* Granger-cause y in the first regression and that y does *not* Granger-cause x in the second regression.

We illustrate using data on consumption and GDP using the data in the workfile "Chow_var.WF1". The test results are given by:

Pairwise Granger Causality Tests
Date: 08/18/09   Time: 16:10
Sample: 1946Q1 1995Q4
Lags: 4

Null Hypothesis:	Obs	F-Statistic	Prob.
GDP does not Granger Cause CS	189	1.39156	0.2387
CS does not Granger Cause GDP		7.11192	2.E-05

For this example, we cannot reject the hypothesis that GDP does not Granger cause CS but we do reject the hypothesis that CS does not Granger cause GDP. Therefore it appears that Granger causality runs one-way from CS to GDP and not the other way.

If you want to run Granger causality tests with other exogenous variables (*e.g.* seasonal dummy variables or linear trends) or if you want to carry out likelihood ratio (LR) tests, run the test regressions directly using equation objects.

Panel causality tests are described in .

Label

This view displays the label information of the group. You can edit any of the field cells in the label view, except the **Last Update** cell which shows the date/time the group was last modified.

Name is the group object name as it appears in the workfile; you can rename your group by editing this cell. If you fill in the **Display Name** cell, EViews will use this name in some of the tables and graphs of the group view. Unlike object names, display names preserve case (allow upper and lower case letters) and may contain spaces.

You may click in the left-hand cell above the **Remarks** field to define a custom attribute for use in the workfile details view and for database search. To define a custom label, enter the

name of your attribute in the left-hand cell, and the attribute value in the right-hand cell. See "Adding Custom Attributes in the Label View," on page 117 for discussion.
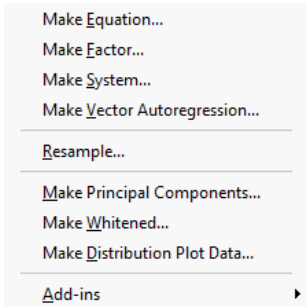
See Chapter 10. "EViews Databases," on page 317 for a discussion of the label fields and their use in database searches.

Group Procedures Overview

There are several procedures available for groups which generate new objects or data.

Make Equation...
Make Factor...
Make System...
Make Vector Autoregression...

Resample...

Make Principal Components...
Make Whitened...
Make Distribution Plot Data...

Add-ins ▶

Make Equation

The **Make Equation...** proc opens an **Equation Specification** dialog box with the first series in the group listed as the dependent variable and the remaining series as the regressors, including a constant term "C". You can modify the specification as desired. See Chapter 19. "Basic Regression Analysis," on page 5 of *User's Guide II* for details.

Make Factor

Make Factor... opens the factor analysis dialog with the correlation specification filled out with the series in the group. See Chapter 49. "Factor Analysis," on page 1043 of *User's Guide II* for details.

Make System

You may use the **Make System...** proc to make a system object. Simply select **Make System...** to open a system specification dialog. The dialog will be filled with the series in the group as dependent variables, and has edit fields that allow you to list common and equation specific regressors, instruments, and dependent variable transformations, if desired.
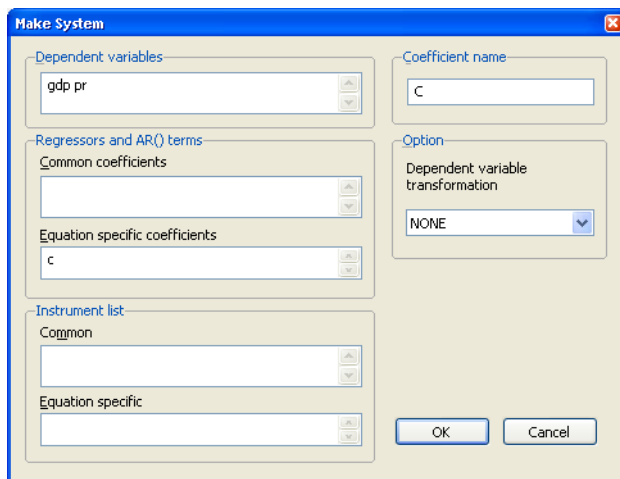
Make Vector Autoregression

Make Vector Autoregression… opens an **Unrestricted Vector Autoregression** dialog box, where all series in the group are listed as endogenous variables in the VAR. See Chapter 40. "Vector Autoregression and Error Correction Models," on page 687 of *User's Guide II* for a discussion of specifying and estimating VARs in EViews.

Resample

Resample… performs resampling on all of the series in the group. A description of the resampling procedure is provided in "Resample" on page 439.

Make Principal Components

Make Principal Components… is used to save various intermediate results from principal components analysis. See "Saving Component Scores" on page 597.

Make Whitened

You may use the **Make Whitened…** procedure from a group (or series) to use the results from a parametric VAR (or AR) model to filter the data. EViews offers whitening based on either a fixed lag specification, or an information criterion based auto-selected lag.
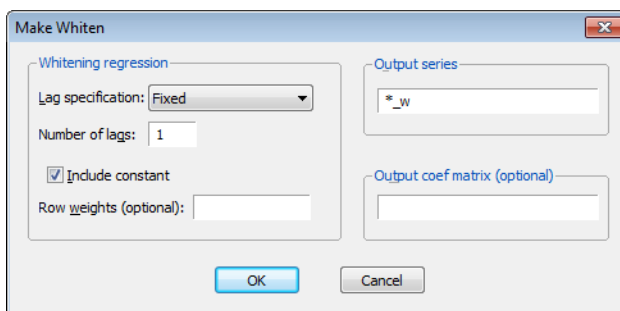
You may use the **Lag specification** dropdown to choose between a **Fixed**, user-specified number of lags on the VAR, or you may choose automatic lag-selection by choosing **Auto - AIC**, **Auto - Schwarz**, or **Auto - Hannan-Quinn**. In the **Fixed** case, EViews will prompt you for the number of lags to use; in the automatic selection case, you will be prompted for the maximum number of lags to use in selection. The default setting of "*" indicates use of an observation based maximum lag of $T^{1/3}$ as suggested by Den Haan and Levin (1997).

You may use the **Include constant** checkbox to indicate whether to include a constant in the VAR, and you may specify **Row weights** in the corresponding edit field.

By default, EViews will save the results in series with the names given by those in the group with "_W" appended to the end. Information on the whitening procedure will be included in the series label. If you would like to change the names, you can provide a list of series names or an alternate wildcard expression in the **Output series** edit field.

Lastly, you can instruct EViews to save a matrix containing the estimated coefficients from the VAR by entering a valid object name in the **Output coef matrix** edit field. The results will be saved in a matrix where each column represents the coefficients for the corresponding series in the group. The coefficients are ordered in the column by: constant (if present) followed by the lag 1 coefficient for each of the series in the group, in order, followed by the lag 2 coefficients, *etc.*

To illustrate the use of this tool, we use the data for industrial production (IP), money supply (M1), and the Treasury bill rate (TB3) in the workfile "Var1.WF1" used in Chapter 40. "Vector Autoregression and Error Correction Models," beginning on page 687. First set the sample to the entire workfile by entering the command

```
smpl @all
```

Next, create a group containing IP, M1, and TB3, name the group GROUP01, then click on **Proc/Make Whitened...** to display the whitening dialog. Set the **Lag specification** dropdown to **Auto - AIC**, and enter "ABC" in the **Output coef matrix** edit field. Click on **OK**.

obs	IP_W	M1_W	TB3_W
1959M01	NA	NA	NA
1959M02	NA	NA	NA
1959M03	NA	NA	NA
1959M04	NA	NA	NA
1959M05	NA	NA	NA
1959M06	NA	NA	NA
1959M07	NA	NA	NA
1959M08	-1.270547	0.299025	0.357210
1959M09	0.304379	-0.618412	0.913382
1959M10	-0.239094	-0.183514	0.217548
1959M11	0.329422	0.439830	0.142489
1959M12	2.204216	0.043035	0.588671
1960M01	0.263078	0.607564	-0.243191
1960M02	-0.891662	1.321452	-0.637435
1960M03	-0.484898	0.675346	-0.026923
1960M04	-0.380524	-0.366037	0.255600
1960M05			

EViews displays a new untitled group containing the three whitened series IP_W, M1_W and TB3_W. In addition, the 22×3 matrix object ABC contains the coefficients for the 7 lag VAR used to filter the data.

Note that the handling of lags differs slightly between the whitening procedure and VAR estimation. The whitening procedure does not use lagged values prior to the beginning of the computation sample, while the VAR estimator will use any available lags, even if they precede the estimation sample. In this case, the results for the whitening procedure will exactly match those of a VAR estimated for the sample from 1959M01 to 1995M04 since there are no valid presample values, but this correspondence may not hold for samples starting after the beginning of the workfile.

Make Distribution Plot Data

Make Distribution Plot Data... may be used to output the graph data used to construct auxiliary kernel, nearest neighbor, and quantile-quantile graphs. See "Auxiliary Graph Types" on page 704.
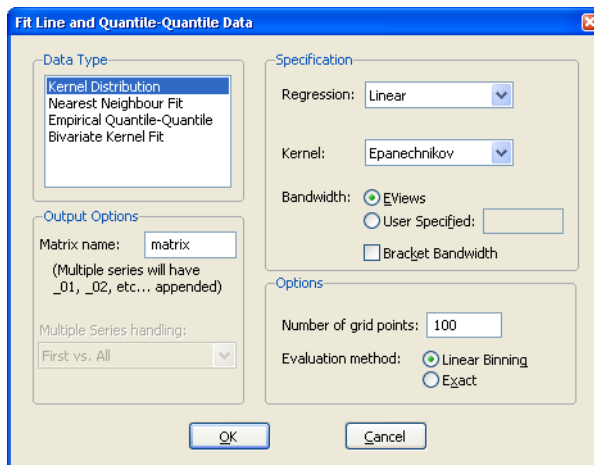
Simply select the desired graph type on the upper-left hand side of the dialog, set the relevant options on the right, choose a base name for the output, and if necessary, specify how to handle multiple sets of series in the group.

References

Aitchison, J. and Greenacre, M. J. (2002). Biplots of compositional data. *Applied Statistics*, 51, 375–392.

den Haan, Wouter J. and Andrew Levin (1997). "A Practitioner's Guide to Robust Covariance Matrix Estimation," Chapter 12 in Maddala, G. S. and C. R. Rao (eds.), *Handbook of Statistics Vol. 15, Robust Inference*, North-Holland: Amsterdam, 291-341.

Gabriel, K. R. (1971). The biplot-graphic display of matrices with application to principal component analysis. *Biometrika*, 58, 453-467.

Granger, C. W. J. (1969). "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods," *Econometrica,* 37, 424–438.

Hamilton, James D. (1994). *Time Series Analysis*, Princeton University Press.

Johnson, Richard A., and Dean W. Wichern (1992). *Applied Multivariate Statistical Analysis*, Third Edition, Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Kendall, Maurice, and Jean Dickinson Gibbons (1990). *Rank Correlation Methods*, Fifth Edition, New York: Oxford University Press.

Matthews, Robert (2000). "Storks Deliver Babies (p = 0.008)", *Teaching Statistics*, 22(2), 36–38.

Sheskin, David J. (1997). *Parametric and Nonparametric Statistical Procedures,* Boca Raton: CRC Press.

Sokal, Robert R. and F. James Rohlf (1995). *Biometry.* New York: W. H. Freeman and Company.

Chapter 13. Graphing Data

Constructing graphs from data is an important part of the process of data analysis and presentation. Results displayed in graphs are generally more concise and often more illuminating; one may be able to detect patterns and relationships in data that are not readily apparent from examining tables of summary statistics.

Accordingly, EViews provides an easy-to-use, full-featured set of tools for the graphical display of information. With EViews, you can quickly and easily display graphs of data, customize those graphs, and output the results so that they may be incorporated into your presentations.

There are many aspects of the graphing of data in EViews. This chapter describes the basics of graphing data in series and groups of series using the **View/Graph...** menu item. (Most of the graphs in this chapter may also be generated from vectors and matrices, but for brevity, we will speak mostly of series and groups).

Three types of graphs are described in this chapter:

- *Observation graphs* which show the data for each observation in the series or group. A line plot of the observations in a series or a scatterplot of observations for pairs of series in a group are examples of observation graphs.

- *Analytical graphs* where we display results obtained from analysis of the series or group data. You might, for example, show a histogram or a boxplot computed from the original data.

- *Auxiliary graphs* are analytical graphs that are not meant to stand alone, but rather are to be added to existing observation graphs. For example, we may display the linear regression or kernel fit line on top of a scatterplot. Strictly speaking, auxiliary graphs are not a graph type, but rather a modification of an existing observation plot.

A fourth class of graphs, *categorical graphs*, consists of observation or analytical graphs formed using data divided into categories defined by factor variables. Categorical graphs are described in Chapter 14. "Categorical Graphs," beginning on page 717.

We do not consider here the specialized series and group routines that produce graphical output. For example, views of an equation produce graphs of the equation forecasts and residuals. Similarly, views of a model object that show graphs of simulation results, and the views of a state space object that show estimated states or signals. These graphs are described in the context of the specific views and procedures.

The remainder of this chapter is structured as follows. The first section offers a quick overview of the process of constructing a series or group graph view. The next two sections describe the process of constructing graphs from series and groups in somewhat greater depth. Next, we describe a handful of the most commonly performed graph view customizations. The final section provides detail on individual graph types.

Quick Start

All of the graph features described in this chapter may be accessed by selecting **View/Graph...** from a series or graph (or vector or matrix) object menu, or by double clicking on a series or group graph view.

For example, let us consider the example workfile "Housedebt.WF1", which contains quarterly data on household debt and financial obligations from 1980 to 2006. The series DSR is an estimate of the debt service ratio representing the ratio of the total of mortgage and consumer debt payments to personal income.

We begin our example by opening DSR, selecting **View/Graph...** from the series menu and clicking on **OK** to accept the defaults. By default, EViews will display a simple (observation) line graph showing the data in the series.

Note that the titlebar of the window shows that we are looking at a view of the series DSR, and that the horizontal axis shows the dates associated with the current workfile sample range. The steep upward trend in the debt service ratio beginning in the early 1990s is readily apparent from the graph.
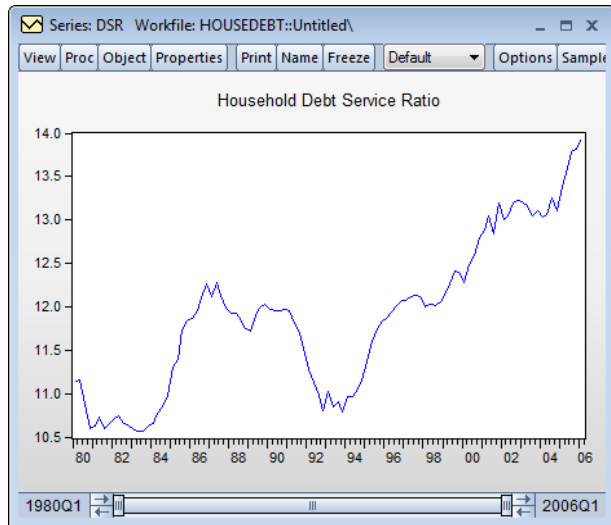
Note the presence of the slider bar at the bottom of the graph window. The slider, which appears in sample based graph views, allows you to control the visible sample in the graph window (see for further discussion). By default, the slider will be set to show the entire workfile sample.

We can more closely examine the start of the trend by hovering the cursor over the data point we wish to identify. EViews will popup a box in the graph showing the observation label and value. Here, we see that the 1992Q4 value for DSR is 10.80.
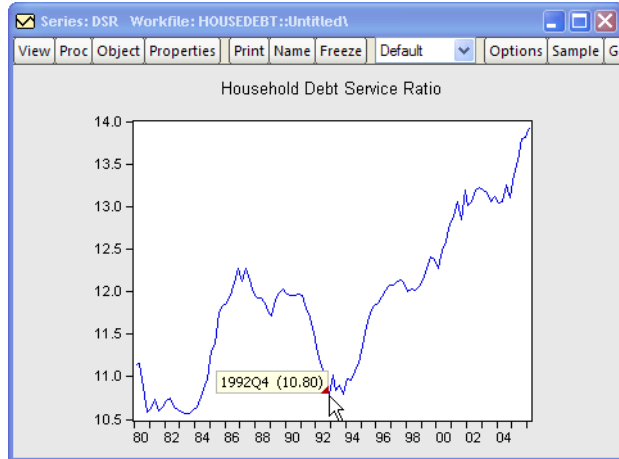
Alternately, if you hover over any point inside the graph frame, EViews will display the location of that point in the statusline located in the lower left-hand corner of your EViews window.

Typically, the flyover display will vanish when you move your cursor. To keep the flyover text with the graph, you may right mouse-click and select **Pin flyover**.

We may change the graph type to show a histogram of the data by double-clicking on the graph area or by selecting **View/ Graph...** to access the **Graph Options** dialog.
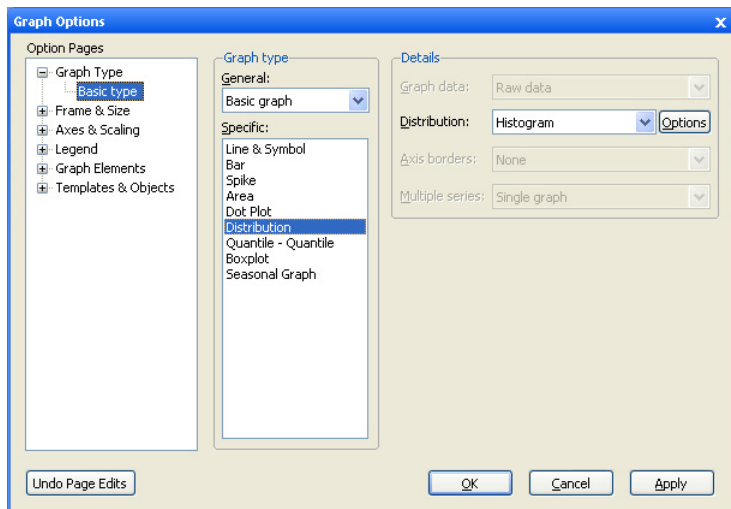
The graph options are broken into several categories, appearing on the left-hand side of the dialog. Make

certain that you are displaying the **Basic type** page, then change the specific type to **Distribution**. Notice that there is a dropdown on the right-hand side that offers you choices on which type of distribution graph to produce. Since we want to display the default histogram, simply click on **OK**.

The view of the series will change to show a histogram of the data in the series.

The histogram is an example of an analytic graph in which the data are plotted against a data scale, not the workfile observation scale; note that the horizontal axis in the graph no longer corresponds to dates in the workfile, but instead represents intervals of data values observed from data included in the workfile sample. In this case, we show bars whose heights represent frequencies for the data intervals depicted on the horizontal axis.
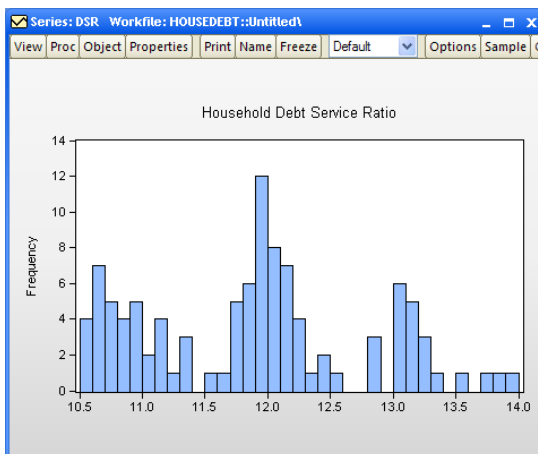
Similarly you may display graphs for a group of series by opening the group, selecting **View/Graph...** and choosing an appropriate type.

First, open the group object GROUP01 in the workfile, then select **View/Graph...** from the group menu and click on **OK** to accept the defaults. By default, EViews will display line graphs of the two series within a single graph frame.

As before, we may change the graph to a histogram view by double-clicking on the graph or selecting **View/ Graph...** to display the dialog.

Change the specific type to **Distribution** and click **OK** to accept the default settings.
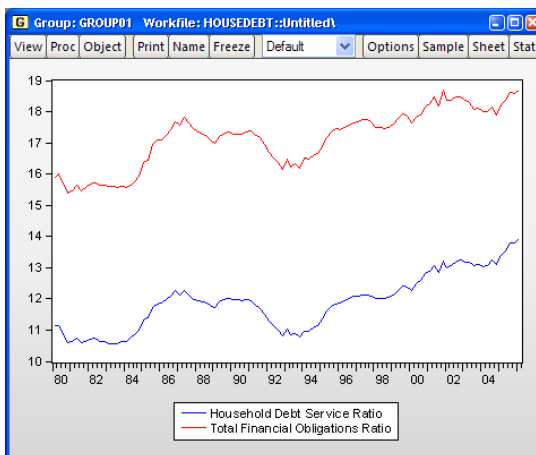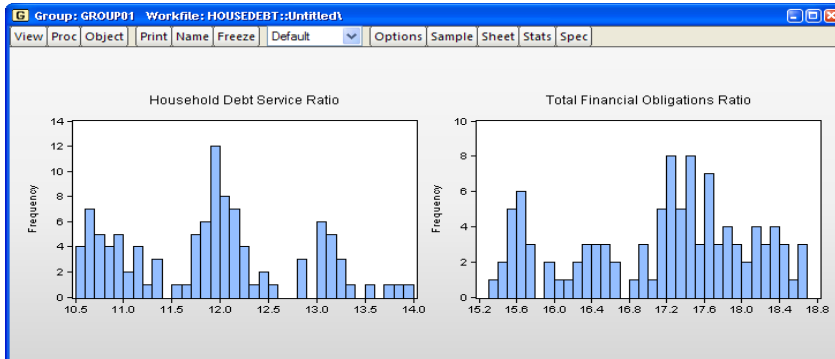
The group view will change to show histograms of the data in the two series. Note that by default, the histograms are displayed in separate frames (we have rearranged the graphs horizontally for presentation by right-clicking on the graph and selecting **Position and align graphs...**; see "Working with Multiple Graphs" on page 784). Note also that each of the frames has its own vertical and horizontal axis scale.

Displaying other graph views is generally just as easy. Most graphs can be displayed with a few mouse clicks and a couple of keystrokes. In general, you can simply open the series or group object, display the graph dialog, select the graph type, set a few options if necessary, and click **OK** to produce acceptable results.

Graphing a Series

Up to this point we have examined graph views for series and groups constructed using default settings. We now consider the process of displaying graph views of a single series in a bit more depth.

Our discussion focuses on the selection of a graph type and setting of associated options. We consider the general features of selecting a graph type for the series, not on the particulars associated with each graph type. Details on the individual graph types are provided in "Graph Types," beginning on page 669.

To display the graph view of a single series, you should first select **View/Graph...** from the series menu to display the **Graph Options** dialog.
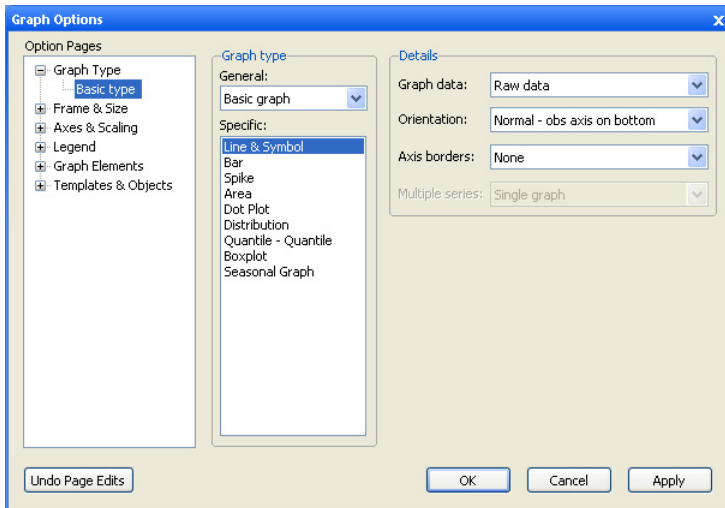
The **Graph Options** dialog has multiple pages that specify various settings for the graph view. The **Basic type** page depicted here is of central importance since it controls the graph you wish to display. The other dialog pages, which control various display characteristics of the graph, will be discussed below ("Basic Customization," beginning on page 648).

Choosing a Type

On the left-hand side of the **Basic type** page, you will see the **Graph type** section where you will specify the type of graph you wish to display.

First, the **General** dropdown menu allows you to switch between displaying a **Basic graph** of the data in the series, and displaying a **Categorical graph** constructed using the data divided into categories defined by factor variables. For now, we will stick to basic graphs; we defer the discussion of categorical graphs until Chapter 14. "Categorical Graphs," beginning on page 717.

Second, the **Specific** list box offers a list of the graph types that are available for use with this object. You may select a graph type by clicking on its name. The default graph type is a **Line & Symbol** plot.
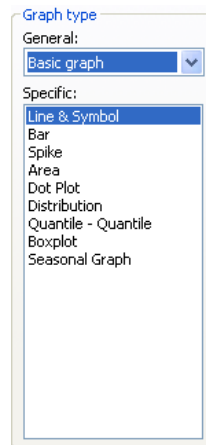
In most cases, these two settings are sufficient to identify the graph type. If, however, you select **Distribution** graph as your specific graph type, the right-hand side of the dialog will offer an option for choosing a specific distribution graph (note that the dropdown menu for **Orientation** has been replaced by one labeled **Distribution**). The **Options** button allows you to customize the selected distribution graph, or to display more than one distribution graph in the frame (see).
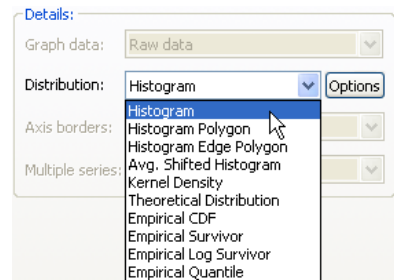
Similarly, if you select either **Quantile - Quantile** or **Seasonal Graph** as your specific type, the dialog will change to provide you with additional options. For theoretical quantile-quantile plots, you may use the **Options** button to specify a distribution or to add plots using different distributions. For seasonal graphs, there will be a dropdown menu controlling whether to panel or overlay the seasons in the graph.

Details

The right-hand side of the dialog offers various options that EViews collectively labels **Details**. The options that are available will change with different choices for the **Specific** graph type. We have, for example, already mentioned the sub-type settings that are available when you select **Distribution**, **Quantile-Quantile**, or **Seasonal Graph**. We now consider the remaining settings.

Graph Data

The **Graph Data** dropdown specifies the data to be used in observation graphs. By default, EViews displays observation graphs that use **Raw data**, meaning that every observation will be plotted. The dropdown allows you to compute summary statistics (**Means**, **Medians**, *etc.*) for your data prior to displaying an observation graph. (Note: if we display an observation graph type using summary statistics for the data, the graph is no longer an observation graph since it no longer plots observations in the workfile. Such a graph is, strictly speaking, a *summary graph* that uses an observation graph type.)

It is worth noting that a summary statistic graph for a single series shows a single data point. For example, we see here the Line & Symbol graph for the mean of the debt service ratio series DSR in our example workfile ("Housedebt.WF1"). Since we are working with a single series, the graph displays data for a single point which EViews displays as a symbol plot.

One will almost always leave this setting at **Raw data** in the basic single series case. As we will see, the **Graph data** option is most relevant when plotting data for multiple series, or when plotting data that have been categorized by some factor.
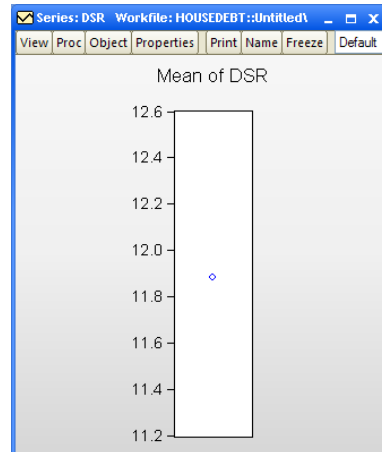
Orientation

The **Orientation** dropdown allows you to choose whether to display observation graphs with the observations along the horizontal or the vertical axis. By default, EViews displays the graph with the observations along the horizontal axis (**Normal - obs/time across bottom**), but you may elect to display them on the vertical axis (**Rotated - obs/time down left axis**).

For example, bar graphs are sometimes displayed in rotated form. Using the workfile "EUpop.WF1", we may display a rotated bar graph of the 1995 population (POP) for each of the 15 European Union members.

As an aside, it is worth mentioning here that graphs of this form, where observations have no particular ordering (unlike graphs involving time series data) sometimes order the bars by size.
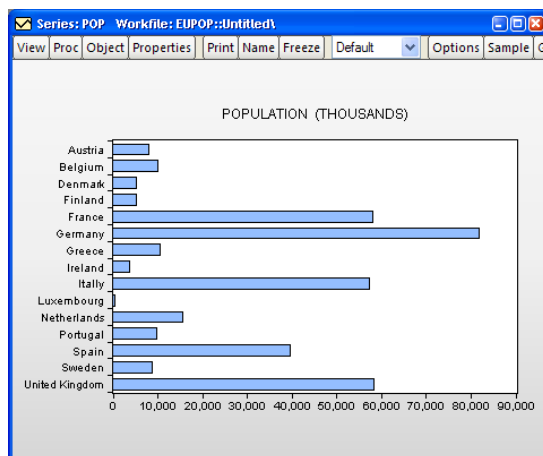
While EViews does not allow you to change the order of data in a series view, you can reorder the observations in a graph object (frozen series view).

While displaying the bar graph view, simply click on the **Freeze** button to create a graph object, then press the right mouse button and select **Sort...** to display the sort dialog. Sorting on the basis of values of POP in ascending order yields the graph depicted on the right. Note that sorting reorders the data in the graph object, not the data in the original series POP.
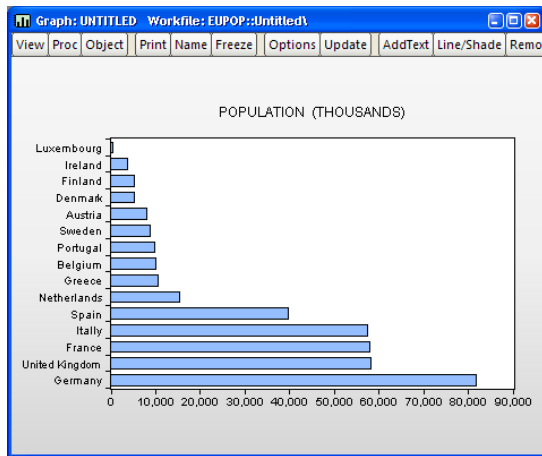
Frequency

When plotting a line graph for a link series (see Chapter 8. "Series Links," beginning on page 233), the graph dialog changes to offer you the option of choosing to plot the data at the native frequency (the frequency of the source page), or at the frequency of the current workfile page (the frequency of the destination page).
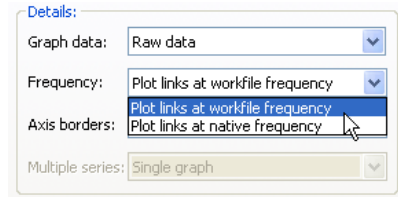
By default, EViews will plot the data at the native frequency of the series. To plot the frequency converted data, you should select **Plot links at workfile frequency**.

Related discussion and examples may be found in "Mixed Frequency Graphs" on page 634.

Note that when plotting links, the **Frequency** dropdown replaces the **Orientation** dropdown. To rotate the graph, you will need to manually assign the series to the bottom axes using the **Axes & Scaling/Data scaling** page of the dialog ("Axis Assignment" on page 652).

Axis Borders

You may use the **Axis borders** dropdown to select a distribution graph to display along the axes of your graphs. For example, you may display a line graph with boxplots or kernel densities along the data (vertical) axis. By default, no axis graphs are displayed (**None**).
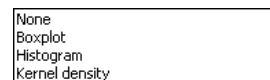
To illustrate, we use the workfile "Treasury2005c.WF1" containing data on 2005 daily market yields for U.S. Treasury securities at constant maturities. We display a line graph for 3-month maturities (TB03MTH) containing a histogram along the data axis.

Note the relationship between the bulges in the distribution and the quarter ends.

Sample Break & NA Handling

By default, an observation graph will leave "spaces" for observations containing missing values. If you look closely at the line graph of TB03MTH above, you may see a few gaps in the line corresponding to days the market was closed.

If there are missing values in your data, the **Basic type** page will change to offer you a choice for how to handle the missing values. You may close the gaps in your graph by checking the box labeled **Connect adjacent non-missing observations** or **Connect adjacent**.

Similarly, if you specify a sample that is non-contiguous, EViews will offer you choices on how to handle the gap in the observation scale. The default is to drop the excluded observations from the graph scale (**Drop excluded obs**), but you may

instead choose to pad the graph with the excluded observations (**Pad excluded obs**). The final option, **Segment with lines**, is the same as **Drop excluded obs**, but with a vertical line drawn at the seams in the observation scale.

In this latter setting, **Connect adjacent** may be used to connect observations across both the seams and across missing values.
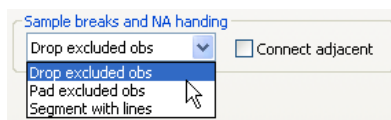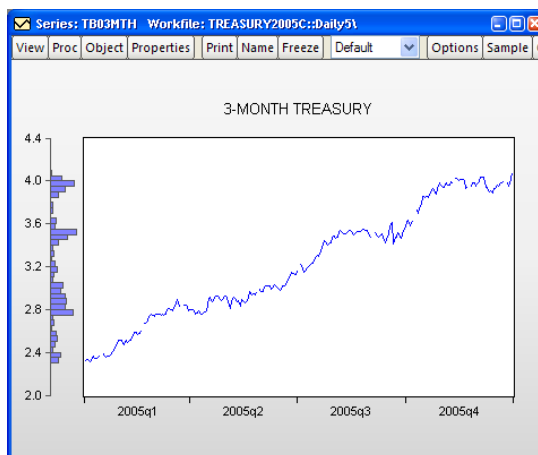
To illustrate, we again consider the TB03MTH series. First, set the workfile sample to exclude missing values for TB03MTH ("smpl if TB03MTH < > NA", and then select **Segment with lines** to produce a graph that highlights the location missing observations.

We see that there are 10 internal missing values in the series, with several, not-surprisingly, in the holiday rich fourth quarter of the year. Notice that the line depicting TB03MTH stops at the two sides of the segment; to connect the lines across the segment, you must select **Connect adjacent**.
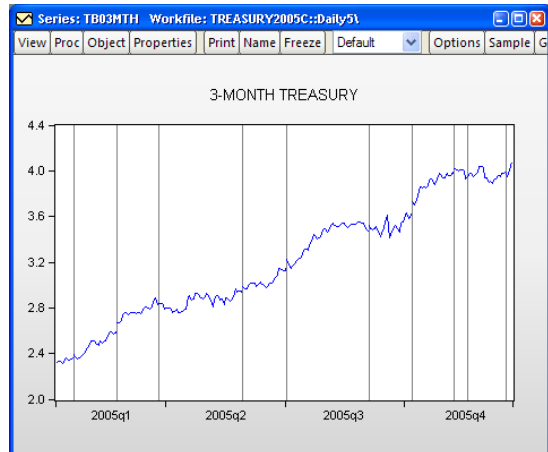
Panel Options

When plotting observation graphs in workfiles with a panel data structure, the **Basic type** page offers additional options for how to use the panel structure. A **Panel options** section will be displayed containing a dropdown menu that controls the panel portion of the display.
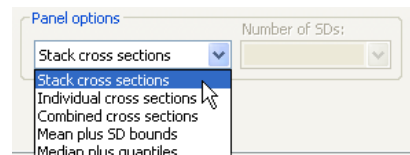
By default, EViews uses the **Stack cross sections** options, which simply stacks the data for each cross- section and plots the data without regard for panel structure. The resulting graph is a observation plot of the entire workfile. For example, a line graph for the series F in the Grunfeld-Baltagi data ("Grunfeld_Baltagi_Panel.WF1"), shows that there is considerable variation across cross-sections, with cross-section 3 in particular having high values:
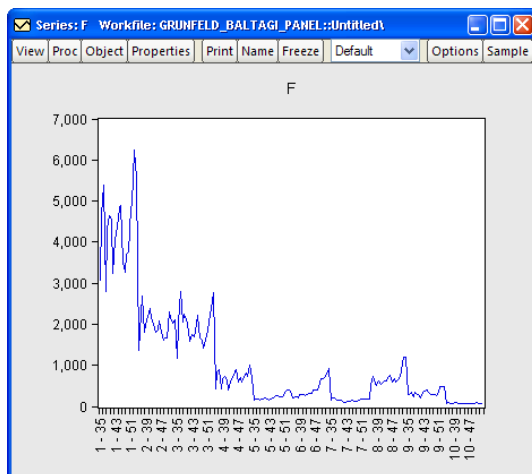
Alternately, you may choose to display a line graph of the data for each cross-section in its own frame (**Individual cross sections**), or in a single frame (**Combined cross sections**).

The combined panel graph for the example panel is given by:
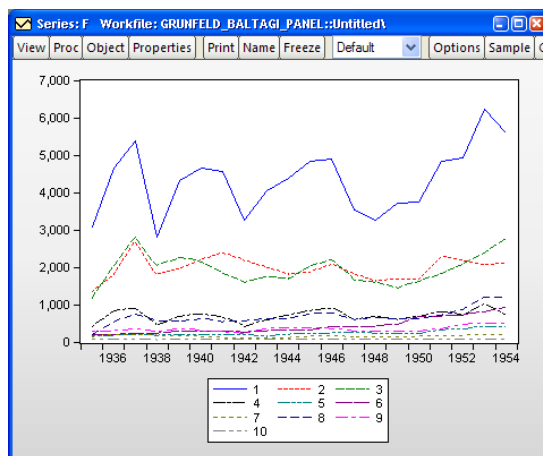
EViews also allows you to plot means plus standard deviation bounds (**Mean plus SD bounds**) or medians plus quantiles (**Median plus quantiles**) computed across cross-sections for every "period". In the latter two cases, EViews will also prompt you for the number of SDs to use in computing the bounds, and the quantiles to compute, respectively.

The means plus/minus two standard deviations graph for the example data is given by:
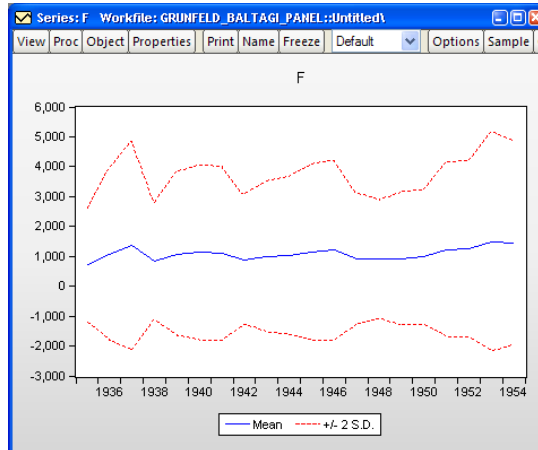
Each observation in the time series for the mean represents the mean value of F taken across all cross-sections in the given period. The standard deviation lines are the means plus and minus two standard deviations, where the latter are computed analogously, across cross-sections for the period.

Multiple Graph Types

We have previously alluded to the fact that we may display multiple Distribution plots or multiple theoretical Quantile-Quantile plots in the same graph. It is easy, for example, to display a graph showing both a histogram of your series data and a fitted normal density curve, or to show Quantile-Quantile plots of your series against both a normal and an extreme value distribution.

To illustrate, we open the debt service ratio series DSR in the "Housedebt.WF1" workfile. We begin by selecting **Distribution** graph as our **Specific** type, and **Histogram** as our specific distribution type.

Next, click on the **Options** button to display the options page. Click on the **Add** button to add an additional distribution graph. EViews displays a new dialog prompting you to select from the list of distribution types that you may add to the histogram. To begin, we select **Theoretical Density**, then click on **OK** to add the element.
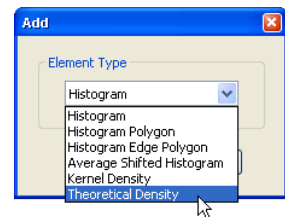
The dialog page changes to reflect your choice. The listbox on the left now shows that we have two different graph elements: the original **Histogram**, and the newly added **Theoretical Distribution**. You may select an element in the list box to show or modify the options for that element. Here we see the options for the **Theoretical Distribution** selection.
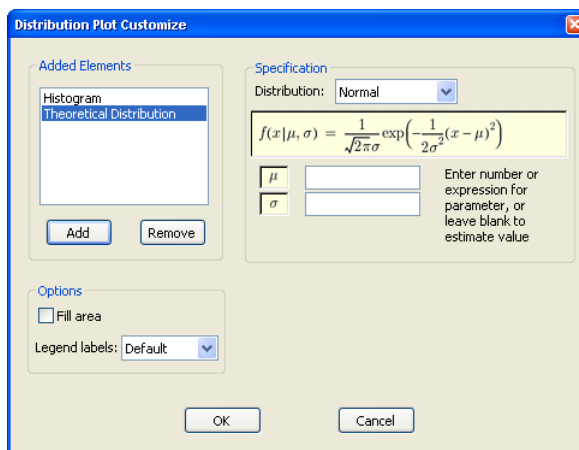
You may add additional elements by clicking on the **Add** button and selecting the desired graph type, or you may remove an element by selecting it in the listbox and pressing the **Remove** button.

For our example, we press the **Add** button again to add a Kernel Density graph to two existing elements.

Returning to the main graph page, you should note that when you have a graph with multiple types, the **Distribution** dropdown on the main page shows that you have a **Custom** graph (not depicted). Click on **OK** to display the specified graph.

EViews displays the fitted normal and kernel density estimates (in red and green, respectively) superimposed over the original histogram. Note that both the kernel density and histograms suggest that there are three distinct groups of observations for the debt service ratio.
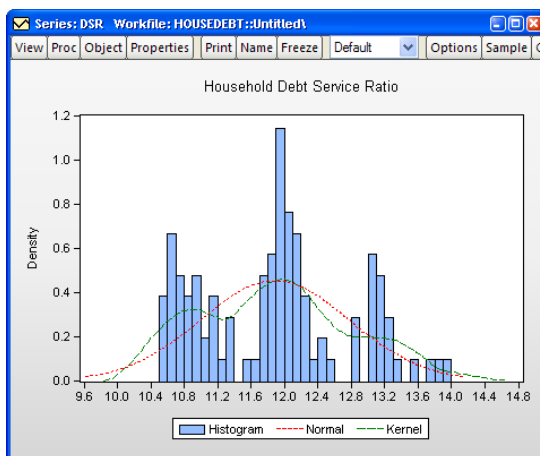
Graphing Multiple Series (Groups)

EViews makes it easy to display graphs of the data in multiple series in a group object. Simply open the group object, select **View/Graph...** and fill out the dialog:
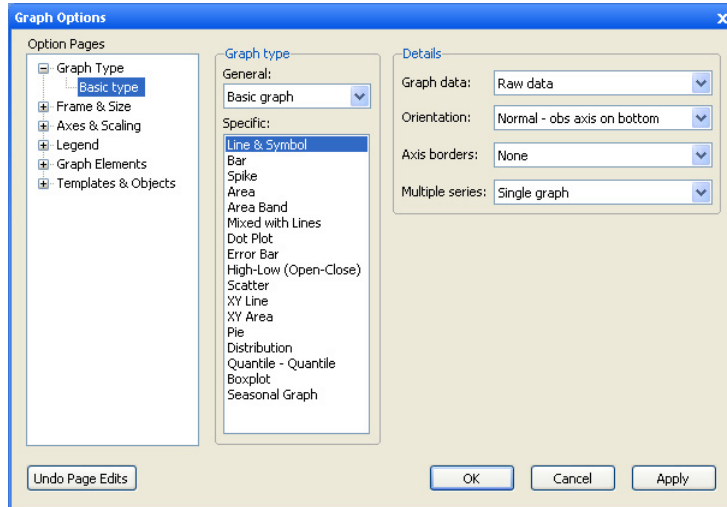
As with the single series dialog, the **Graph Options** dialog has multiple pages that specify various settings for the graph view. We again focus exclusively on the **Basic type** page. The other pages, which control various display characteristics of the graph, are described below ("Basic Customization," beginning on page 648).

Choosing a Type

To select a graph type simply click on its name in the **Specific** type listbox. The options that you will see on this page will depend on the selected graph type. Some of the options (**Orientation**, **Axis borders)** have already been considered (see "Details," on page 623), so we focus here on the remaining settings. To aid in our discussion we divide entries in the listbox into three classes:

- The first class consists of all of the graphs available in the series graph dialog (Line, Bar, Spike, Distribution, *etc.*). For this class, EViews will produce a graph of the specified type for each series in the group. Options will control whether to display the graphs in a single frame or in individual (multiple) frames.

- The second class of graphs use the multiple series to produce specialized observation plots of series data (Area Band, Mixed, Error Bar, High-Low (Open-Close), Pie).

- The final class produce pairwise plots of series data against other series data (Scatter, Bubble, XY Line, XY Area, XY Bar). Options will be used to control how to use the different series in the group, and if relevant, whether to display the graph in a single or multiple frames. Note that these graphs are observation plots in the sense that data for each observation are displayed, but unlike other observation graphs we have seen (*e.g.*, line graphs), data are not plotted against workfile observation indicators.

We consider the settings for each of these classes in turn.

Single Series Graphs

Returning to our Treasury bill workfile ("Treasury2005c.WF1"), we first open the group GROUP01 containing the 1-month, 3-month, 1-year, and 10-year Treasuries, then click on **View/Graph...** to display the graph options dialog.

Multiple Series

When you select any of the individual series graph types in a group with more than one series, the right-hand side of the dialog changes to reflect your choice. In addition to the **Graph data**, **Orientation**, and **Axis borders** settings considered previously there will be a dropdown menu, labeled **Multiple series** which controls whether to display: the individual series in a single frame (**Single graph**), the stacked individual series in a single frame (**Stack in single graph**), or the series in individual frames (**Multiple graphs**).
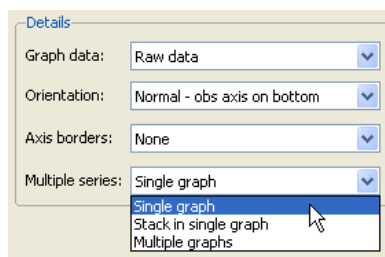
By default, EViews will display all of the series in the group in a single graph frame as depicted here. Each series is given a different color and a legend is provided so that you may distinguish between the various lines.

We see four distinct lines in the graph, each corresponding to one of the series in the group. Displaying the series in the same graph highlights a most notable feature of the Treasury rate data: the narrowing of the spread between yields at different maturities over the course of the year.
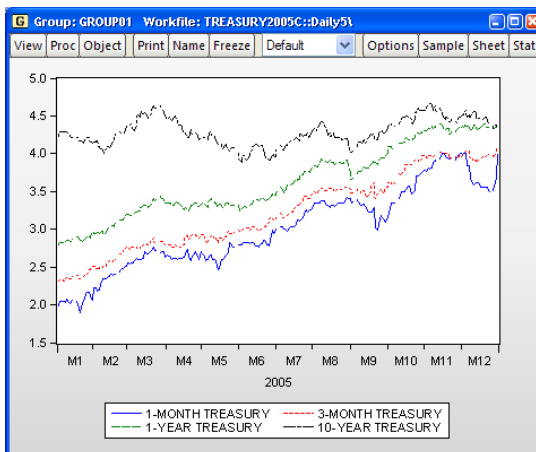
Setting the dropdown to **Multiple graphs** instructs EViews to display each of the series in its own graph, with the individual graphs arranged in a larger graph as shown here for an area graph. We have selected the **Connect adjacent non-missing observations** setting to remove gaps due to missing values.

Note that in contrast to the **Single graph** setting where each
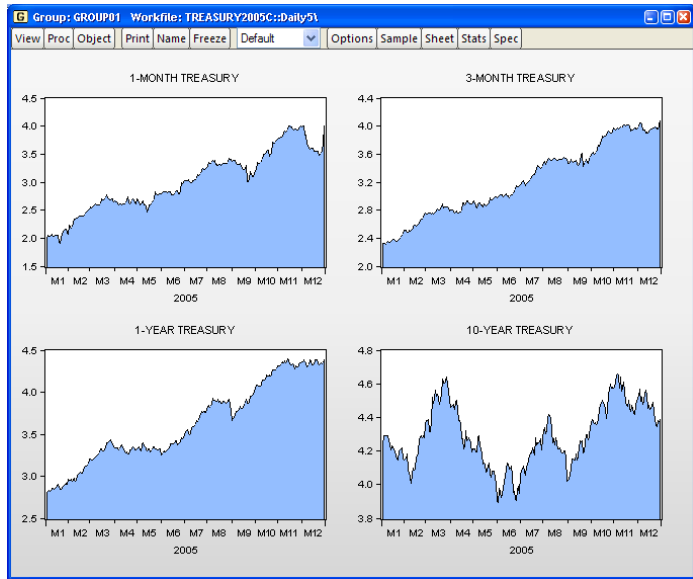
series is plotted on the same scale, each graph is given a different vertical axis scale. This display emphasizes the individual variation in the series, but makes it more difficult to compare across series. Later, we will show how we may control the vertical axes scales ("Axes & Scaling," on page 651).

The final dropdown setting, **Stack in single graph**, allows you to plot data that are sums of the series in the group. This method is available for most, but not all, individual graph types. The first graph element will be the first series plotted in the usual way; the second element will be the sum, for every observation, of the first series and the second. The third element will contain the sum of the first three series, and so forth.

We illustrate the stacked graph using data on newspaper advertising revenue data ("Newspapers.WF1"). The three series in the group object GROUP01 (CLASSIFIED, RETAIL, and NATIONAL), are the three components of TOTAL advertising revenue.

The height of the stacked bar for each observation shows the total amount of newspaper advertising revenue. We see that national advertising is by far the smallest component of advertising revenue and retail is the largest, though classified appears to be growing as a share of total revenue.
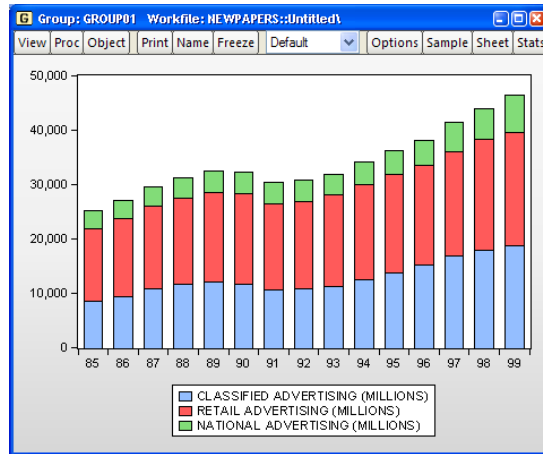
Graph Data

Earlier we saw that the **Graph data** dropdown allows you to display summary statistic graphs (**Means**, **Medians**, *etc.*) for your data ("Graph Data" on page 623). For graphs of a single series, displaying summary data may be of limited value since the graph will show a single summary value. For multiple series, the dropdown allows us to display graphs that compare values of the statistics for each of the series in the group.
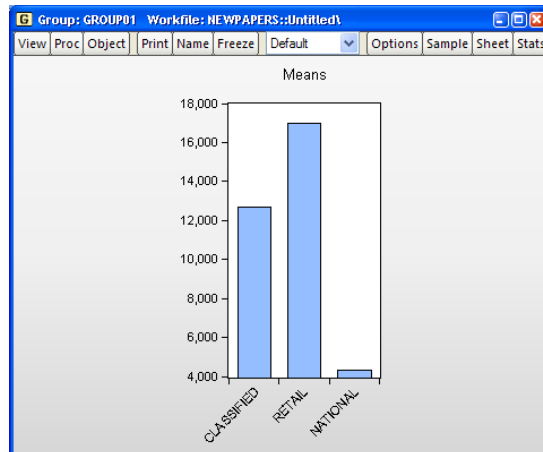
Once again using the newspaper advertising revenue series in group GROUP01, we set the **Graph data** dropdown to **Mean** and display a bar graph with the multiple series displayed in a single frame. We see that the means of both RETAIL and CLASSIFIED advertising revenue are significantly greater than the average NATIONAL revenue.

Mixed Frequency Graphs

One important application of multiple series graphs involves displaying line graphs of mixed frequency data. You may, for example, have a workfile with two pages, one containing data sampled at a monthly frequency, and the other sampled at a quarterly frequency. EViews

allows you to display line graphs of data from both pages in a single graph, with each series plotted at its native frequency.

To illustrate, we again use our Treasury bill workfile ("Treasury2005c.WF1"). We work with the group GROUP02 in the "Daily5" page, containing the series TB03MTH, AAA, and GDP-CHG. TB03MTH is, as we have already seen, the 3-month T-bill series measured at a 5-day daily frequency.

The other series in the group are link series. (See Chapter 8. "Series Links," beginning on page 233 for a discussion of links). AAA, which is linked from the **Monthly** workfile page, contains data on Moody's Seasoned Aaa Corporate Bond Yield. GDPCHG, which is linked from the **Quarterly** workfile page, measures the (annualized) quarterly percent change in GDP (in chained 2000 dollars). Both links convert the low frequency data to high using the constant-match average frequency conversion method.

Note that since the two link series are tied to data in other workfile pages, EViews has access to both the native (monthly and quarterly) and the converted (daily 5) frequencies for the AAA and GDPCHG. Accordingly, the main graph dialog for GROUP02, prompts you for whether you wish to plot your links using the native frequency data, or whether you wish to plot links using the workfile frequency (frequency converted) data.

We first display a line graph of the series in the group using the **Plot links at native frequency** setting. Since TB03MTH is sampled at the workfile frequency, this graph is a mixed frequency graph, with TB03MTH plotted at a daily-5 frequency, AAA plotted at a monthly frequency, and GDPCHG plotted at a quarterly frequency. To make it easier to see the different frequencies in the plot, we display AAA and GDP-CHG using lines and symbols ("Lines and Symbols" on page 665), and we add vertical grid lines ("Frame" on page 649) to the graph.
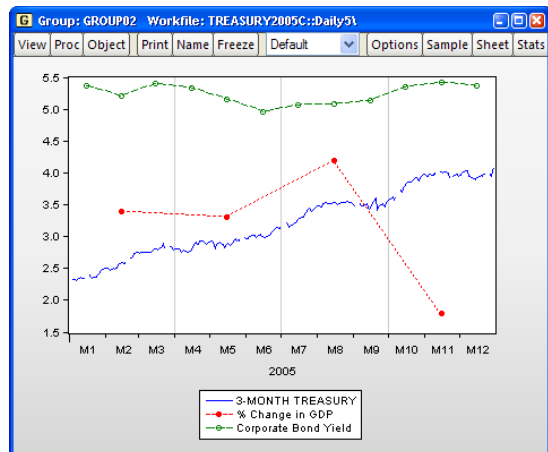
Note that the GDPCHG line connects the four quarterly values of the series measured at its native frequency. The four points are each centered on the corresponding range of daily-5

dates. Similarly, the 12 monthly values of AAA are connected using line segments, with the individual points centered on the appropriate range of daily-5 values.

We may compare this graph to the same plot using the **Plot links at workfile frequency** setting. Here, all three series are plotted at the daily-5 frequency, with the AAA and GDP-CHG series using the frequency converted values. Note that the graph simply uses the values that are displayed when you examine the link series in the spreadsheet view.
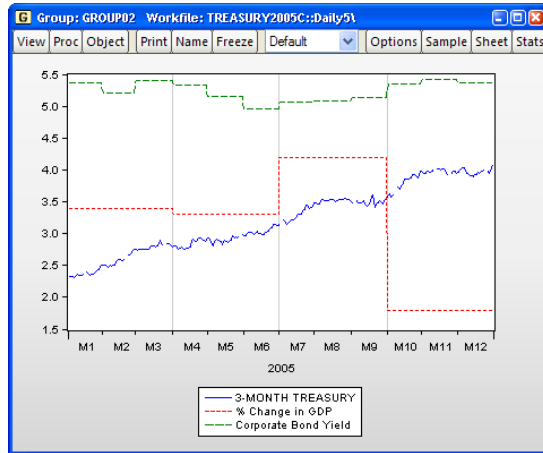
In contrast to the earlier graph, AAA and GDPCHG are displayed for each daily-5 date. Since the frequency conversion method for both series was to use a constant value, the graphs for AAA and GDPCHG are step functions with steps occurring at the native frequency of the links.

Specialized Graphs

The Area Band, Mixed, Error Bar, High-Low (Open-Close), and Pie graph types use multiple series in the group to form a specialized graph. Each specific type has its own set of options. For additional detail and discussion, see the description of the individual graph type in "Observation Graphs," beginning on page 669.

Pairwise Graphs

The final class of graphs use data for a given observation in pairs, plotting the data for one series against data for another series (Scatter, Bubble, XY Line, XY Area, XY Bar).

For Scatter, XY Line, and XY Area graphs for groups containing exactly two series, there is no ambiguity about how to use the data in the group; there will be a single graph frame with the first series placed along the horizontal axis and the second series along the vertical axis. When there are more than two series, you will be prompted on how to use the multiple series to form data pairs and whether to display the graphs in a single or multiple frames.
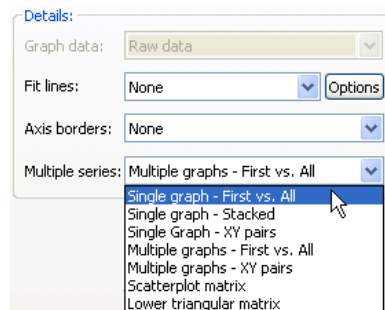
Single graph - First vs. All

This setting forms graph pairs using the first series along the horizontal axis plotted against each of the remaining series along the vertical axis. The graph displays all of the graph pairs in a single frame.

We illustrate using a scatterplot of GROUP01, which contains our Treasury data at different maturities. The first series in the group is the 1-month Treasury rate, which is plotted against the remaining series in the group.
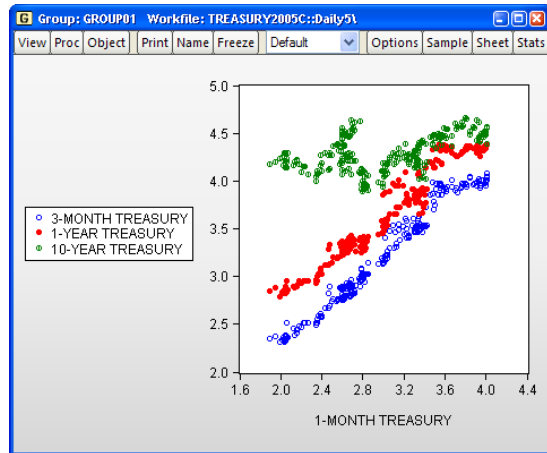
Single graph - Stacked

As the name suggests, this setting plots the first series against the remaining series in stacked form. Thus, the first series is plotted against the second series, against the sum of the second and third series, against the sum of the second through fourth series, and so forth.

We illustrate using our data on newspaper advertising revenue data ("Newspapers.WF1"). For GROUP01, we show the stacked XY graph that plots CLASSIFIED against RETAIL and CLASSIFIED against the sum of RETAIL and NATIONAL.
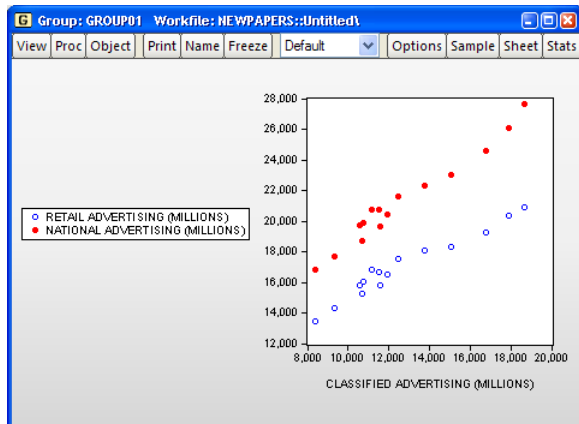
Single graph - XY pairs

This setting forms pairs by using successive pairs of series in the group. The first series is paired with the second, the third with the fourth, and so on, with the first series in each pair placed on the horizontal axis, and the second series placed on the vertical axis. If the group contains an odd number of series, the last series will be ignored. The graph uses a single frame for all of the graph pairs.

Multiple graphs - First vs. All

Like **Single graph** - **First vs. All**, this setting plots the first series against the remaining series, but instead places each pair in an individual graph frame.

Multiple graphs - XY pairs

Like **Single graph** - **XY pairs**, this setting forms pairs by using successive pairs of series in the group, but places each pair in an individual graph frame.

We again illustrate using an XY line graph of the group object GROUP01 containing our Treasury data. The first series in the group is the 1-month Treasury rate, which is plotted against the remaining series in the group.

Note that each graph has its own data frame and vertical axis scale. In addition, we may manually set the vertical axes scales ("Axes & Scaling," on page 651).
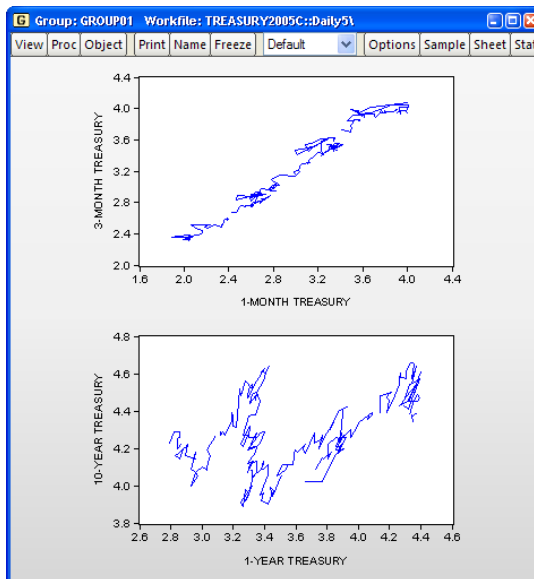
Scatterplot matrix

The **Scatterplot matrix** setting forms pairs using all possible pairwise combinations for series in the group and constructs a plot using the pair. If there are k series in the group, there will be a total of k^2 plots, each in its own frame.
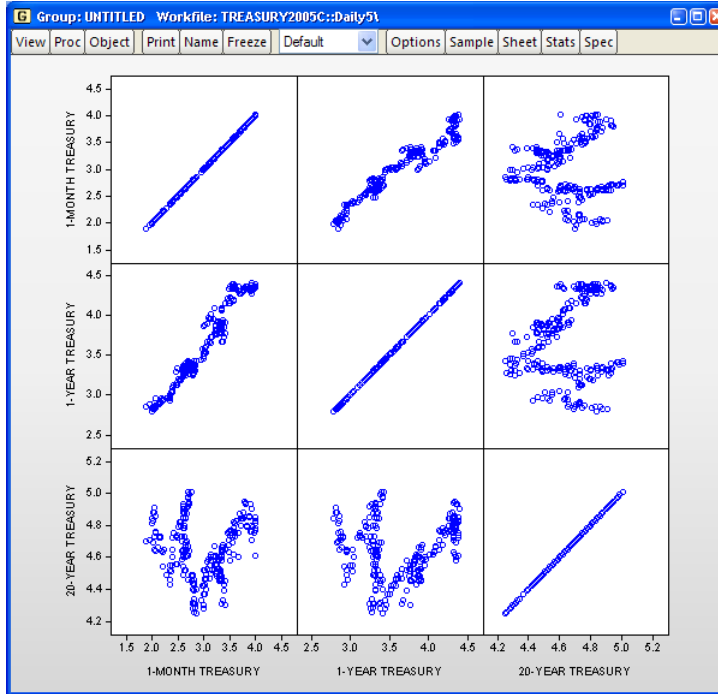
Note that the frames of the graphs in the scatterplot matrix are locked together so that the individual graphs may not be repositioned within the multiple graph frame.

Lower triangular matrix

This setting constructs the same plots as **Scatterplot matrix**, but displays only the lower triangle elements consisting of the unique pairs of series not including the series against itself. There are a total of $k(k-1)/2$ distinct pairwise graphs, each displayed in its own frame.
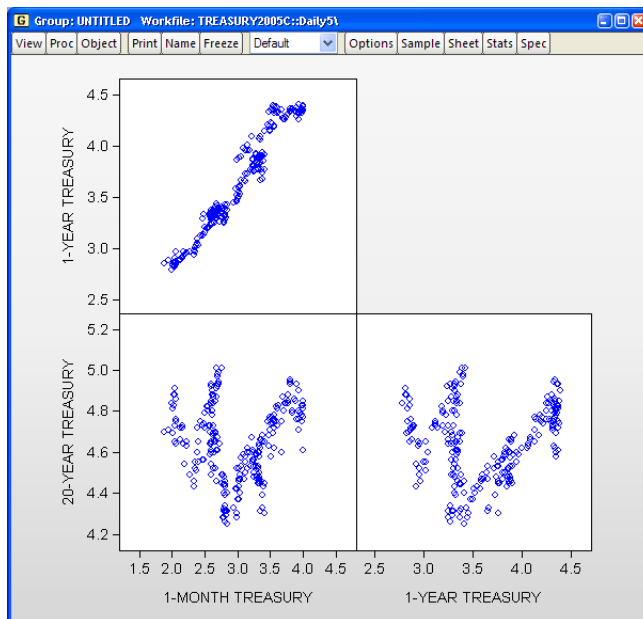
Note that the frames of the graphs in the lower triangular matrix are locked together so that the individual graphs may not be repositioned within the multiple graph frame.

Fit Lines

EViews provides convenient tools for superimposing auxiliary graphs on top of your Scatter or XY line plot, making it easy to put regression lines, kernel fits, and other types of auxiliary graphs on top of your XY plots.

When you select **Scatter** or **XY Line** from the **Specific** type listbox, the right-hand side of the page changes to offer a **Fit lines** option, where you may add various types of fit lines to the graph as outlined in "Auxiliary Graph Types," beginning on page 704. You may also use the **Options** button to add additional auxiliary graphs. To illustrate, we use the familiar "Old Faithful Geyser" eruption time data considered by Simonoff (1996) and others ("Geyser.WF1"), and add both a regression line and a nearest neighbor fit relating eruption time intervals to previous eruption durations.

First, we open the group GROUP01 and select **Scatter** as our **Specific** type, then select **Regression Line** in the **Fit lines** dropdown to add a linear regression line. Next, click on the **Options** button to display the **Scatterplot Customize** page.

Click on the **Add** button to add an additional fit line to the existing graph. EViews displays a new dialog prompting you to select from the list of fit lines types that may add to the scatterplot with regression line. We will select **Nearest Neighbor Fit** to be added to the existing graph. Click on **OK** to accept your choice.
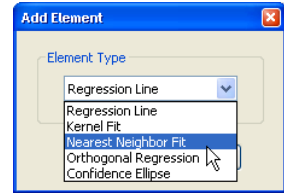
You may elect to add additional elements by clicking on the **Add** button, or to remove an element by selecting it in the listbox and clicking on the **Remove** button.

Returning to the main graph page, we see that the **Fit lines** dropdown now reads **Custom** (not depicted) indicating that we are using multiple graph types.

Click on **OK** to accept the graph settings, and EViews displays the scatterplot with both the linear regression fit and the default LOWESS nearest neighbor fit superimposed on the observations. Note that since there are two lines in the graph, EViews provides legend information identifying each of the lines.
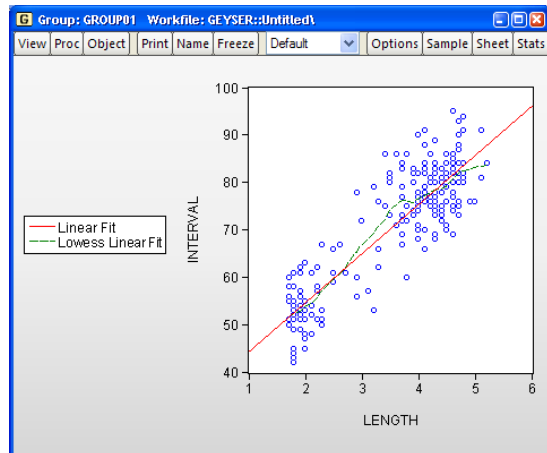
We see the nearest neighbor fit has a slightly higher slope for lower values of INTERVAL and a lower slope at higher values of INTERVAL than the corresponding linear regression.

The Graph Sample

The graph *sample* slider bar, located at the bottom of a sample based graph, allows you to dynamically modify the display sample of the graph by resizing and dragging the bar.

Here, we display a line graph of series containing stock prices in a workfile dated from 2005 to 2012.
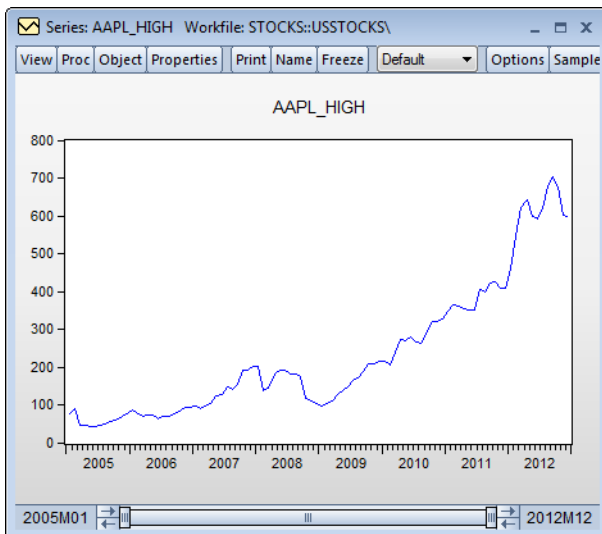
We see the slider bar at the bottom of the window. The left-hand text shows the first observation in the current graph view ("2005m01") and the right-hand text shows the last observation ("2012m12"). In this example, the display sample spans the entire workfile so the bar is drawn from end to end.

To focus attention on the last two years of data, we may move the graph view start date by sliding the left handle of the slider bar toward the right until the left-hand date indicates that we are showing data from 2011 onward. The line graph will change dynamically as you move the slider bar, bringing 2011 and 2012 into better view.
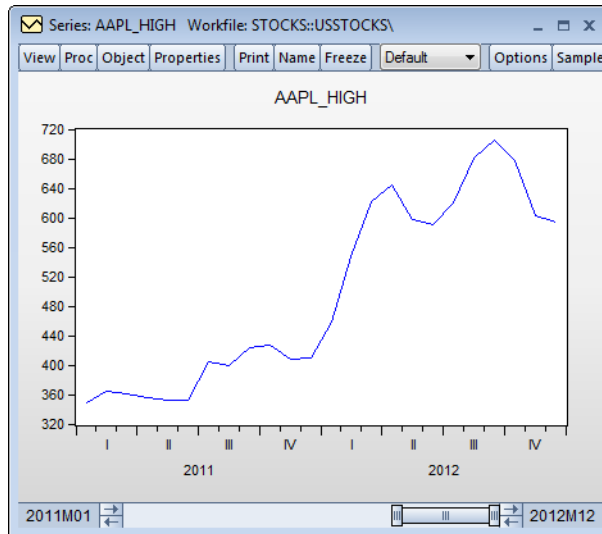
Similarly, you may drag the ending date for the graph view to display only data prior to some date in your graph. If you wish to look at a sliding window of two years, you may click on the middle of the slider bar and drag it back and forth.
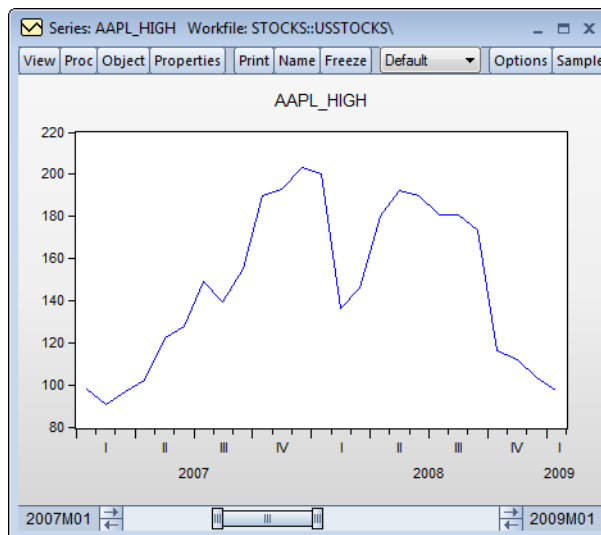
Notice that the start and end dates associated and the graph image both dynamically update to reflect the current slide position.

For more precise graph sample adjustment, you may use the arrows on either side of the slider bar to adjust the graph sample endpoints in single increments.

Note that for graph views of series and group objects, the slider bar sample changes are temporary. The graph sample will be reset to the workfile sample when the window is closed or updated. Graph objects, however, retain the sample that has been set by the bar.

If you'd like to update the workfile sample to match the current graph sample right-click anywhere in the slider and select **Set as workfile sample**.

Graph Pan and Zoom

Panning and zooming is available on graphs and graph views that feature the observation slider bar (*i.e.,* those graph views that allow for subsample display). You may use these features to alter the display to focus on particular regions of the graph, much like cropping a photograph.

To zoom, left-click on the graph and drag the cursor to draw a box around the area into which you would like to zoom.
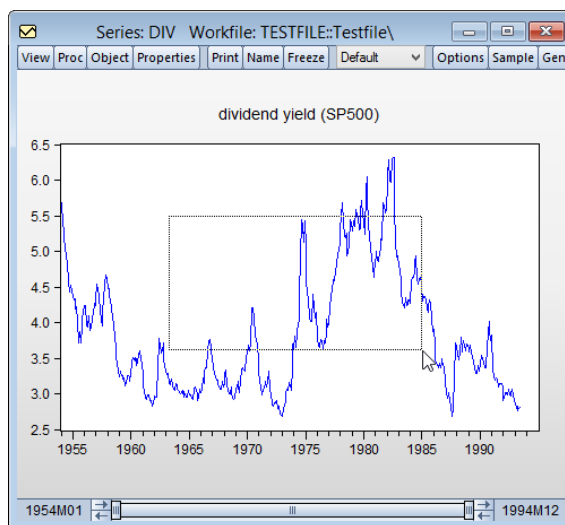
then release the mouse. The display changes to show only the area within the box and the observation slider bar and axis labels will change to reflect the modified display.
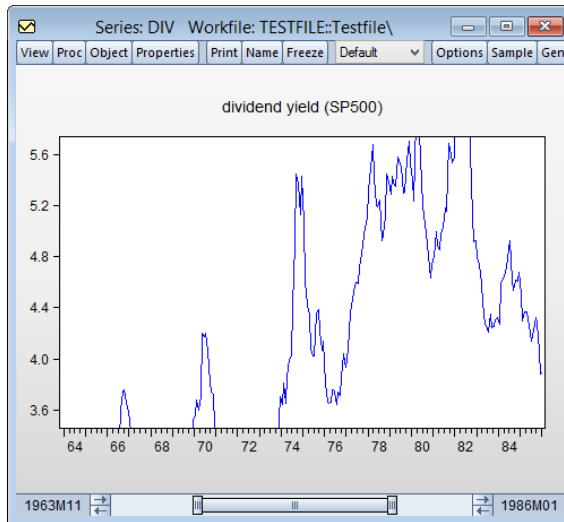
Note that there will be no indication that observations from the original graph have been trimmed away.

To pan, right-click on the graph, and when the cursor change to a hand, drag the image and release when the display shows the portion of the graph that is of interest.
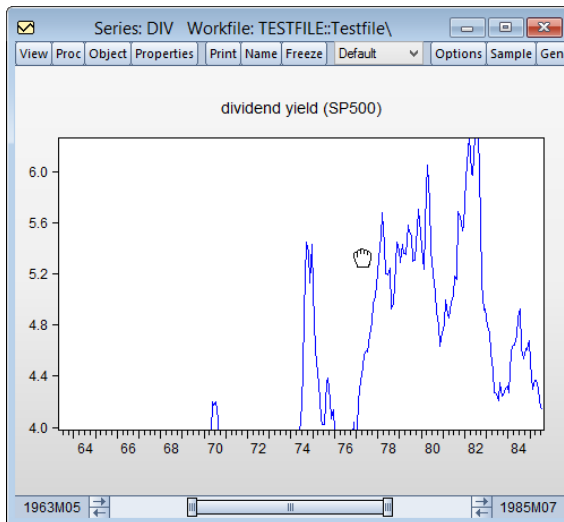
Note that zooming and panning does not change the graph, only the visibility of the graph in the window. The observation slider bar and axis labels will change to reflect the modified display. Freezing the graph will take a snapshot of the current display.

To restore the original graph display, press ESC.

Multi-graph Slideshow

The graph slideshow makes it easier to examine multiple graphs in the number of graphs in the frame results in individual graphs where it is too small to see any detail. Consider, for example, the graph of impulse responses for a three variable VAR:
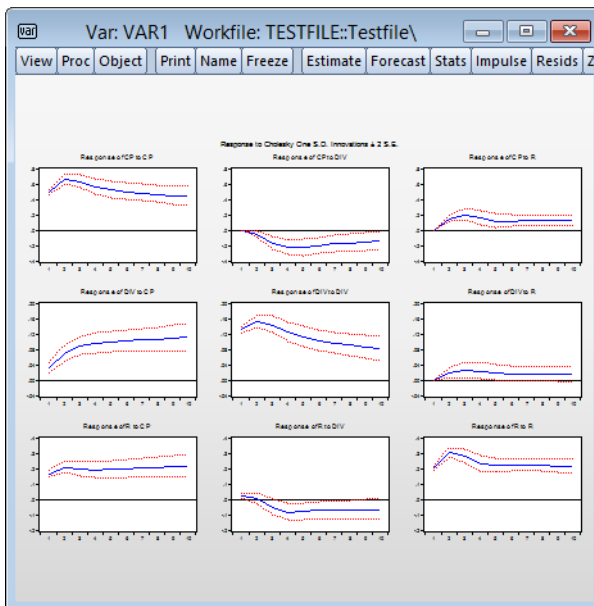
EViews allows you to zoom in on the individual graphs and to display the contents of the graph in a slideshow format. Click on the **Zoom** button on the toolbar to enable the slide-show:
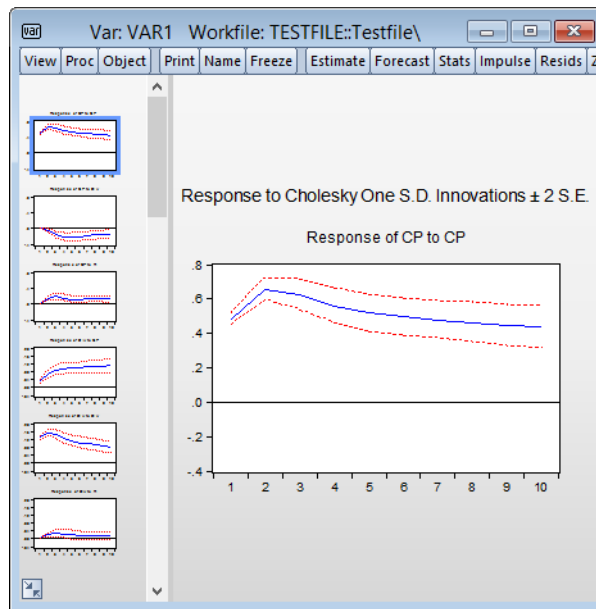
Here we see the zoom view. On the left is a gallery of the individual graphs; on the right is a larger display of the selected graph. You can page through the individual graphs using the arrow keys or **Page Up/Page Down** to select the next graph, or the **Home** and **End** keys to move to the first or last graph. Alternately, you can click on a gallery item to jump to that graph.

Click on ESC to exit zoom mode.

If you wish to enable the slideshow for a subset of the graphs, return to the original graph and select the graphs of interest by clicking or drag clicking, then press the **Zoom** button or right-mouse click and select **View/Selected graphs**.
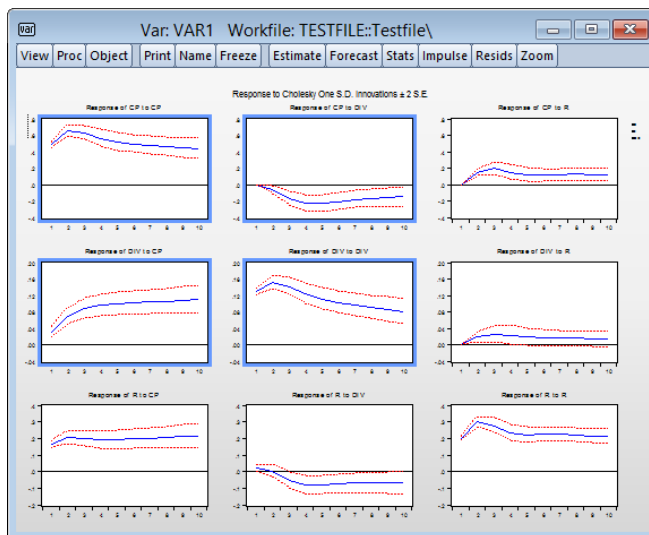
then press the **Zoom** button or right-mouse click and select **View/Selected graphs**. EViews will show the gallery for just the selected items:
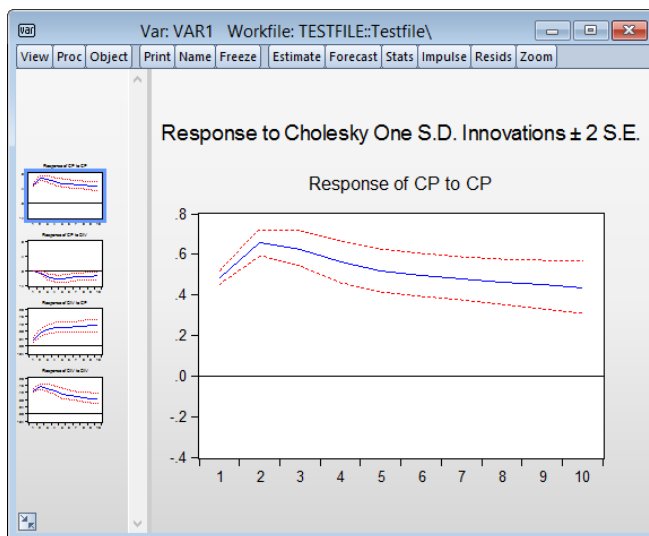
Basic Customization

EViews allows you to perform extensive customization of your graph views at creation time or after the view is displayed. You may, for example, select your graph type, then click on the other option groups to change the graph aspect ratio, the graph line colors and symbols,

and the fill colors, then click on **OK** to display the graph. Alternately, you may double-click on an existing graph view to display the **Graph Options** dialog, change settings, then display the modified graph. And once a graph view is frozen, there are additional features for adding text, lines, and shading.

We defer a detailed discussion of graph customization to later chapters. Here we describe a handful of the most commonly performed graph view customizations.

You should be aware that many of the options that we describe below are transitory and will be lost if you change the graph type. For example, if you set the symbol colors in a scatterplot and then change the graph to a line graph, the color changes will be lost. If you wish to make permanent changes to your graph, you should freeze the modified graph view or freeze the graph view and then make your change to the resulting graph object.

Frame

The **Frame** option group controls characteristics of the basic graph view window, including the frame appearance, colors in the graph not related to data, and the position of the data within the frame.

Color & Border

The **Color & Border** section of the **Frame** group contains two sets of options. The **Color** group controls fill and background colors, while the **Frame border** group modifies the graph frame itself.

Color

The **Color** group allows you to choose both the color inside the graph frame (**Frame fill**) and the background color for the entire graph (**Background**). You may also apply a fade effect to the frame color or background color using the corresponding drop-down menus.

The final two settings are related to the behavior of graphs when they are printed or saved to file. The first option, **Make background transparent for printing and file saving**, should be used to ignore the background color when printing or saving the graph (typically, when printing to a black-and-white printer). Unchecking the last option, **Use grayscale-no color**, changes the display of the graph to grayscale, allowing you to see how your graph will look when output to a black-and-white device.
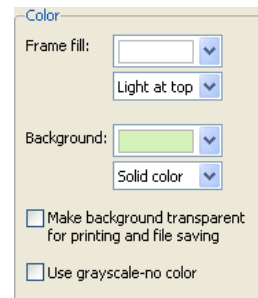
Here, we display a histogram of data on three month CD rate data for 69 Long Island banks and thrifts ("CDrate.WF1"). These data are used as an example in Simonoff (1996).

We have customized the graph by changing the color of the background (obviously not visible in black-and-white), and have applied a fade fill to the graph frame itself. The frame fill is light at the top and dark at the bottom.
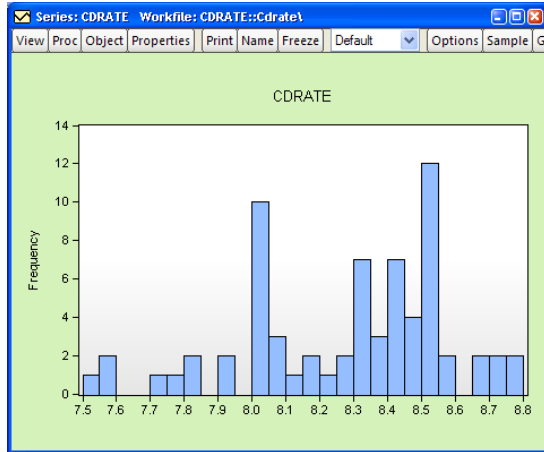
Frame Border

The **Frame border** group controls the drawing of the graph frame. The **Axes** dropdown describes the basic frame type. The first entry in the dropdown, **Labeled Axes**, instructs EViews to draw a frame line for each axis that is used to display data in the graph. The last entry, **none**, instructs EViews not to draw a frame. The remaining dropdown entries are pictographs showing the positions of the frame lines. In this example, we will display a box frame.
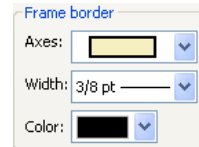
Size & Indents

The **Size & Indents** section provides options for positioning and sizing the frame. The **Frame size** group controls the height and width of the graph, while **Graph position in frame** positions the data plot inside the graph.

Frame Size

The **Frame size** group is used to control the aspect ratio of your graph and the relative size of the text in the graph. (Note that you can also change the aspect ratio of the graph by click-and-dragging the bottom or right edges of the graph.)

The first two settings, **Height in inches** and **Width**, determine the graph frame size in virtual inches. You may specify the width in absolute inches, or you may specify it in relative terms. Here, we see that the graph frame is roughly 6×3 inches since the height is 3.00 inches and the width is 2.000 times the height.
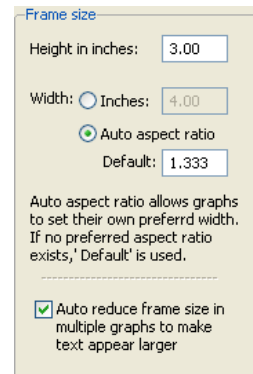
Note that all text in graphs is sized in terms of absolute points (1/72 of a virtual inch), while other elements in the graph are sized relative to the frame size. Thus, reducing the size of the graph frame increases the relative size of text in the graph. Conversely, increasing the size of the graph frame will reduce the relative size of the text.

We can see the effect of changing both the aspect ratio and the absolute graph size using our example workfile "Housedebt.WF1". We display a line plot of the data in GROUP01, with the **Auto aspect ratio Default** set to 3, and the **Height in inches** to 1. The resulting graph is now three times as wide as it is tall. Note also the increase in axis label and legend text size compared with the corresponding example in "Quick Start," on page 618.
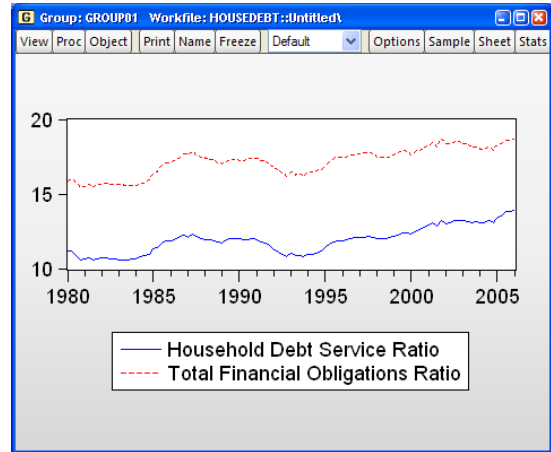
There is one additional checkbox, labeled **Auto reduce frame size in multiple graphs to make text appear larger**, which, when selected does as advertised. When displaying multiple graphs in a given frame size, there is a tendency for the text labels and legends to become small and difficult to read. By automatically reducing the frame size, EViews counteracts this undesired effect.

Position of graph in frame

The two dropdown menus in **Graph position in frame** control the position of the plot within the graph frame. Note that different graph types use different default settings, but you may override them using the two dropdowns. Using positive values for these settings can help insure that your data points are not obscured by drawing them on top of the axes scale lines.
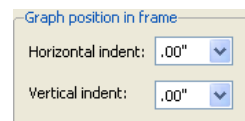
Axes & Scaling

The **Axes & Scaling** option group controls the assignment of data to horizontal and vertical axes, the construction of the axis scales and labeling of the axes, and the use of tickmarks and grid lines.

This group is divided into sections based on the type of scale being modified. There are two different scale types: *data scales,* and *workfile (observation) scales*. When series data are assigned to a given axis, the axis is said to have a data scale, since the data for the series are plotted using that axis. Alternately, if observation identifiers are plotted along an axis, we say that the axis has a workfile or observation scale.

Some observation graphs (Line graphs, Bar graphs, *etc.)* have both data and observation scales, since we plot data against observation indicators from the workfile. Other observation graphs (Scatter, XY Line, *etc.*) have only data scales, since data for multiple series are plotted against each other. Similarly, analytic graphs (Histogram, Empirical CDF, *etc.*) have only data scales since the derived data are not plotted against observations in the workfile.

The first two sections in this group, **Data scaling** and **Data axis labels**, relate to settings for the data scales. The third section, **Obs/Date axis**, sets options for the workfile (observation) scales. The fourth section, **Grid Lines**, can be used to add or modify grid lines on any axis.

Data scaling

The **Data scaling** section allows you to arrange data among axes and set data scaling parameters for each axis.
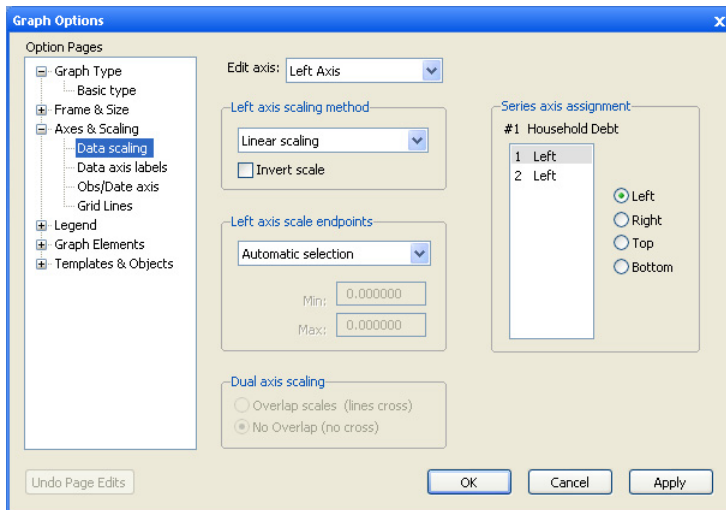
Axis Assignment

The right-hand side of the dialog contains a section labeled **Series axis assignment,** which you may use to assign each series in the graph to an axis. The listbox shows each data element along with the current axis assignment. Here we see the assignment for a scatterplot where the first series is assigned to the bottom axis and the second series is assigned to the left axis.

To change the axis assignment, simply click on a graph element in the listbox, then click on one of the radio buttons to perform the desired assignment. Note that when you select an element, the top of the section shows information about the selected data series.
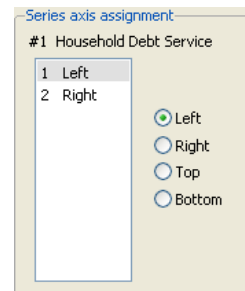
Note that the rules of graph construction imply that there are restrictions on the assignments that are permitted. For example, when plotting series or group data against workfile identifiers (as in a line graph), you may assign your data series to any combination of the left and right axes, or any combination of the top and bottom axes, but you may not assign data to both vertical and horizontal axes. Similarly, when plotting pairwise series data, you may not assign all of your series to a single axis, nor may you assign data to all four axes.

We have already seen one example of changing axis assignment. The **Orientation** dropdown on the **Type** page is essentially a shorthand method of changing the axis assignments to display the graph vertically or horizontally (see "Orientation," on page 624).

A second common example of axis assignment involves setting up a dual scale graph, where, for example, the left hand scale corresponds to one or more series, and the right-hand scale corresponds to a different set of series.

Once again working with GROUP01 in our debt service ratio dataset, we see the display of a dual scale line graph where the first line is assigned to the left axis, and the second line is assigned to the right.
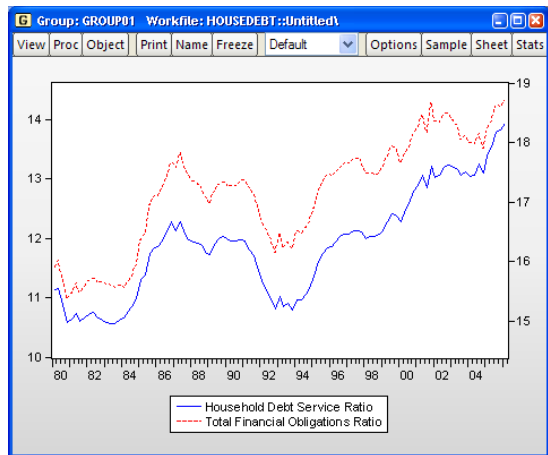
When you specify a dual scale graph with series assigned to multiple axes, the bottom left-hand side of the dialog will change to offer you the choice to select the axis scales so that the lines do no overlap, or to allow the lines to overlap.
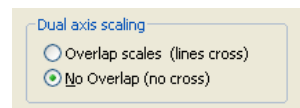
Selecting an Axis to Edit

The **Edit axis** dropdown at the top of the dialog is used to select a scale (left, right, top, bottom) for modification. When you select an entry, the remainder of the dialog will change to reflect the characteristics of the specified scale.
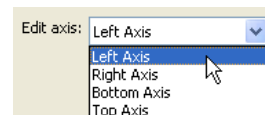
Defining Data Scales

The remainder of the **Data scaling** page allows you to specify properties of the data scale.

We have selected the left axis in our example, so we see the two relevant sections of the dialog labeled **Left axis scaling method** and **Left axis scale endpoints**.

Axis Scale

The **Axis scaling method** describes the method used in forming the selected axis scale. By default, EViews displays a **Linear scale**, but you may instead choose: a linear scale that always includes the origin (**Linear – force zero**), a logarithmic scale (**Logarithmic scaling**), or a linear scale using the data standardized to have mean 0 and variance 1 (**Normalized data)**. If you select the **Invert scale** option, EViews will reverse the scale so that it ranges from high values to low.

You may use the **Axis scale endpoints** to control the range of data employed by the scale. If you select **User specified**, you will be prompted to enter a minimum and maximum value for the scale. Note that if either of these are within the actual data range, the graph will be clipped.

We illustrate log scaling and user-specified axis endpoints using the GDP series from our newspaper advertising revenue data ("Newspapers.WF1") workfile. In addition to drawing the data with log scaling, we have set the endpoints for our vertical axis to 0 and 10,000 (the default endpoints are 4,000 and 10,000).

Note that EViews has chosen to place tickmarks at every 1,000 in the scale, leading to the unequal spacing between marks.
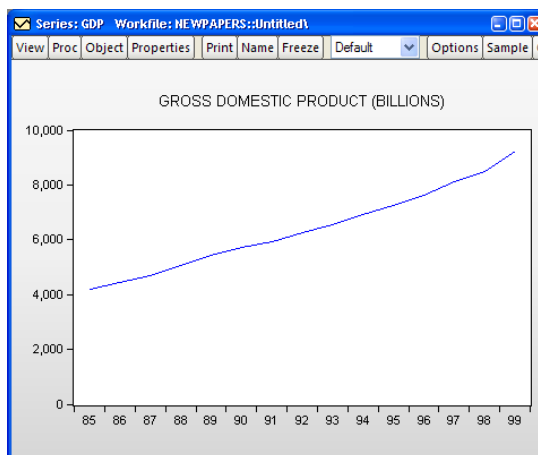
Labeling the Data Axis

The next section in the **Axes & Scaling** group is **Data axis labels**. This section contains options for all characteristics of the data axes which do not affect the scaling of the data.
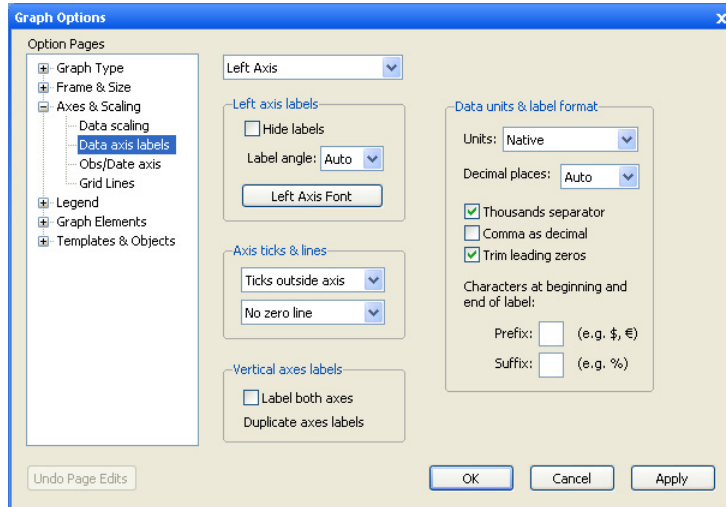
Similar to the **Edit axis** dropdown menu on the **Data scaling** page, you may use the drop-down menu at the top of the **Data axis labels** page to select an axis to edit.

Axis Labels

You may suppress all labels for the selected axis by checking the **Hide labels** box.

If you do choose to display labels for the specified axis, you may use the **Label angle** dropdown to rotate your labels. Note that the values in the dropdown describe counterclockwise rotation of the labels, hence selecting **45** in the dropdown menu rotates the axis labels 45 degrees counter-clockwise while selecting **−30** rotates the labels 30 degrees clockwise.
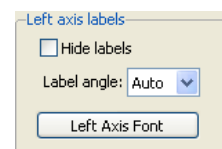
Clicking on **Axis Font** brings up a font dialog allowing you to change the size and typeface of your labels.

Ticks and Lines

The **Ticks & lines** section controls the display of tickmarks. The first dropdown menu determines the placement of tickmarks: you may choose between **Ticks outside axis**, **Ticks inside axis**, **Ticks outside & inside**, and **No ticks**.

The **Zero line** checkbox controls whether to draw a horizontal or vertical line at zero along the specified axis. Note that it is possible to select **Zero line** for an axis scale that does not include the origin; in this case, the option has no effect.
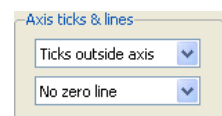
Duplicating Axis Labels

For a graph with all data series assigned to one horizontal or one vertical axis (or, at most, one vertical and one horizontal in the case of an XY graph), the bottom left-hand side of the dialog offers you the choice to **Label both axes**. This option will duplicate the labels onto the corresponding empty axis. For example, for a graph with all series assigned to the left axis, you may elect to label the right axis with the same labels by checking this box.

Data Units & Labels

The **Data units & label format** group can be used to label your axis using scaled units or to customize the formatting of your labels.

- The **Units** dropdown menu allows you to display your data using a different scale. You may choose between the default setting **Native**, **Percent: .01**, **Thousands: 1e3**, **Millions: 1e6**, **Billions: 1e9**, and **Trillions: 1e12**. For example, selecting **Thousands: 1e3** will display the data in units of a thousand; it is equivalent to dividing the data by 1,000 before graphing. Similarly, selecting **Percent: .01** effectively multiplies the data by 100 prior to display.

- The **Decimal places** dropdown specifies the number of digits to display after the decimal. In addition to the default **Auto** setting, you may choose any integer from 0 to 9.

- The **Thousands separator** option controls whether numbers employ separators to indicate thousands. By default, EViews will display a separator between thousands (*e.g.,* "1,234" and "2,123,456", or "1.234" and "2.123.456" if **Comma as decimal** is selected), but you may uncheck the **Thousands separator** option to suppress the delimiter.

- The **Comma as decimal** option controls whether the comma is used as the decimal delimiter. If checked, the decimal and comma indicators will be swapped: the decimal indicator will be the comma instead of the period, and the thousands separator, if used, will be the period instead of the comma.

- By default, EViews will trim leading zeros in numbers displayed along the axis, but you may uncheck the default **Trim leading** zeros checkbox to display these zeros.

- In addition, you may provide a single character prefix and/or suffix for the numbers displayed along the axis.

For example, suppose that we have data that are expressed as proportions ("0.153"). To display your axis as percentages ("15.3%"), you may select **Percent: .01** as the **Units**, and add "%" as the suffix.

Defining Observation Scales

The third option group under **Axes & Scaling** is **Obs/Date axis**. This page should be used to define all aspects of the workfile or observation scale.
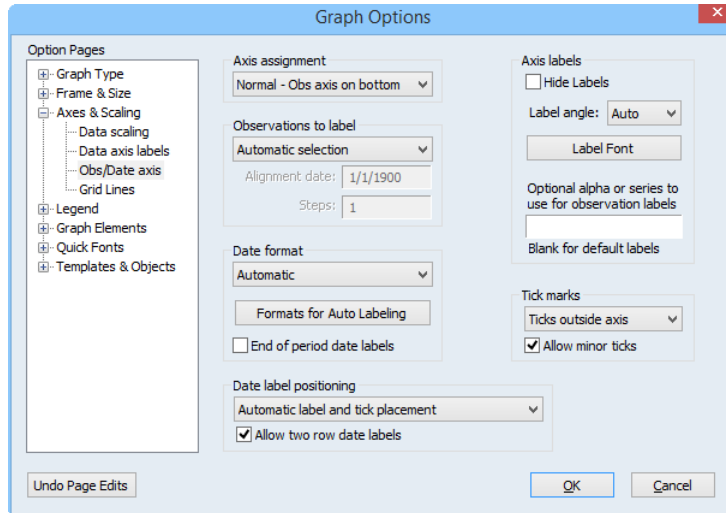
For graphs which do not have a date axis, such as XY graphs, the **Axis assignment** drop-down menu in the top left-hand corner of the page will read **Graph has no Obs/Date axis**, and any changes to the settings on this page will be irrelevant for the current graph type.

Rotating the Graph

If you are working with a graph which has a date axis, this drop-down menu provides you with a shortcut for rotating the graph. You may select **Normal - Obs axis on bottom** or **Rotated - Obs axis on left**. Choosing the latter will move the date axis to the left-hand axis and reassign all remaining axes to the bottom axis. The dropdown menu provides the most common options, assigning to either the left or bottom axis. Note that the same results may be accomplished using the **Series axis assignment** control on the **Data scaling** page, as discussed above.

Date Label Frequency

The **Observations to label** group controls the frequency with which labels are displayed on the axis. For workfiles with a date structure, you may choose between **Automatic selection**, **Endpoints only**, **Every observation**, **Custom (Step = One obs)** where you specify an anchor position and number of steps between labels, and other **Custom** settings based on a date frequency. Only the first four settings are available for workfile scales that do not have a date structure. Some of the dated custom settings are not

available for workfiles with low frequencies (*e.g.*, Custom (Step = Quarter) is not available in annual workfiles).

When you select any of the custom settings, EViews offers you the **Alignment date** and **Steps** edit fields where you will fill in an alignment position and a step size. EViews will place a label at the alignment position, and at observations given by taking forward (and backward steps) of the specified size.

We illustrate custom date labeling by specifying 20 year label intervals for our LOGVOL series from our stock data workfile ("NYSEvolume.WF1"), by putting "1900" in the **Alignment date** edit field, and by entering "20" in the **Steps** field.
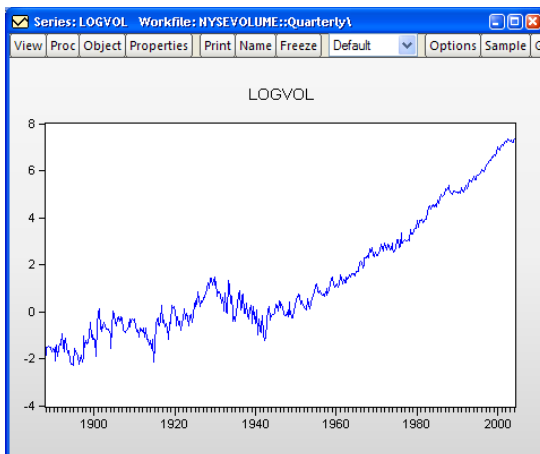
Date Label Formatting

The **Date format** section allows you to select the method of forming labels for the date axis. You may use the dropdown to select the default **Automatic**, in which EViews will choose an appropriate format, you can select from alternatives that display strings made up of various parts of the dates, or you may select **Custom**, which allows you to specify a fully custom date format string. Depending on your choice, you will be prompted for additional formatting information. You may also use the **End of period date labels** checkbox to tell EViews to use the end of period date for an observation when constructing its label.
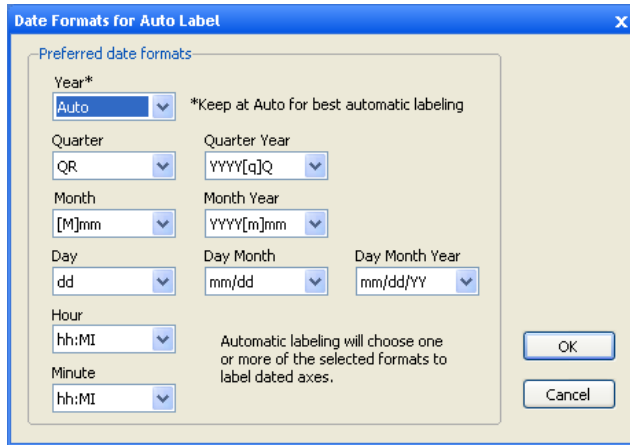
For example, using the default **Automatic** setting, EViews will provide automatic date formats using a default set of guidelines for displaying dates. If you would like to exert some control over the guidelines, you may click on the **Formats for Auto Labeling** button. This brings up the **Date Formats for Auto Label** dialog, in which you may specify your preferred formats without locking the graph to a specific date string.

Alternately, you may decide that you always want your labels to be formed using both the month and year of the observation. If you select **Month Year** from the dropdown, EViews will display an additional drop-down specifying how the month/year string is to be formatted. For date formats that include a year, click the **Use 4 digit year** checkbox to update all formats in the drop-down to a 4 digit display.

If you would like complete control over the display of your date strings, you should select **Custom**. The dialog will change to provide an edit field in which you should specify a date format using a standard EViews date format specifications in the **Format** edit field (see "Date Formats," on page 97 of the *Command and Programming Reference*).

If you would like EViews to
continue to provide auto-
matic date formats, but
would prefer some control
over its choices, leave the
default **Automatic** setting
and click the **Formats for
Auto Labeling** button. This
brings up the **Date Formats
for Auto Label** dialog, in
which you may specify your
preferred formats without
locking the graph to a spe-
cific date string.

Observation scales without a date structure are always labeled using the **Automatic** setting.

Date Label Positioning

The **Date label positioning** group determines whether
the labels are centered over period intervals, or
whether they are placed at the beginning of the inter-
val. You may change the setting in the dropdown menu from the default **Automatic label
and tick placement** to label the start or center of the period.

Label start of period - **tick on obs** instructs EViews to place the label at the beginning of the
period, and the tick on the observation. If you select **Center label on period** - **tick between
obs**, EViews places the ticks between observations and centers the label over the period.

The last option, **Center label on period** - **trim space at axis ends**, is essentially the same as
the previous option, **Center label on period** - **tick between obs**, except that the empty
space on either end of the date axis will be removed. The beginning of the first period and
end of the last period will fall at the edges of the graph frame, taking into account any indent
set in the **Size & Indents** page in the **Frame** option group.

Depending on the frequency of your graph, date labeling can be made clearer if we include a
second row of labels. Checking the **Allow two row date labels** box allows EViews to utilize
a second row of labels where appropriate. For example, daily data can be labeled first by
month, with a second row of labels indicating the year.

Axis Labels

The **Axis labels** group on the **Obs/Date axis** page allows you to modify label characteristics such as angle and font for the time axis. These options are similar to those discussed above for data axis labels. However, on the time axis, you can also customize the text of the labels.

The top half of the group mimics the behavior of the controls found in the **Axis labels** group on the **Data axis labels** page.

The bottom half of the group allows you to specify custom observation axis labels. See "Adding Custom Labels," beginning on page 773 for further discussion.
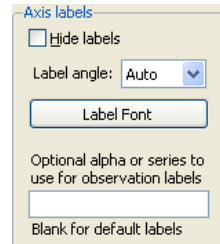
Tick Marks

The dropdown menu in the **Tick marks** section determines the placement of tickmarks: you may choose between **Ticks outside axis**, **Ticks inside axis**, **Ticks outside & inside**, and **No ticks**. The **Allow minor ticks** checkbox determines whether smaller ticks are placed between the major ticks.

Grid Lines

The final section of the **Axes & Scaling** group, **Grid Lines**, controls grid lines for all axes.
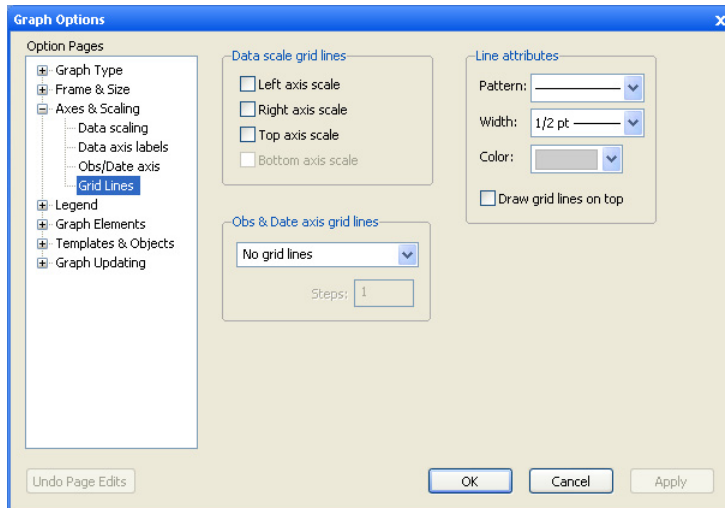
Data Scale Grid Lines

You may use the **Data scale grid lines** group to add grid lines to your graph by checking one or more of the four checkboxes. Left axis scale grid lines originate from the left vertical axis; those from the right correspond to the right vertical axis. Top and bottom grid lines are drawn from the corresponding horizontal axes.
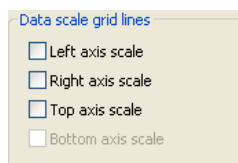
Notice that the **Bottom axis scale** checkbox is disabled in the image to the right. This indicates that the date scale for our current graph is assigned to the bottom axis, so the **Obs & Date axis grid lines** group should be used to control grid lines for this axis.

Note that if an axis associated with a specified grid line is not in use, the corresponding grid line option will be ignored.

Date Axis Grid Lines

The **Obs & Date axis grid lines** group applies grid lines to the observation or date axis. The first option in the dropdown, **No grid lines**, is self explanatory. The second option, **Automatic grid placement**, allows EViews to place grid lines at intervals it deems appropriate for the given frequency and range.
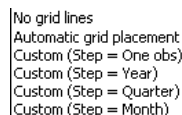
The remaining **Custom** options in the dropdown menu allow you to add grid lines at specified intervals, based on the frequency of your workfile. After specifying a step length in the dropdown, enter a value for the number of steps between grid lines in the **Steps** edit field.

For example, suppose we have data on monthly housing starts over the period 1990M01-1992M12 (a line graph of series HS in the workfile "House.WF1"). When we turn on grid lines and select **Automatic grid placement**, EViews opts to draw a grid line at the end of each year.
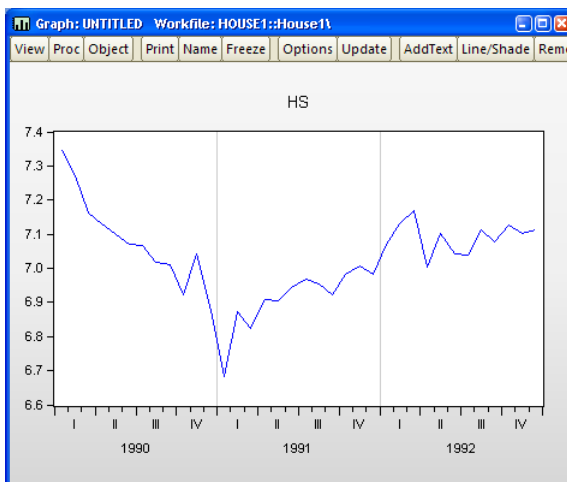
However, we might like the grid lines to show a bit more granularity. You may change the grid setting to **Custom (Step = Month)** and enter "2" in the **Steps** field. EViews then draws a graph with a grid line every other month.

Similarly, if we change the grid setting to **Custom (Step = Quarter)** and enter "2" in the **Steps** field, EViews will now display the graph with a grid line drawn semi-annually.
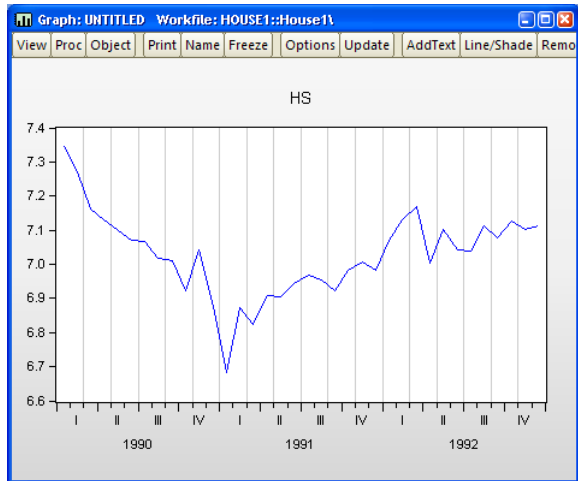
Grid Line Attributes

You may modify the appearance of grid lines in the **Line attributes** group. Use the **Pattern** dropdown to select a line pattern, the **Width** dropdown to modify the grid line width, and the **Color** dropdown to update the line color. The **Draw grid lines on top** option can be useful for filled graph types, such as bars or areas. Note that all grid lines will be modified with these settings.
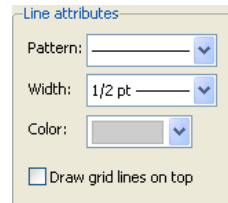
Legend

The next main category in the graph options dialog is **Legend**. This section provides control over the text, placement, and appearance of the legend.
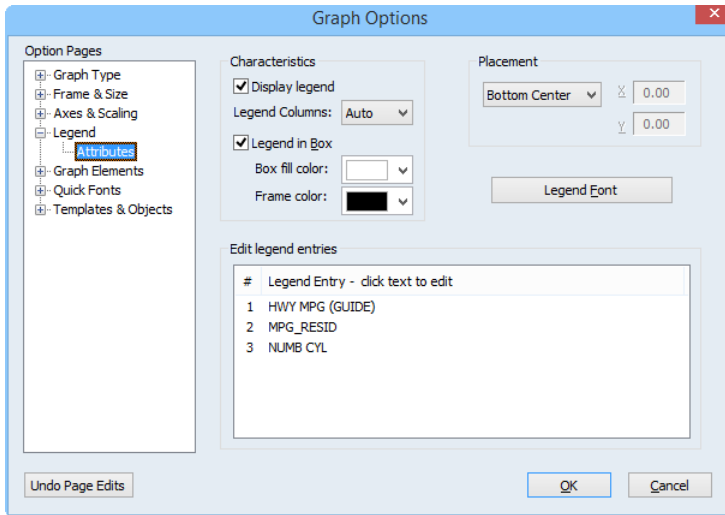
Legend Characteristics

You may elect to show or hide the legend and modify its appearance in the **Characteristics** group.

The first checkbox, **Display legend**, provides the most important option: to show or hide the legend. Click the checkbox to override EViews' default setting. Note that for single series graphs, the default is to hide the legend.

EViews attempts to display the legend entries in a visually appealing layout. To modify the number of columns, change the **Legend Columns** dropdown menu from the default **Auto** setting. Select an integer between **1** and **5**, and EViews will rearrange the legend entries with the corresponding number of columns and rows.

By default, EViews draws a box around legends. You may elect not to enclose the legend in a box by unchecking the **Legend in Box** checkbox. If a box is to be drawn, you may select the **Box fill color** and a box **Frame color** using the two color selectors.

Legend Placement

The legend may be positioned anywhere on the graph using the options presented in the **Placement** section. EViews provides a few common positions in the drop-down menu: **Bottom Center**, **Bottom Left**, **Bottom Right**, **Left**, **Right**, in addition to the **User** option. All locations are relative to the graph frame.

If **User** is selected, edit fields for the **X** and **Y** location are enabled. Note that the origin (0, 0) is the top left corner of the graph frame, and the coordinates are in units of virtual inches.

Individual graphs are always 4 x 3 virtual inches (scatter diagrams are 3 x 3 virtual inches) or a user-specified size, regardless if their current display size. It is often much easier to position the legend by dragging it to the desired location on the graph and not being concerned with its coordinates.

Note that if you place the legend using the **User** option, the relative position of the legend may change if you change the graph frame size. This does not include changing its size on the screen, but may occur if its size in virtual inches is changed via the **Size & Indents** section of the **Frame** options.

Legend Font

Click the **Legend Font** button to open the **Font** dialog and modify the text font and color.

Legend Text

To modify text shown in the legend, use the **Edit legend entries** section of the **Legend** page. Simply click on the text you wish to edit in the second column of the listbox, and alter the legend text in the edit field.
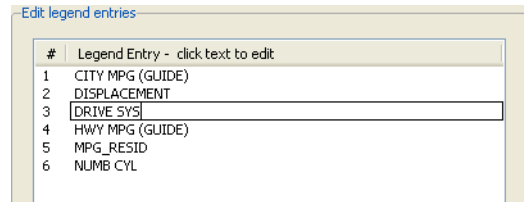
Lines and Symbols

For many graph types, the **Lines & Symbols** section under the **Graph Elements** group permits you to display your graph using lines only, lines and symbols, or symbols alone. In addition, you may specify various line and symbol attributes (color, line pattern, line width, symbol type and size).

To change the settings for your graph, click on the **Lines & Symbols** section under the **Graph Elements** group to show the line attributes. In the **Attributes** section you will see a list of the graph elements that you may change. Click on an element in the right-hand side of the dialog to access its settings, then use the dropdown menus to change its characteristics.
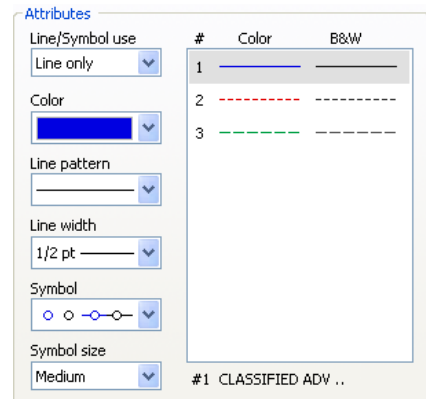
The **Line/Symbol** use dropdown determines the combination of lines and symbols used to display the selected element. You should change the **Line/Symbol use** dropdown to **Line & Symbo**l to display both lines and symbols, or **Symbol only** to suppress the lines. The corresponding **Color**, **Line pattern**, **Line width**, **Symbol**, and **Symbol size** dropdowns control the characteristics of the selected element. Note that the settings will sometimes have no

effect on the graph (*e.g.*, symbol choice if you are only displaying lines; line pattern if you are only displaying solid lines).

For the most part, the settings are self-explanatory; simply change the symbol and line use, size, and color of a graph element and the preview region will change to reflect your choices. Click on **Apply** or **OK** to apply your selections.

There are two **Symbol** choices that deserve explicit mention. The last two entries in the dropdown specify that the symbol should be the observation label from the workfile (the first entry uses the observation label itself; the second choice also includes a small circle with an attached text label). In some cases, these labels will be the dates, in other cases they will be integer values (1, 2, 3, ...), and in others, they will be text labels corresponding to the observations in the workfile. You may use this setting to display identifiers for each point in the observation graph.
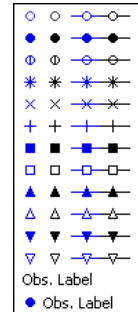
We illustrate this choice by displaying a scatterplot of the Matthews (2000) data on stork breeding pairs and number of births ("Storks.WF1").

Observation labels are displayed in the graph so that we may identify the data associated with each observation in the workfile. The graph shows immediately that the upper right-hand corner outlier is Turkey, and that, among others, Polish, Spanish, and Romanian storks have relatively low productivity.
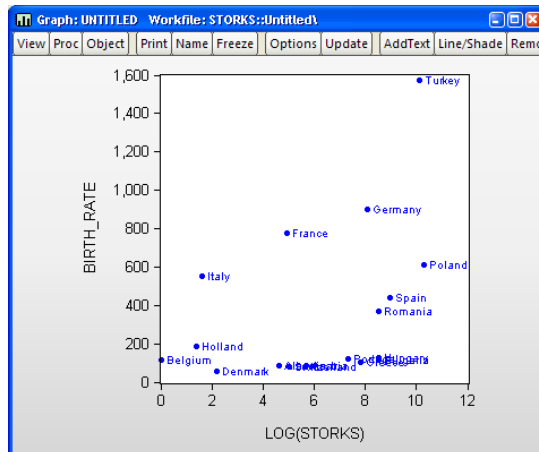
Once you have selected observation labels from the **Symbol** dropdown, you can modify their font attributes by clicking on the **Font** button in the **Observation Label** section on the left-hand side of the dialog. This button brings up a **Font** dialog, where you can set the font style, font size, and color of the observation labels.

The **Pattern use** section of the dialog requires a bit of discussion. By default, this option is set to **Auto choice**, meaning that EViews graphs will use different line pattern settings depending on whether you are outputting to a color or a black and white device; lines will be solid when shown on color devices (like your monitor), but will print with a pattern
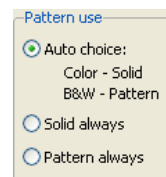
on a black-and-white printer. You may instead select **Solid always** or **Pattern always** so that the pattern of lines in the two types of devices always match.

The effect of different choices for this setting are shown in the **Attributes** section of the dialog, which shows what your graph elements will look when output to both types of devices. Our line graph example above uses the **Pattern always** setting so that the second and third lines are dashed when displayed on both color and black-and-white devices (for related preview tools, see "Color & Border" on page 649).

At times, you may notice that the data line in your line graph does not look perfectly smooth. To allow EViews to use a smoothing algorithm when drawing the line, make sure anti-aliasing is enabled. In the Anti-aliasing groupbox, selecting **Auto** will allow EViews to use anti-aliasing when it determines that doing so will make a visible difference. This choice is dependent primarily on the number of data points. To force anti-aliasing on or off when drawing lines, select **On** or **Off**, respectively.

If you wish to perform curve fitting on your data, you may use the **Interpolation** options at the bottom of the page. Select from **Linear** interpolation or a cardinal spline of varying degrees: **Mild**, **Medium**, or **Full**.

Fill Areas

You may use the **Fill Areas** section under the **Graph Elements** group to specify a fill color, gray shade (black-and-white representation of the fill color), or to add hatching to the fill region. The fill colors are used in graphs involving bars, areas, and area bands.
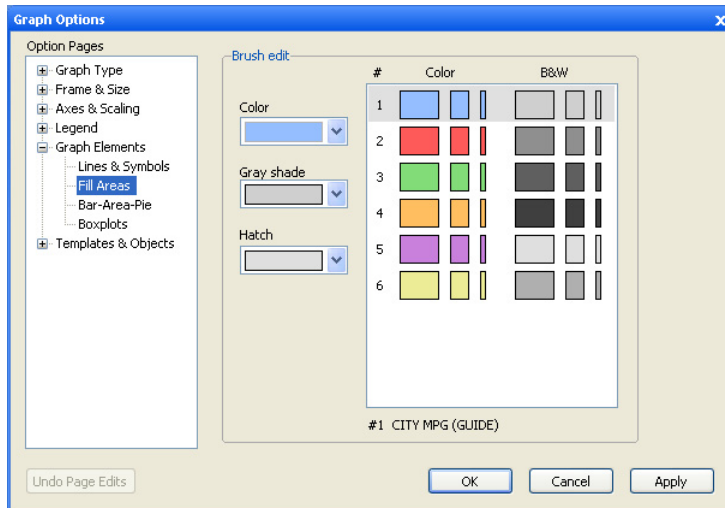
In the **Brush edit** section, simply click on one of the entries on the right to select the fill whose characteristics you wish to change (there are two in this example and the first is selected), then use the dropdown menus to alter the color, gray shade, and hatching as desired.

Note that the **Color** settings are used for fills that are displayed on a color device; the **Gray shade** dropdown controls the fill display when displayed on black-and-white devices. The preview and selection area on the right shows the characteristics of the fill element in both settings (for related preview tools, see "Color & Border" on page 649).

Additional options for Fill Areas

There are a few additional options for graphs with filled areas in the **Bar-Area-Pie** section of the **Graph Elements** group.
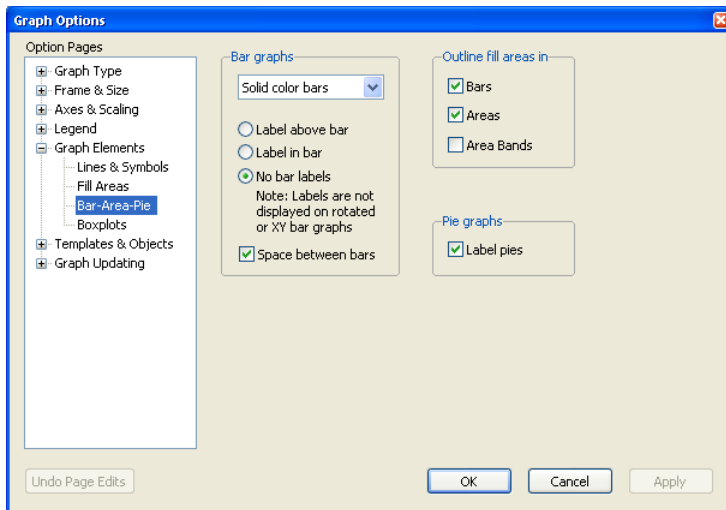
The **Bar graphs** section allows you to add value labels to bars, in addition to shading and a 3D appearance. These options are described in "Bar" on page 670.

You may choose to outline or not outline the fill regions for various fill types using the **Outline fill areas in** checkboxes. Here we see that **Bars** and **Areas** will be outlined but **Area Bands** will not.

Clicking **Label pies** in the **Pie graphs** group adds or removes observation labels from each pie in a pie graph.

Graph Types

The following is a description of the basic EViews graph types. We divide these graph types into three classes: *observation graphs* that display the values of the data for each observation; *analytical graphs* that first summarize the data, then display a graphical view of the summary results; *auxiliary graphs*, which are not conventional graph types, *per se,* but which summarize the raw data and display the results along with an observation graph of the original data.

The discussion for each type is limited to a basic overview of each graph type and does not discuss many of the ways in which the graphs may be customized (*e.g.,* adding histograms to the axes of line graphs or scatterplots;), nor does it describe the many ways in which the graphs are displayed when using multiple series or categorizations (*e.g.*, stacking; see "Basic Customization," beginning on page 648).

Observation Graphs

Observation graphs display the values of the data for each observation in the series or group. Some observation graphs are used for displaying the data in a single series (Line & Symbol, Area, Bar, Spike, Dot Plot, Seasonal Graphs), while others combine data from multiple series into a graph (Area Band, Mixed, Error Bar, High-Low(-Open-Close), Scatter, Bubble, XY Line, XY Bar, XY Area, Pie).

Line & Symbol

The line and symbol plot is a simple plot of the data in the series against observation identifiers. The plot shows data values as symbols, lines, or both symbols and lines.

To display a line and symbol plot for a single series or for each series in a group, select **View/Graph...** from the series or group menu to bring up the main **Graph Options** dialog, which will automatically open to the **Basic type** page. Then choose **Line & Symbol** in the **Specific** listbox.
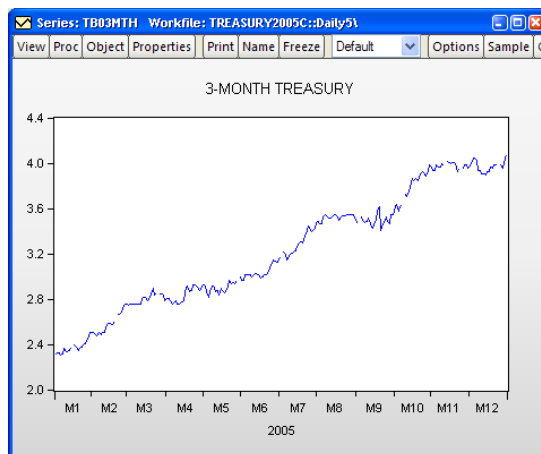
By default, EViews will display the data in the series using a line. To illustrate, we use the workfile "Treasury2005c.WF1" containing data on 2005 daily market yields for U.S. Treasury securities at constant maturities. The default line graph for the 3-month maturity series TB03MTH is depicted. If you look closely, you may be able to see gaps corresponding to holidays.

You may display your graph symbols alone, or using lines and symbols by clicking the **Lines & Symbols** section in the **Graph Elements** option group, and changing the desired attributes ("Lines and Symbols" on page 665). There are other settings for controlling color, line pattern, line width, symbol type, and symbol size that you may modify.

Area

Area graphs are observation graphs of a single series in which the data for each observation in the series is plotted against the workfile indicators. Successive observations are connected by a line, with the area between the origin and the line filled in.

To display an area graph of a single series or each series in a group, you should select **View/Graph...** from the series or group menu to display the **Graph Options** dialog, and then select **Area** in the **Specific** listbox.
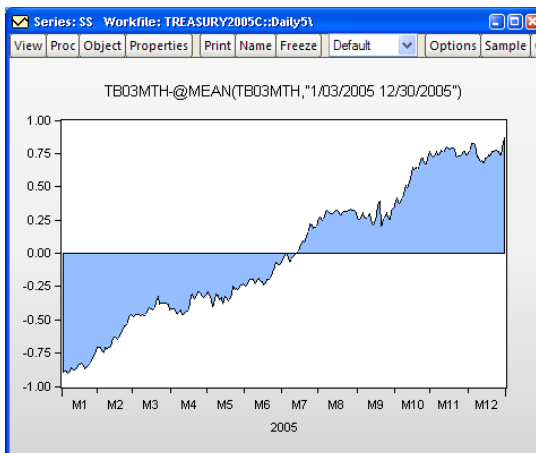
Our illustration depicts the area graph of the deviations of the 3-month Treasury bill series TB03MTH ("Treasury2005c.WF1") around the mean. Note that positive and negative regions use the same fill color, and that we have connected adjacent non-missing observations by checking the box labeled **Connect adjacent non-missing observations**.

Bar

The bar graph uses a bar to represent the value of each observation in a single series.

Bar graphs may be displayed for a single series or each series in a group by selecting **View/Graph...** from the series or group menu, and clicking **Bar** in the **Specific** listbox.

Our illustration shows a bar graph for the series TOTAL (from the
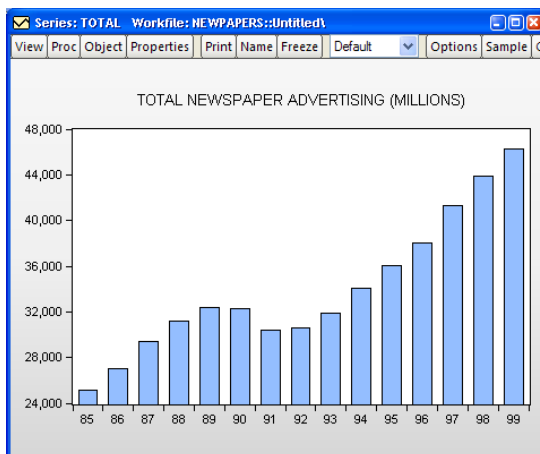
workfile "Newspapers.WF1") containing annual data on total advertising expenditures for the years 1985 to 1999.

Bar graphs are effective for displaying information for relatively small numbers of observations; for large numbers of observations, bar graphs are indistinguishable from area graphs since there is no space between the bars for successive observations.

You may add numeric value labels to your bars by double clicking on the bar to display the graph dialog, selecting the **Bar-Area-Pie** section in the **Graph Elements** option group, and select either **Label above bar**, or **Label in bar** in the **Bar graphs** section of the dialog page. EViews will add a label showing the height of the bar, provided that there is enough space to display the label.
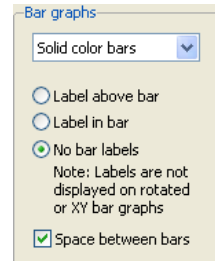
You may use the drop-down menu to apply fade effects to your bars. By default, EViews displays the **Solid color bars**, but you may instead choose to display **3D rounded bars**, **Fade: dark at zero**, **Fade: light at zero**. The latter two entries fade the bars from light to dark, with the fade finishing at the zero axis. Note that at press time, fades are not supported when exporting graphs to PostScript.

Moreover, while we discourage you from doing so, you may also use the **Fill Areas** page to remove the spacing and/or the outlines from the bars.

Here, we see the bar graph for the TOTAL newspaper advertising expenditures for the years 1997 to 1999, with value labels placed above the bar, and 3D rounded bars. It is worth pointing out that we restrict the sample to the three years, as the labels are not large enough to be visible when displaying lots of bars.
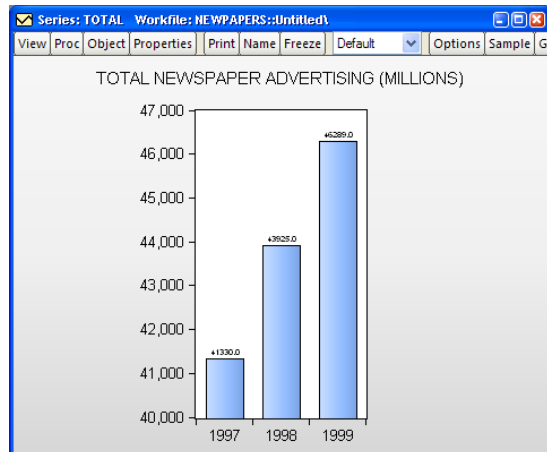
The **Fill Areas** section in the **Graph Elements** option group may be also used to change the basic characteristics of the fill area (color, gray shading, hatching, *etc.*). See "Fill Areas," on page 667 for details.

Area Band

The area band graph is used to display the band formed by pairs of series, filling in the area between the two. While they may be used in a number of settings, band graphs are most often used to display forecast bands or error bands around forecasts.

You may display an area band graph for any group object containing two or more series. Select **View/ Graph...** from the group menu, and then choose **Area Band** in the **Specific** listbox. The **Fill Areas** and
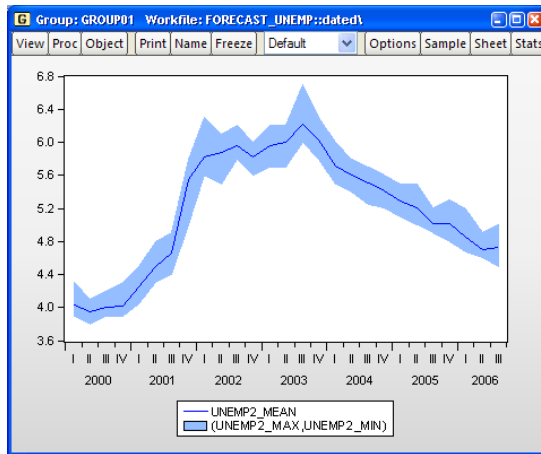
Lines & Symbols dialogs under the **Graph Elements** group may be used to modify the characteristics of the lines and shades in your graph.

EViews will construct bands from successive pairs of series in the group. If there is an odd number of series in the group, the final series will, by default, be plotted as a line.

Our example of the area band graph uses data from the Federal Reserve Bank of Philadelphia's Survey of Professional Forecasters ("Forecast_unemp.WF1"). UNEMP_MAX and UNEMP_MIN contain the high and low one-quarter ahead forecasts of the unemployment rate for each period in the workfile; UNEMP_MEAN contains the mean values over the individual forecasts. To construct this graph, we create a group GROUP01 containing (in order), the series UNEMP_MAX, UNEMP_MIN, and UNEMP_MEAN. Note that reversing the order of the first two series does not change the appearance of the graph.

Spike

The spike plot uses a bar to represent the value of each observation in a single series. Spike plots are essentially bar plots with very thin bars. They are useful for displaying data with moderate numbers of observations; settings where a bar graph is indistinguishable from an area graph.

To display a spike plot for a single series or for each series in a group, select **View/Graph...** from the series or group menu, and then choose **Spike** in the **Specific** listbox.
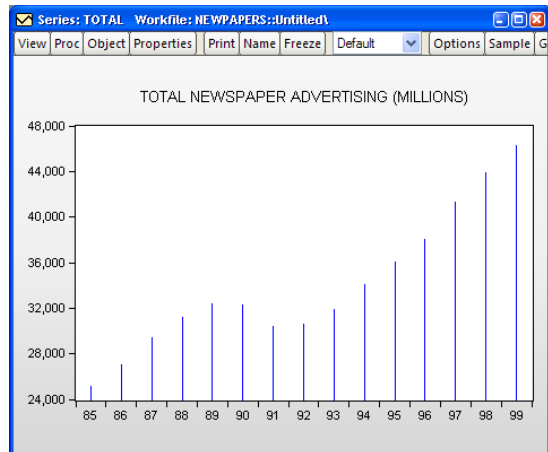
Our illustration shows a spike graph for the annual total newspaper advertising expenditure data in the series TOTAL in "Newspapers.WF1". It may be directly compared with the bar graph depiction of the same data (see "Bar" on page 670).

Note that for large numbers of observations, the spike graph is also indistinguishable from an area graph.

Mixed

This graph type combines line, bar, spike, area, or area band graphs. You may, for example, use the mixed graph to display multiple series in single graph, with the first series shown as a bar, spike, or area graph, or with the first two series displayed as an area band graph, with the remaining series depicted using lines.

To display a mixed plot, you must have with a group object containing two or more series. Select **View/Graph...** from the group menu, and then choose **Mixed** in the **Specific** listbox.
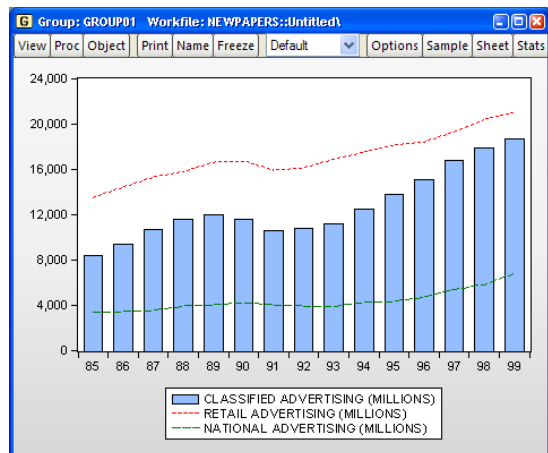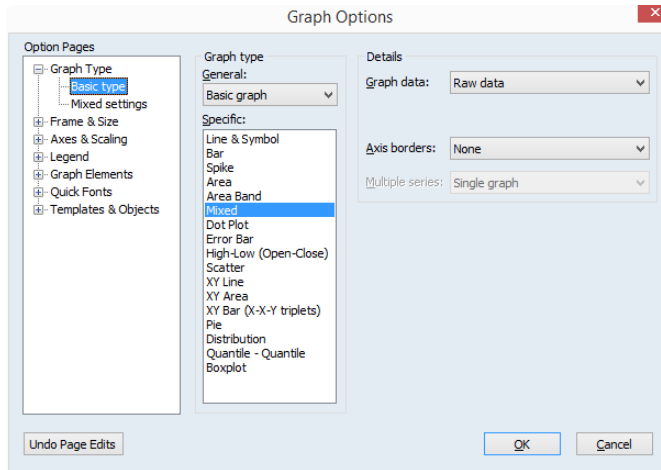
When you select **Mixed**, the left-hand side tree adds an additional page under **Graph Type** for **Mixed settings**. You will use this node to specify the types of graphs you wish to mix. When you click on this node, the right-hand side of the dialog changes to display the new settings:
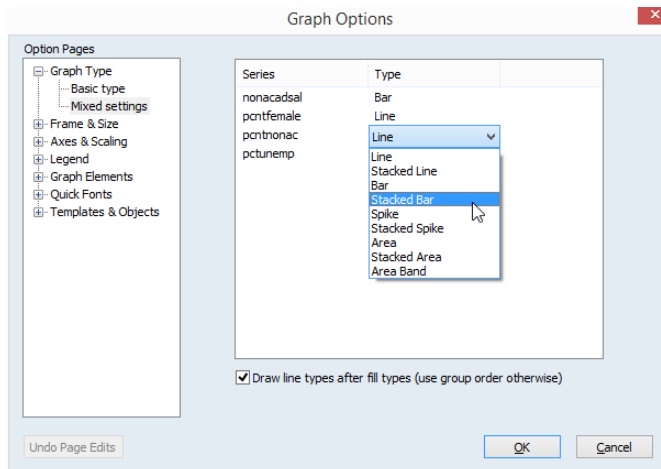
By default, the first series will be plotted as a bar and remaining series will be plotted as lines. You may use the dropdown menus to select different types for each series. If there is a type such as **Area Band** that requires more than one series, EViews will construct the graph using the series defined by a pair of series. Note that the pairs need not be contiguous.

In addition, for legibility, EViews will draw all of the line types after the fill types, but you may override this behavior using the checkbox.

Our illustration uses data from our newspaper advertising example ("Newspapers.WF1"). The data in GROUP01 are displayed as a **Mixed** graph, with the first series dropdown on the right-side of the dialog set to **Bar** and the remainder to **Line**.

Dot Plot

The dot plot is a symbol only version of the line and symbol graph ("Line & Symbol" on page 669) that uses circles to represent the value of each observation. It is equivalent to the **Line & Symbol** plot with the lines replaced by circles, and with a small amount of indenting to approve appearance.

Dot plots may be displayed for a single series or each series in a group by selecting **View/Graph...** from the series or group menu, and clicking **Dot Plot** in the **Specific** listbox.

Symbol options may be accessed using the **Lines & Symbols** dialog under the **Graph Elements** group.

Dot plots are often used with cross-section data. For example, using the series POP in the workfile "EUpop.WF1", we may produce a dot plot of the 1995 population of each of the 15 European Union members (as of 1995). With a bit of effort we can see that Germany is the clear population outlier.
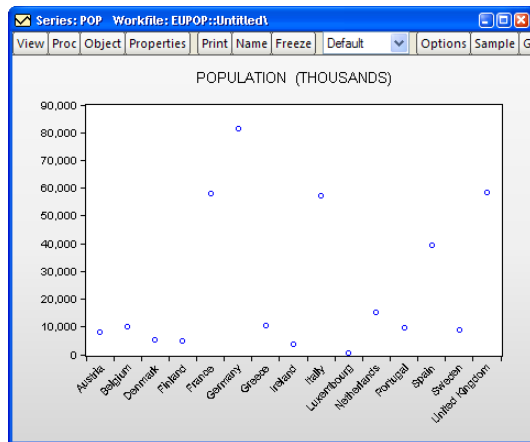
Dot plots are sometimes rotated so that the observations are on the vertical axis, often with horizontal gridlines. EViews provides easy to use tools for performing these and other modifications to improve the appearance of this graph ("Orientation," on page 624 and "Grid Lines," on page 661).

Error Bar

The error bar graph is an observa-
tion graph designed for displaying
data with standard error bands. As
with the related area band graph,
error bars are most often used to
display forecast intervals or error
bands.

The graph features a vertical error
bar connecting the values for the
first and second series. If the first
series value is *below* the second
series value, the bar will have out-
side half-lines. The optional third
series is plotted as a symbol.
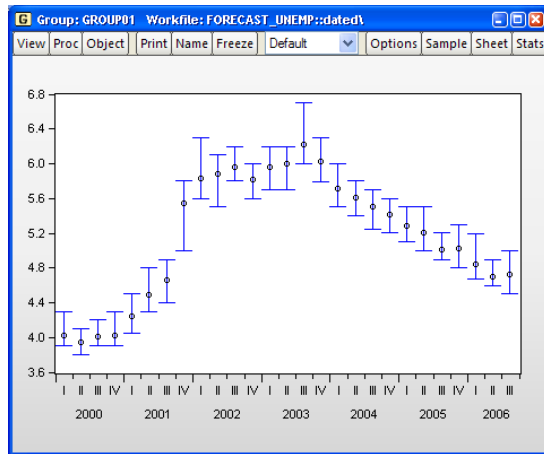
You may display an error bar graph for any group object containing two or more series; the
error bar will use data for, at most, the first three series. To display an error bar graph,
View/Graph... from the group menu, and then choose **Error Bar** in the **Specific** listbox.

Our illustration shows an error graph for the forecasting data in the group GROUP01 in
"Forecast_unemp.WF1". It may be directly compared with the area band graph of the same
data ("Area Band" on page 672).

High-Low (Open-Close)

The High-Low (Open-Close) is an
observation graph type commonly
used to display daily financial data.
As the name suggests, this chart is
commonly used to plot the daily
high, low, opening and closing val-
ues of asset prices.

The graph displays data for two to
four series. Data from the first two
series (the high-low values) will be
connected as a vertical line. If pro-
vided, the third series (the open
value) is drawn as a left horizontal
half-line, and the fourth series (the
close value) is drawn as a right horizontal half-line.
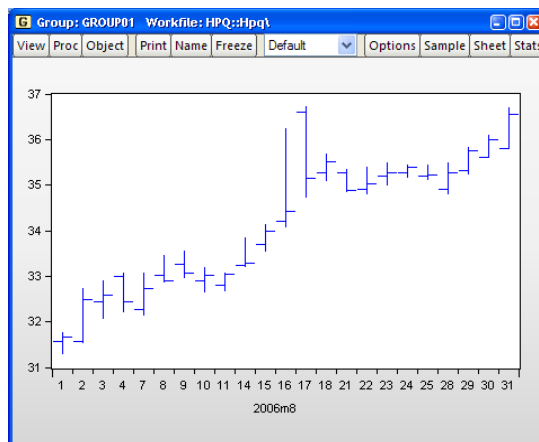
You may display a high-low graph for any group object containing two or more series. To display an high-low graph, **View/Graph...** from the group menu, and then choose **High-Low (Open-Close)** in the **Specific** listbox. Data for up to four series will be used in forming the graph.

We illustrate this graph type using daily stock price data for Hewlett-Packard (ticker HPQ) for the month of August, 2006 ("HPQ.WF1"). We display the graph for data in the group GROUP01 containing the series HIGH, LOW, OPEN, and CLOSE.

Scatter

A scatterplot is an observation graph of two series in which the values of the second series are plotted against the values of the first series using symbols. Scatterplots allow you to examine visually the relationship between the two variables.

We may display a scatterplot of a group containing two or more series by selecting **View/Graph...** from the main menu, and then selecting **Scatter** in the **Specific** listbox.
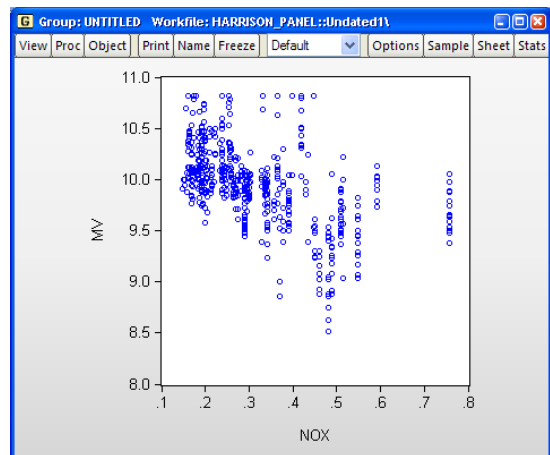
Our illustration uses data from the Harrison and Rubinfeld (1978) study of hedonic pricing ("Harrison_Panel.WF1"). The data consist of 506 census tract observations on 92 towns in the Boston.

We focus on the variables NOX, representing the average annual average nitrogen oxide concentration in parts per hundred million, and MV, representing the log of the median value of owner occupied houses (MV). We form the group SCATTER containing NOX and MV, with NOX the first series in the group since we wish to plot it on the horizontal axis. The scatter shows some evidence of a negative relationship between air pollution and house values.

Note that EViews provides tools for placing a variety of common graphs on top of your scatter (see "Auxiliary Graph Types," beginning on page 704).

Bubble plots

A bubble plot is an extension of a scatter plot where the third dimension may be used to specify the size of a data point. Unlike traditional scatter plots, where bubble sizes are fixed, bubble plots allow for variable size bubbles.

To create a bubble plot, select Bubble Plot as your Basic Type:

Bubble plots requires a minimum of three series (a series triplet). When creating a bubble plot from a group object, you have two ways of defining the triplet: **First vs All** or **XYZ triplets**.

First vs All assigns the first series of the group to the horizontal axis, uses the last series to determine the bubble size, and assigns all other series (from the second to the penultimate) to the vertical axis. Suppose, for example, we have a group consisting of the series X, Y1, Y2, Y3, and Z.

If we plot this as a bubble plot and select **Single graph - First vs All (XY..YZ)**, EViews will produce:
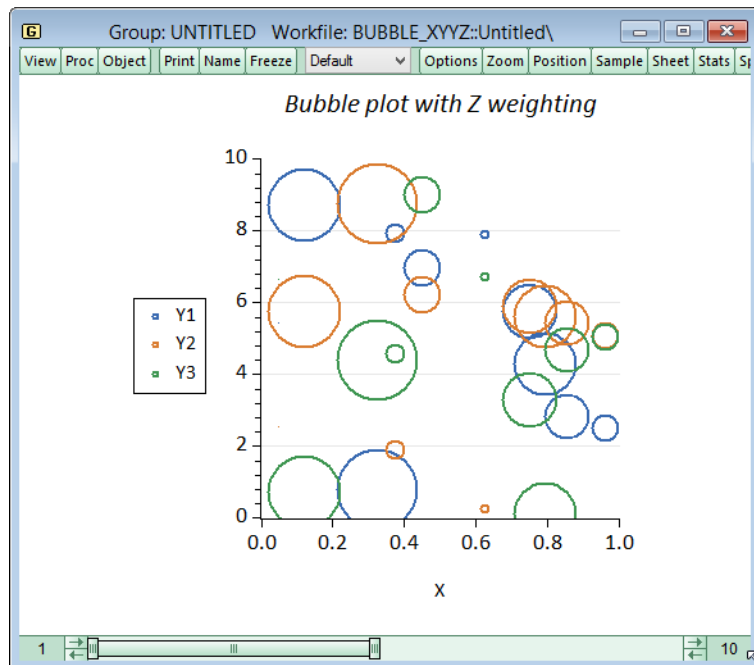
Alternatively, XYZ triplets uses successive series triplets in the group to create one plot. When using successive triplets, the first series of the triplet is assigned to the horizontal axis, the second series is assigned to the vertical axis and the last series of the triplet determines bubble size.

You may also choose to have all the triplets appear in one single graph or to create multiple graphs, with a single graph for each data triplet.

Below we have collected data on inflation, unemployment, and population from several European countries for 2016, and imaginary data for 2020.

We create a Phillips curve bubble plot for each year by selecting **Multiple graphs - XYZ triplet**.
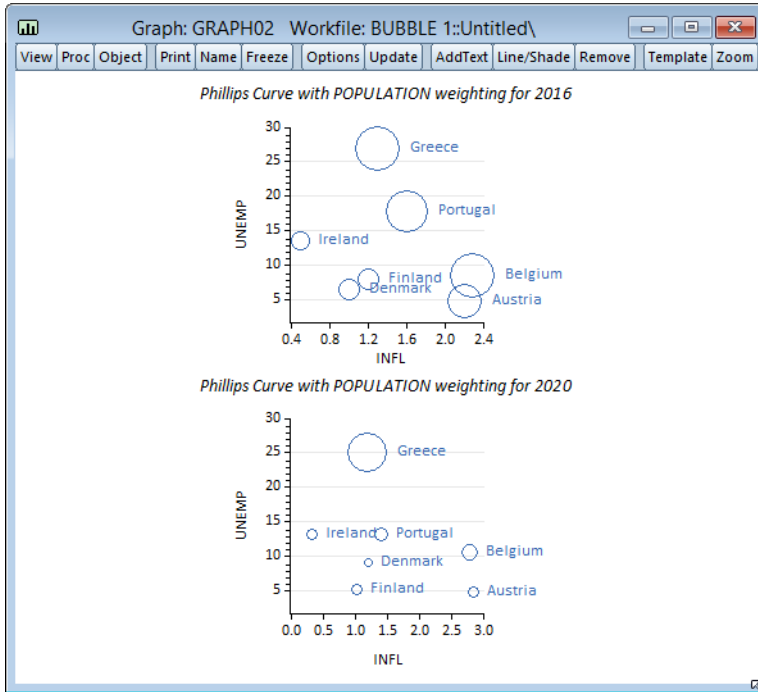
XY Line

An XY line graph is an observation graph of two series in which the values of the second series are plotted against the values of the first series, with successive points connected by a line.

XY line graphs differ from scatterplots both in the use of lines connecting points and in the default use of a 2:1 aspect ratio.
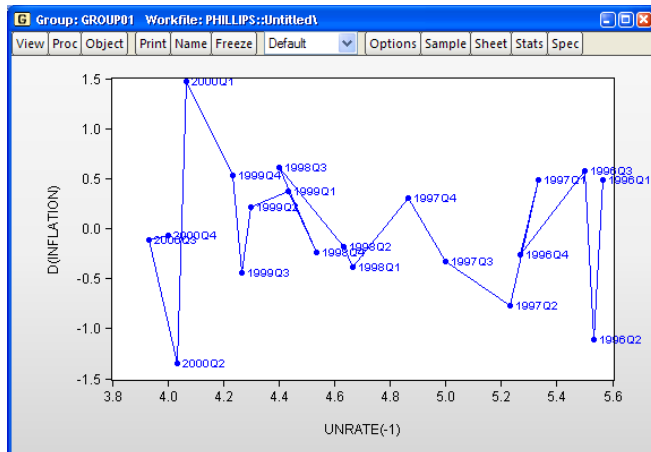
To display a XY line graph we first open a group containing two or more series, then select **View/Graph...** main menu, and then choose **XY Line** in the **Specific** listbox.

As with the scatterplot, EViews provides tools for placing a variety of common graphs on top of your XY line graph (see "Auxiliary Graph Types," beginning on page 704).

Our illustration uses data on unemployment rates and inflation for the U.S. from 1996 through 2000. Following the discussion in Stock and Watson (2007), we plot the change in the inflation rate against the previous period's unemployment rate; to make it easier to see the ordering of the observations, we have turned on observation labeling ("Lines and Symbols" on page 665).

XY Area

The XY area graph is an observation graph of two series in which the values of the second series are plotted against the values of the first series. In contrast with the scatterplot, successive points are connected by a line, and the region between the line and the zero horizontal axis is filled. Alternately, one may view the XY area graph as a filled XY line graph (see"XY Line").

To display a XY area graph we first open a group containing two or more series, then select **View/Graph...** from the main menu, and then choose **XY Area** in the **Specific** listbox.
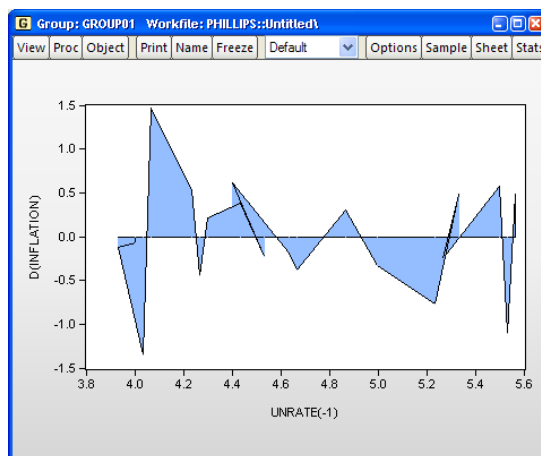
We may customize the graph by changing display characteristics using the **Lines & Symbols** and **Fill Areas** dialogs under the **Graph Elements** dialog group.

Our illustration of the XY area graph uses data on U.S. unemployment as discussed in "XY Line," on page 680. Note that the example graph is not particularly informative as XY area graphs are generally employed when the values of the data in the X series are monotonically increasing. For example, XY area graphs are the underlying graph type that EViews uses to display filled distribution graphs.

XY Bar (X-X-Y triplets)

XY bar graphs display the data in sets
of three series as a vertical bar. For a
given observation, the values in the
first two series define a region along
the horizontal axis, while the value in
the third series defines the vertical
height of the bar. While technically an
observation graph since every data
observation is plotted, this graph is
primarily used to display summary
results. For example, the XY bar is the
underlying graph type used to display
histograms ("Histogram," on
page 685).
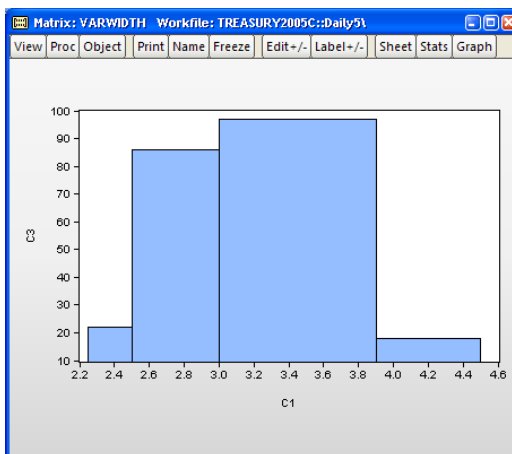
Our illustration uses the XY bar graph to create a variable width histogram for the 3-month
Treasury security data from "Treasury2005c.WF1". We first use **Proc/Generate by Classifica-
tion...** to divide the series into categories defined by the intervals [2.25, 2.5), [2.5, 3), [3,
3.9), [3.9, 4.5). The classified series is given by TB03MTH_CT. The frequency view of this
series is given by:

Tabulation of TB03MTH_CT
Date: 10/05/06   Time: 12:54
Sample: 1/03/2005 12/30/2005
Included observations: 250
Number of categories: 4

Value	Count	Percent	Cumulative Count	Cumulative Percent
[2.25, 2.5)	22	8.80	22	8.80
[2.5, 3)	86	34.40	108	43.20
[3, 3.9)	97	38.80	205	82.00
[3.9, 4.5)	45	18.00	250	100.00
Total	250	100.00	250	100.00

Next, we use the data in this table to create a matrix. We want to use a matrix instead of a
series in the workfile since we want each row to correspond to a bin in the classification.
Accordingly, we create a 4×3 matrix VARWIDTH where the first column contains the low
limit points, the second column contains the high limit points, and the last column contains
the number of observations that fall into the interval.

To display the XY bar graph shown in the example illustration, select **View/Graph...** from
the matrix main menu, and then choose **XY Bar (X-X-Y triplets)** in the **Specific** listbox.

Pie

This graph is an observation graph where each observation is pictured as a pie chart, with the wedges of the pie representing the series value as a *percentage* of the group total. (If a series has a negative or missing value, the series value will be dropped from the calculation for that observation.)

Pie graphs are available for groups containing two or more series. To display the graph, select **View/ Graph...** from the group menu, and then select **Pie** in the **Specific** list-box.
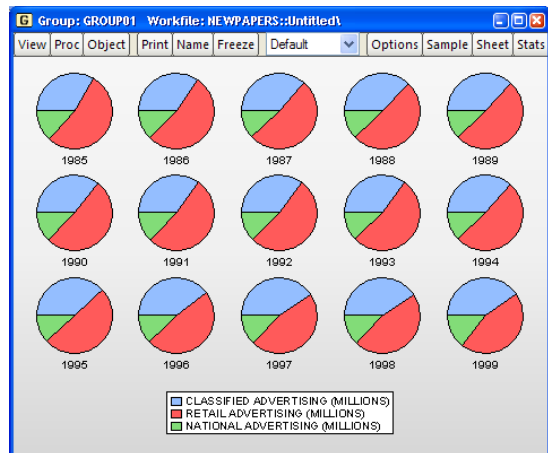
You may choose to label each pie with an observation number. To change the setting from the default value, select the **Bar-Area-Pie** dialog from the **Graph Elements** group, and select or unselect the **Label pies** option in the **Pie graphs** section of the page.

Our illustration uses the newspaper advertising revenue data ("Newspapers.WF1"). The three series in GROUP01, CLASSIFIED, RETAIL, and NATIONAL, are the three components of TOTAL advertising revenue. Each pie in the graph shows the relative proportions; retail is the dominant component, but its share has been falling relative to classified.

Seasonal Graphs

Seasonal graphs are a special form of line graph in which you plot separate line graphs for each season in a regular frequency monthly or quarterly workfile.

To display a seasonal graph for a single series or for each series in a group, select **View/ Graph...** from the series or group menu, and then choose **Seasonal Graph** in the **Specific** listbox. Note that if your workfile does not follow a monthly or quarterly regular frequency, **Seasonal Graph** will not appear as a specific graph type.

When you select **Seasonal Graph,** the right-hand side of the page changes to provide a **Seasonal type** dropdown containing two options for displaying the graph. The first option, **Paneled lines & means**, instructs EViews to divide the graph into panels, each of which will contain a time series for a given season. If, for example, we have a monthly work-

file, the graph will be divided into 12 panels, the first containing a time series of observations for January, the second containing a time series for February, *etc.* The second option, **Multiple overlayed lines**, overlays the time series for each season in a single graph, using a common date axis.
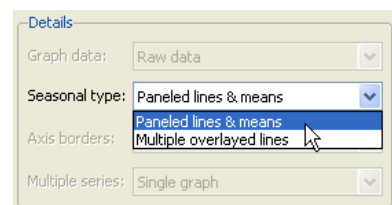
To see the effects of these choices, we consider two examples of seasonal graphs. The EViews workfile "Starts_cust.WF1" contains Census Bureau data on monthly new residential construction in the U.S. (not seasonally adjusted) from January 1959 through August 2006. We will consider the series TOTAL containing data on the total of new privately owned housing starts (in thousands) for the subsample from January 1990 through August 2006.

We first display a **Paneled lines & means** seasonal graph for the series TOTAL. Note that the graph area is divided into panels, each containing a time series for a specific month. The graph also contains a set of horizontal lines marking the seasonal means.

It is easy to see the seasonal pattern of housing starts from this graph, with a strong reduction in housing starts during the fall and winter months. The mean of January starts is a little over 120 thousand units, while the mean for May starts is around 180 thousand.
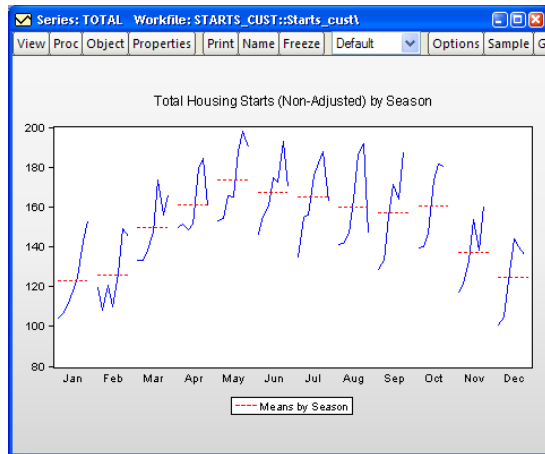
We may contrast this form of the seasonal graph with the **Multiple overlayed lines** form of the seasonal graph. The differences in the individual time series lines provide a different form of visual evidence of seasonal variation in housing starts. The overlayed form of the seasonal graph makes it easier to compare values for a given period.

Here, we see that January values for housing starts are roughly two-thirds of their summer counterparts.
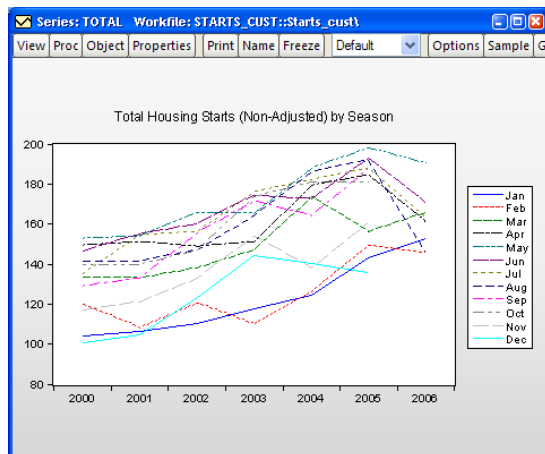
Analytical Graph Types

Analytical graphs are created by first performing data reduction or statistical analysis on the series or group data and then displaying the results visually. The central feature of these graphs is that they do not show data for each observation, but instead display a summary of the original data.

The following is a brief summary of the characteristics of each of these graph types. Unless otherwise specified, the examples use data on three month CD rate data for 69 Long Island banks and thrifts ("CDrate.WF1"). These data are used as an example in Simonoff (1996).

Histogram

The histogram graph view displays the distribution of your series in bar graph form. The histogram divides the horizontal axis into equal length intervals or bins, and displays a count or fraction of the number of observations that fall into each bin, or an estimate of the probability density function for the bin.

To display a histogram for a single series or for each series in a group, select **View/Graph...** from the series or group menu, then select **Distribution** from the **Specific** listbox. The right-hand side of the dialog will change to show **Distribution** options. Select **Histogram** from the drop-down (the default).
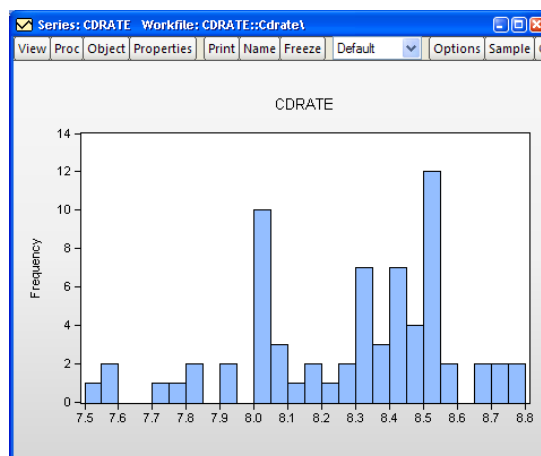
(Note that specialized tools also allow you to place histograms along the axes of various graph types.)

When you select **Histogram**, EViews displays an **Options** button that opens the **Distribution Plot Customize** dialog. This dialog allows you to customize your histogram estimate or to add additional distribution graphs. You may, for example, add a fitted theoretical distribution plot or kernel density to the histogram.

Adding additional graph elements may be done using the **Add** button in the **Added Elements** section of the dialog. As you add elements, they will appear in the listbox on the left. You may select any graph element to display its options on the right-hand side of the page. In this example, there is only the single histogram graph element (which is selected), and the dialog shows the options for that histogram.
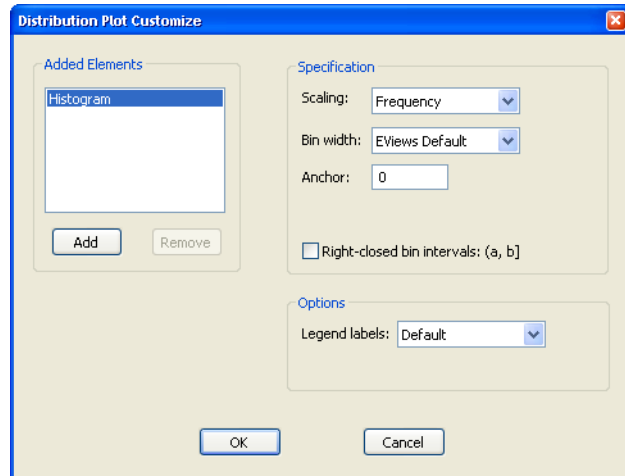
First, the **Scaling** dropdown menu lets you choose between showing the count of the number of observations in a bin (**Frequency**), an estimate of the density in the bin (**Density**), and the fraction of observations in each bin (**Relative frequency**). The density estimates are computed by scaling the relative frequency by the bin width so that the area in the bin is equal to the fraction of observations.

Next, **Bin Width** and **Anchor** specify the construction of the bin intervals. By default, EViews tries to create bins that are defined on "nice" numbers (whole numbers and simple fractions). These estimates do not have any particular statistical justification.

Simple data based methods for determining bin size have been proposed by a number of authors (Scott 1979, 1985a; Silverman 1986; Freedman-Diaconis 1981). The supported methods all choose a bin width h that minimizes the integrated mean square error of the approximation (IMSE) using the formula, $h = a\hat{\sigma}N^{-1/3}$:

- Normal (Sigma): $a = 3.49$, $\hat{\sigma} = s$
- Normal (IQR): $a = 3.49$, $\hat{\sigma} = IQR/1.34$
- Silverman: $a = 3.49$, $\hat{\sigma} = \min(s, IQR/1.34)$
- Freedman-Diaconis: $a = 2.0$, $\hat{\sigma} = IQR$

where s is the sample standard deviation, IQR is the interquartile range, and N is the number of observations.

For our example data, displaying a density histogram of the CDRATE data using the Normal (Sigma) bin-width method shows a histogram with considerably fewer bins and modified vertical axis scaling. One could argue that the shape of the CDRATE distribution is more apparent in this plot, at the cost of detail on the number of observations in easily described categories.

It is well-known that the appearance of the histogram may be sensitive to the choice of the anchor (see, for example, Simonoff and Udina, 1997). By default, EViews sets the anchor position for bins to 0, but this may be changed by entering a value in the **Anchor** edit box.
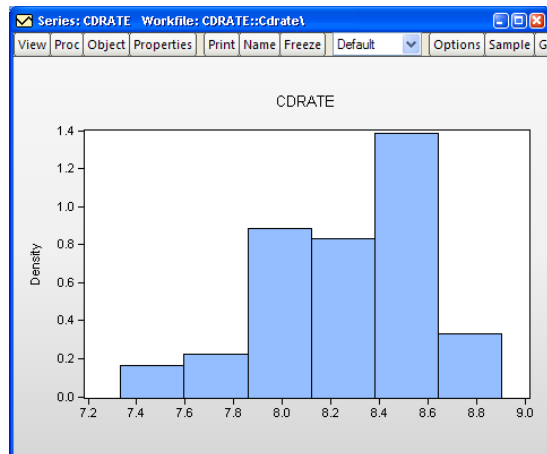
The **Right-closed bin intervals** checkbox controls how observations that equal a bin endpoint are handled. If you select this option, observations equal to the right-endpoint of a bin will be classified as being in the bin, while observations equal to the left-endpoint will be placed in the previous bin.

By default, EViews provides the minimum legend information sufficient to identify the graph elements. In some instances, this means that no legend is provided; in other cases, the legends may be rather terse. The **Legend labels** dropdown menu allows you to override this setting; you may elect to display a short legend (**Short**), to display detailed information (**Detailed**), or to suppress all legend information (**None**).

Histogram Polygon

Scott (1985a) shows that the histogram polygon (frequency polygon), which is constructed by connecting the mid-bin values of a histogram with straight lines, is superior to the histogram for estimating the unknown probability density function.

To display a histogram polygon for a single series or for each series in a group, select **View/Graph...** from the series or group menu, then select **Distribution** from the **Specific** listbox. Then choose **Histogram Polygon** from the **Distribution** options on the right-hand side.
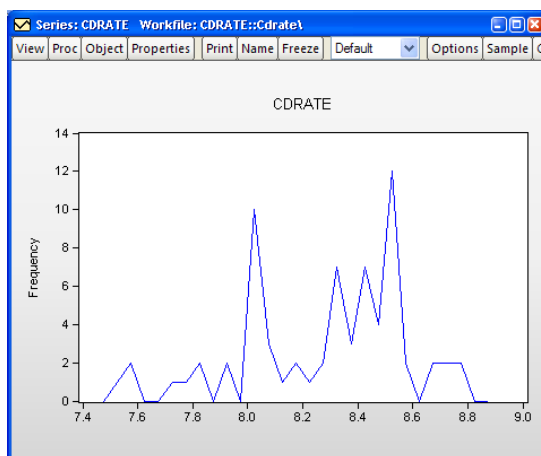
We use the default settings to display the frequency polygon for the three-month CD rate data. The EViews defaults, which were designed to generate easy to interpret histogram intervals, undersmooth the data.

You may control the computation of the histogram polygon by clicking on the **Options** button, and filling out the resulting dialog. In addition to all of the options described in "Histogram" on page 685, you may instruct EViews to display the fill the area under the polygon by clicking on the **Fill area** checkbox.

Note that the data based methods for determining bin size differ from those for the frequency polygon. The bandwidth is chosen as in the frequency polygon with $a = 2.15$ for the **Normal (Sigma)**, **Normal (IQR)**, and **Normal (Silverman)** methods, and $a = 1.23$ for **Freedman-Diaconis**. The constant factor in the Freedman-Diaconis is a crude adjustment that takes the histogram value for a and scales it by the ratio of the normal scaling factors for the frequency polygon and the histogram $(2.15/3.49)$.

Histogram Edge Polygon

Jones, *et al.* (1998) propose a modification of the frequency polygon that joins the bin right-edges by straight lines. This modification generates a smoothed histogram that improves on the properties of the frequency polygon.

To display a edge polygon, select **View/Graph…** from the series or group menu, then select **Distribution** from the **Specific** listbox. Then choose **Histogram Edge Polygon** from the **Distribution** drop-down to the right.
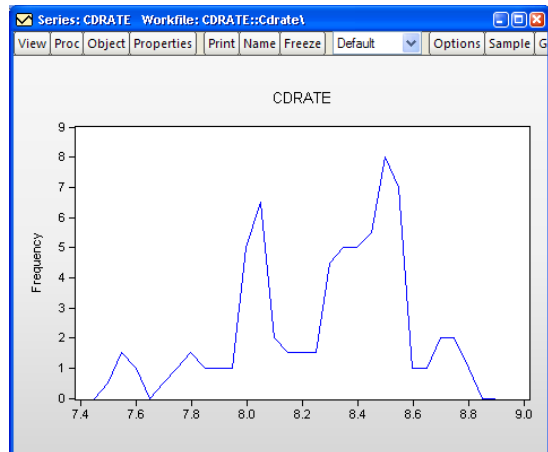
The default edge frequency graph for the CD rate data is displayed here. The EViews defaults, which were designed to generate easy to interpret histogram intervals, appear to undersmooth the data.

You may control the computation of the histogram polygon by clicking on the **Options** button, and filling out the resulting dialog. All of the options are described in "Histogram Polygon" on page 688.

Note that the data based methods for determining bin size generate different bin widths than those for the frequency polygon. The bandwidth is chosen as in the histogram and frequency polygon with $a = 1.50$ for the **Normal (Sigma)**, **Normal (IQR)**, and **Normal (Silverman)** methods, and $a = 0.86$ for **Freedman-Diaconis**.

Average Shifted Histogram

The average shifted histogram (ASH) is formed by computing several histograms with a given bin width but different bin anchors, and averaging these histograms (Scott, 1985b). By averaging over shifted histograms, the ASH minimizes the impact of bin anchor on the appearance of the histogram.

Scott (1985b) notes that the ASH retains the computational simplicity of the histogram, but approaches the statistical efficiency of a kernel density estimator. EViews computes the frequency polygon version of the ASH, formed by connecting midpoints of the ASH using straight lines.
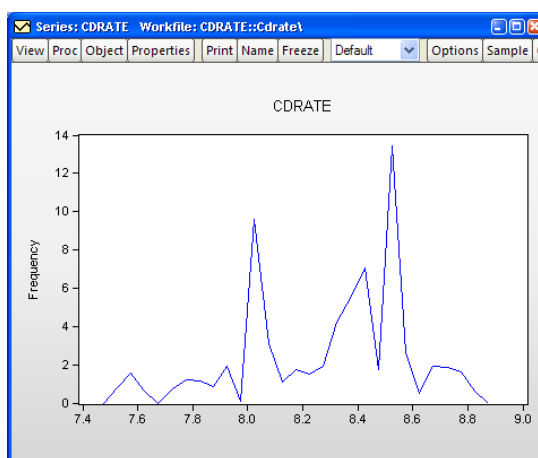
To compute an ASH, select **View/Graph...** from the series or group menu, then select **Distribution** from the **Specific** listbox. Then choose **Average Shifted Histogram** from the **Distribution** drop-down to the right.

The default ASH for the Long Island CD rate data is displayed above. The EViews defaults, which were designed to generate easy to interpret histogram intervals, undersmooth the data.

When you select **Average Shifted Histogram**, EViews displays an **Options** button that opens the **Distribution Plot Customize** dialog allowing you to customize your ASH or to add additional distribution graphs (see "Histogram" on page 685 for a discussion of the latter topic).

Almost all of the settings on the right-hand side of the dialog are familiar from our discussion of histograms.
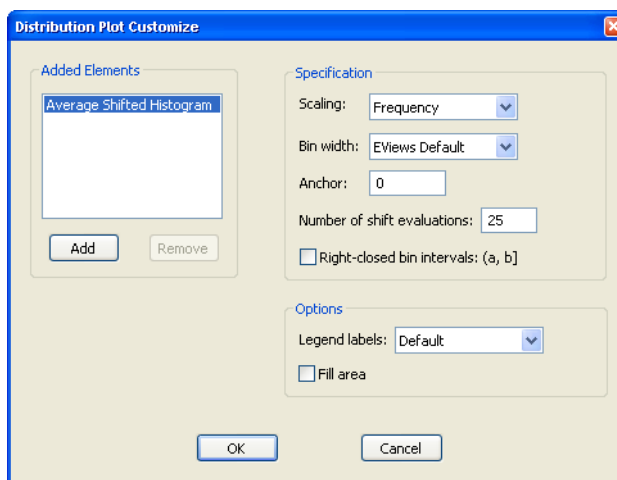
The only new setting is the edit box for the **Number of shift evaluations**. This setting controls the number of histograms over which we average. By default, EViews will compute 25 shifted histograms.

Kernel Density

The kernel density graph displays a kernel density estimate of the distribution of a single series. Heuristically, the kernel density estimator is an adjusted histogram in which the "boxes' the histogram are replaced by "bumps" that are smooth (Silverman, 1986). Smoothing is done by putting less weight on observations that are further from the point being evaluated. Specifically, the kernel density estimate of a series X at a point x is estimated by:

$$f(x) \; = \; \frac{1}{Nh} \sum_{i\,=\,1}^{N} K\!\left(\frac{x - X_i}{h}\right), \tag{13.1}$$

where N is the number of observations, h is the bandwidth (or smoothing parameter) and K is a kernel weighting function that integrates to one.

To compute and display a kernel density estimate for a single series or for each series in a group, select **View/Graph...** from the series or group menu, then choose **Distribution** in the **Specific** listbox. The right-hand side of the dialog page will change to provide a **Distribution** dropdown menu prompting you to choose a distribution graph. You should select **Kernel Density**.

(Note also that specialized tools allow you to place kernel density plots along the axes of various graph types.)
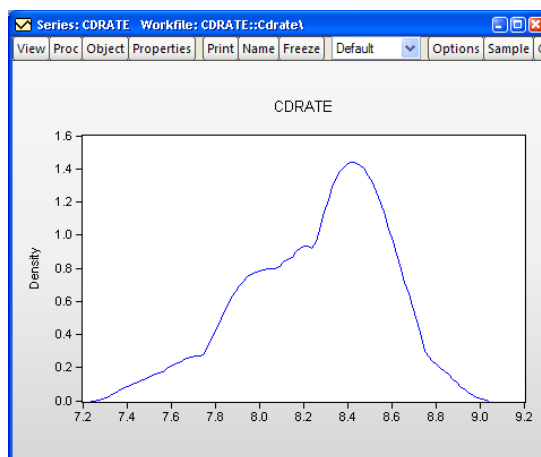
The default kernel density estimate for the CD rate data (see "Histogram" on page 685) is depicted above.

When you select **Distribution/Kernel Density**, EViews displays an **Options** button that opens the **Distribution Plot Customize** dialog. This dialog allows you to customize your kernel density estimate, or to add additional distribution graphs. You may, for example, choose a different kernel function, or a different bandwidth, or you may add a histogram or fitted theoretical distribution plot to the graph.
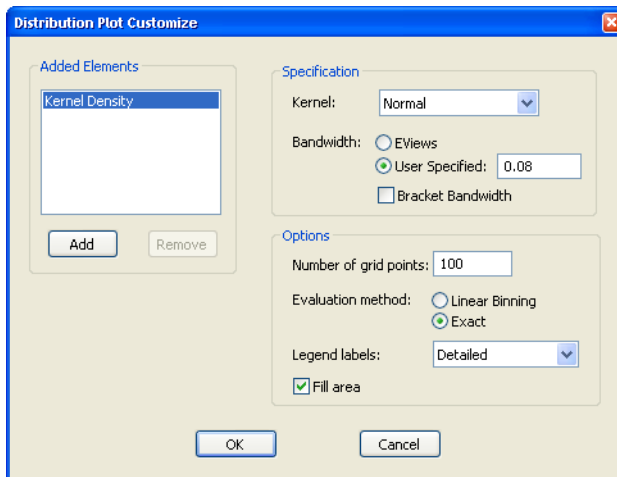
Adding additional graph elements may be done using the **Add** button in the **Added Elements** section of the dialog.

The **Specification** section of the dialog allows you to specify your kernel function and bandwidth selection:

- **Kernel**. The kernel function is a weighting function that determines the shape of the bumps. EViews provides the following options for the kernel function K:

Epanechnikov (default)	$\frac{3}{4}(1 - u^2)I(|u| \leq 1)$
Triangular	$(1 - |u|)(I(|u| \leq 1))$
Uniform (Rectangular)	$\frac{1}{2}(I(|u| \leq 1))$
Normal (Gaussian)	$\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}u^2\right)$
Biweight (Quartic)	$\frac{15}{16}(1 - u^2)^2 I(|u| \leq 1)$
Triweight	$\frac{35}{32}(1 - u^2)^3 I(|u| \leq 1)$
Cosinus	$\frac{\pi}{4}\cos\left(\frac{\pi}{2}u\right)I(|u| \leq 1)$

where u is the argument of the kernel function and I is the indicator function that takes a value of one if its argument is true, and zero otherwise.

- **Bandwidth**. The bandwidth h controls the smoothness of the density estimate; the larger the bandwidth, the smoother the estimate. Bandwidth selection is of crucial importance in density estimation (Silverman, 1986), and various methods have been suggested in the literature. The **Silverman** option (default) uses a data-based automatic bandwidth:

$$h = 0.9kN^{-1/5}\min(s, (IQR)/1.34) \tag{13.2}$$

where N is the number of observations, s is the standard deviation, and IQR is the interquartile range of the series (Silverman 1986, equation 3.31). The factor k is a canonical bandwidth-transformation that differs across kernel functions (Marron and Nolan 1989; Härdle 1991). The canonical bandwidth-transformation adjusts the bandwidth so that the automatic density estimates have roughly the same amount of smoothness across various kernel functions.

To specify a bandwidth of your choice, click on the **User Specified** option and type a nonnegative number for the bandwidth in the corresponding edit box. Although there is no general rule for the appropriate choice of the bandwidth, Silverman (1986, section 3.4) makes a case for undersmoothing by choosing a somewhat small bandwidth, since it is easier for the eye to smooth than it is to unsmooth.

The **Bracket Bandwidth** option allows you to investigate the sensitivity of your estimates to variations in the bandwidth. If you choose to bracket the bandwidth, EViews plots three density estimates using bandwidths $0.5h$, h, and $1.5h$.

The remaining options control the method used to compute the kernel estimates, the legend settings, and whether to fill the area under the estimate:

- **Number of Points**. You must specify the number of points M at which you will evaluate the density function. The default is $M = 100$ points. Suppose the minimum and maximum value to be considered are given by X_L and X_U, respectively. Then $f(x)$ is evaluated at M equi-spaced points given by:

$$x_i = X_L + i \cdot \left(\frac{X_U - X_L}{M}\right), \text{ for } i = 0, 1, \ldots M - 1. \tag{13.3}$$

EViews selects the lower and upper evaluation points by extending the minimum and maximum values of the data by two (for the normal kernel) or one (for all other kernels) bandwidth units.

- **Method**. By default, EViews utilizes the **Linear Binning** approximation algorithm of Fan and Marron (1994) to limit the number of evaluations required in computing the density estimates. For large samples, the computational savings are substantial.

The **Exact** option evaluates the density function using all of the data points for each X_j, $j = 1, 2, \ldots, N$ for each x_i. The number of kernel evaluations is therefore of order $O(NM)$, which, for large samples, may be quite time-consuming.

Unless there is a strong reason to compute the exact density estimate or unless your sample is very small, we recommend that you use the binning algorithm.

- **Legend labels**. This dropdown menu controls the information placed in the legend for the graph. By default, EViews uses a minimalist approach to legend labeling; information sufficient to identify the estimate is provided. In some cases, as with the kernel density of a single series, this implies that no legend is provided. You may elect instead to always display a short legend (**Short**), to display detailed kernel choice and bandwidth information (**Full**), or you may elect to suppress all legend information (**None**).

- **Fill area**. Select this option if you wish to draw the kernel density as a filled line graph.

This density estimate for the CD rate data seems to be oversmoothed. Simonoff (1996, chapter 3) uses a Gaussian kernel with bandwidth 0.08. To replicate his results, we fill out the dialog as follows: we select the **Normal** (Gaussian) kernel, specify a bandwidth of 0.08, select the **Exact** evaluation method (since there are only 69 observations to evaluate the kernel), select **Detailed** from **Legend labels**, and check the **Fill area** checkbox.
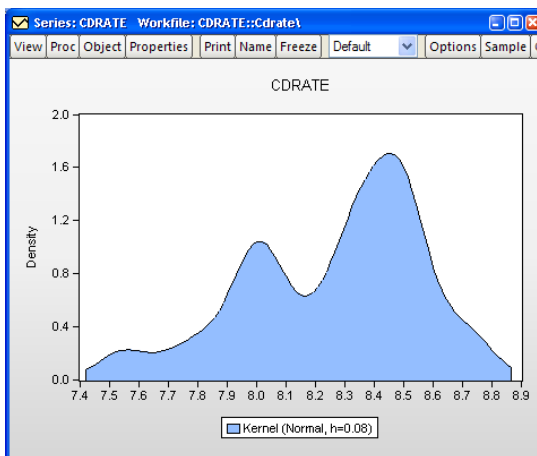
This density estimate has about the right degree of smoothing. Interestingly enough, this density has a trimodal shape with modes at the "focal" numbers 7.5, 8.0, and 8.5. Note that the shading highlights the fact that the kernel estimates are computed only from around 7.45 to around 8.85.

Theoretical Distribution

You may plot the density function of a theoretical distribution by selecting **View/Graph...** from the series or group menu, and choosing **Distribution** in the **Specific** listbox. The right-hand side of the dialog page will change to provide a **Distribution** dropdown menu prompting you to choose a distribution graph. You should select **Theoretical Distribution**.

By default, EViews will display the normal density function fit to the data.
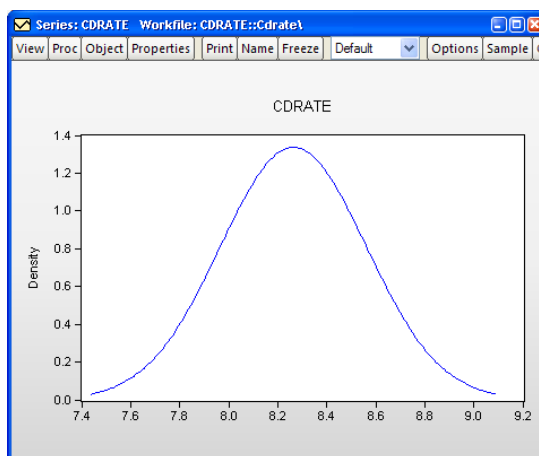
The **Options** button may be used to display the **Distribution Plot Customize** dialog. As with other distribution graphs, the left-hand side of the dialog may be used to add distribution graphs to the current plot (*e.g.*, combining a histogram and a theoretical distribution).

The right-hand side of the dialog allows you to specify the parametric distribution that you wish to display. Simply select the distribution

of interest from the drop-down menu. The small display window will change to show you the parameterization of the specified distribution.

You can specify the values of any known parameters in the edit field or fields. If you leave any field blank, EViews will estimate the corresponding parameter using the data contained in the series.

The **Estimation Options** provides control over iterative estimation, if relevant. You should not need to use these settings unless the graph indicates failure in the estimation process.

Most of the options are self-explanatory. If you select **User-specified starting values**, EViews will take the starting values from the C coefficient vector.

Empirical CDF

The empirical CDF graph displays a plot of the empirical cumulative distribution function (CDF) of the series. The CDF is the probability of observing a value from the series not exceeding a specified value r:

$$F_x(r) \ = \ \Pr(x \leq r)$$

To display the empirical CDF, you should select **View/Graph...** from the series or group menu, choose **Distribution** in the **Specific** listbox, and select **Empirical CDF** in the **Distribution** dropdown.
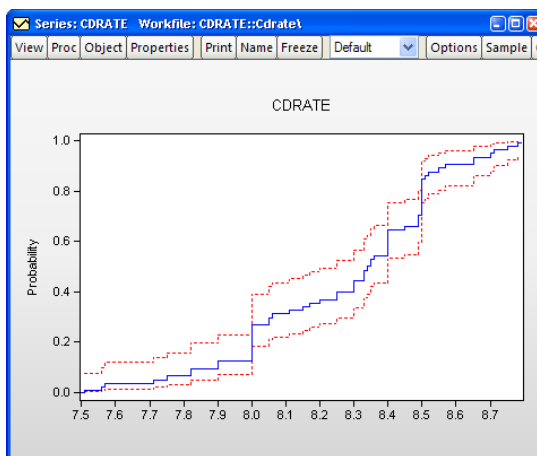
By default, EViews displays the empirical CDF for the data in the series along with approximate 95% confidence intervals. The confidence intervals are based on the Wilson interval methodology (Wilson, 1927; Brown, Cai and Dasgupta, 2001).

Clicking on the **Options** button displays a dialog that allows you to specify the method for computing the CDF, to turn on or off the displaying of confidence intervals, to specify the size of the confidence interval, and to control the display of legend entries.

The **Quantile Method** dropdown controls the method of computing the CDF. Given a total of N observations, the CDF for value r is estimated as:

Rankit (default)	$(r - 1/2)/N$
Ordinary	r/N
Van der Waerden	$r/(N+1)$
Blom	$(r - 3/8)/(N + 1/4)$
Tukey	$(r - 1/3)/(N + 1/3)$
Gumbel	$(r - 1)/(N - 1)$

See Cleveland (1994) and Hyndman and Fan (1996). By default, EViews uses the Rankit method, but you may use the dropdown to select a different method.

Empirical Survivor

The empirical survivor graph of a series displays an estimate of the probability of observing a value at least as large as some specified value r:

$$S_x(r) \; = \; \Pr(x > r) \; = \; 1 - F_x(r)$$

To display the empirical survivor function, select **View/Graph...** from the series or group menu, choose **Distribution** in the **Specific** listbox, and select **Empirical Survivor** in the **Distribution** dropdown.

By default, EViews displays the estimated survivor function along with a 95% confidence interval (Wilson, 1927; Brown, Cai and Dasgupta, 2001).

See "Empirical CDF" on page 696 for additional discussion and a description of the dialog brought up by the **Options** button.

Empirical Log Survivor

The empirical log survivor graph for a series displays the log of the probability of observing a value at least as large as some specified value r.

To display the empirical log survivor function, select **View/Graph...** from the series or group menu, choose **Distribution** in the **Specific** listbox, and select **Empirical Survivor** in the **Distribution** dropdown.

By default, EViews displays the logarithm of the estimated survivor function along with a 95% confidence interval (Wilson, 1927; Brown, Cai and Dasgupta, 2001).
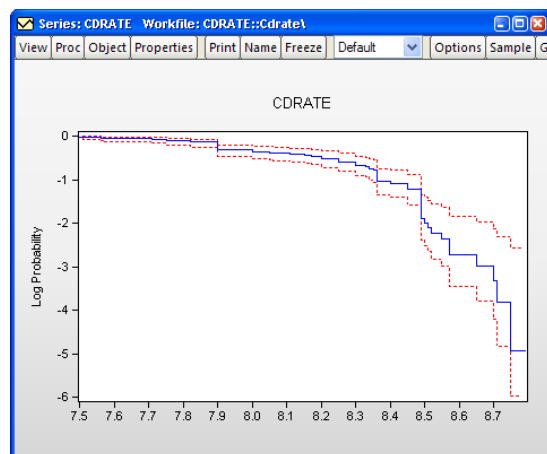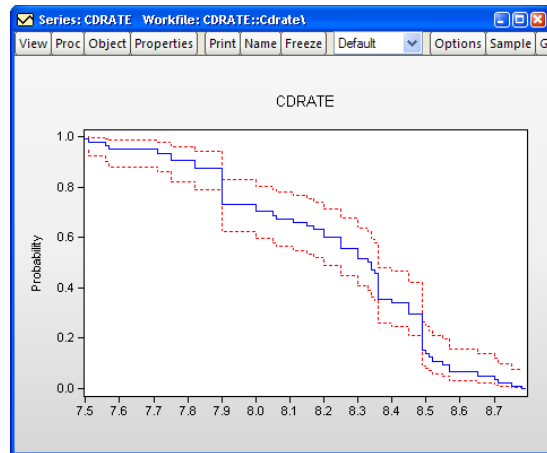
See "Empirical Survivor" on page 697 for additional discussion and a description of the dialog brought up by the **Options** button.

Empirical Quantile

This graph type plots the empirical quantiles of the series against the associated probabilities. The quantile is the inverse function of the CDF; graphically, the quantile can be obtained by flipping the horizontal and vertical axis of the CDF.

For $0 < q < 1$, the q-th quantile $x_{(q)}$ of x is a number such that:

$$Pr(x \leq x_{(q)}) \geq q$$
$$Pr(x \geq x_{(q)}) \leq 1 - q$$

The graph plots the values of $x_{(q)}$ against q.
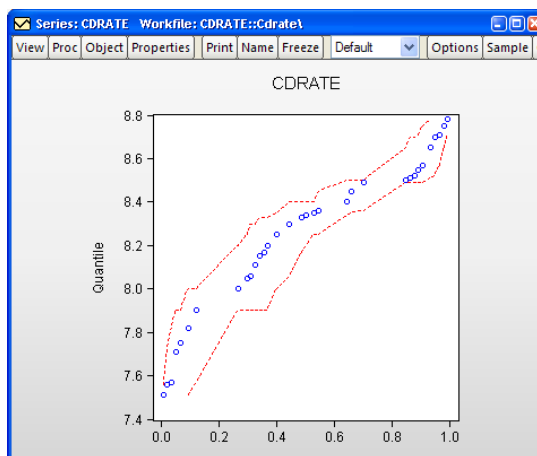
To display the empirical quantile plot, select **View/Graph...** from the series or group menu, choose **Distribution** in the **Specific** listbox, and **Empirical Quantile** in the **Distribution** dropdown.

By default, EViews displays the empirical quantiles along with approximate 95% confidence intervals obtained by inverting the Wilson confidence intervals for the CDF (Wilson, 1927; Brown, Cai and Dasgupta, 2001).

See "Empirical Survivor" on page 697 for a description of the dialog brought up by clicking the **Options** button.

Quantile-Quantile (Theoretical)

Theoretical quantile-quantile plots are used to assess whether the data in a single series follow a specified theoretical distribution; *e.g.* whether the data are normally distributed (Cleveland, 1994; Chambers, *et al.* 1983). If the two distributions are the same, the QQ-plot should lie on a straight line. If the QQ-plot does not lie on a straight line, the two distributions differ along some dimension. The pattern of deviation from linearity provides an indication of the nature of the mismatch.
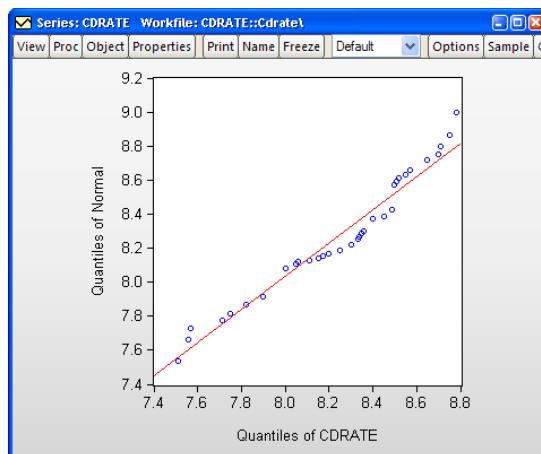
To display the theoretical quantile-quantile plot, select **View/Graph...** from the series or group menu, choose **Quantile - Quantile** in the **Specific** listbox, and select **Theoretical** in the **Q-Q graph** dropdown to the right.

By default, EViews displays the QQ-plot comparing the quantiles of the data with the quantiles of a fitted normal distribution.

The **Options** button may be used to display the **Quantile-Quantile Plot Customize** dialog. The left-side of this graph may be used to add additional QQ-plots to the current plot, allowing you to compare your data to more than one theoretical distribution.

The right-hand side of the dialog allows you to specify the parametric distribution that you wish to display. See "Theoretical Distribution" on page 695 for a discussion of these settings.

In addition, the customize page offers you several methods for computing the empirical quantiles. The options are explained in the section "Empirical CDF" on page 696; the choice should not make much difference unless the sample is very small.

Lastly, the **Display fit line** checkbox provides you with the option of plotting a regression line through the quantile values.

Quantile-Quantile (Symmetry)

An alternative form of the quantile-quantile (QQ)-plot examines the symmetry of the data directly by comparing each quantile value with the corresponding upper-tail quantile value. You may think of this procedure as plotting the distance to points above the median against the distance to the corresponding point below the median (Cleveland, et al. 1983, p. 17). The resulting QQ-symmetry plot has the property that the more symmetric are the data, the closer are the points to the 45 degree line.
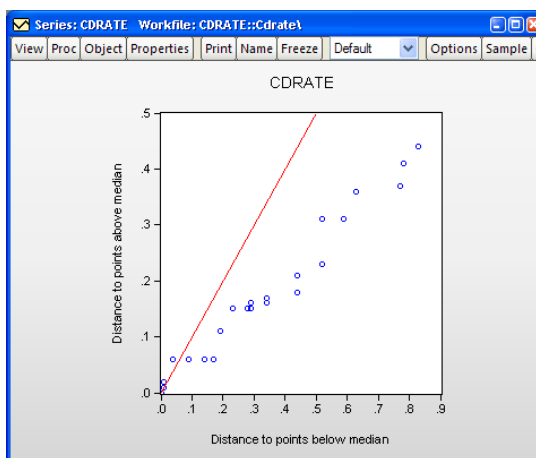
To display the quantile-quantile symmetry plot, select **View/Graph...** from the series or group menu, choose **Quantile - Quantile** in the **Specific** listbox, and select **Symmetry** in the **Q-Q graph** dropdown to the right.

Our example uses the workfile "CDrate.WF1", for which we have plotted the symmetry of the series CDRATE. The default QQ-symmetry plot shows that the data is highly asymmetric.

The **Options** button may be used to display the **Quantile-Quantile Plot Customize** dialog. The settings offered in this dialog are limited; you may choose whether to display the 45 degree line and may modify the legend settings.
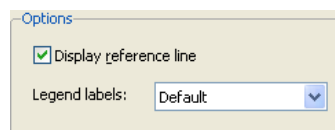
Quantile-Quantile (Empirical)

The empirical quantile-quantile (QQ)-plot plots the quantiles of one series against the quantiles of a second series (Cleveland, 1994; Chambers, *et al.* 1983). If the distributions of the two series are the same, the QQ-plot should lie on a straight line.

To display the empirical quantile-quantile plot for a group with two or more series, select **View/Graph...** from the group menu, choose **Quantile - Quantile** in the **Specific** list-box, and select **Empirical** in the **Q-Q graph** dropdown.
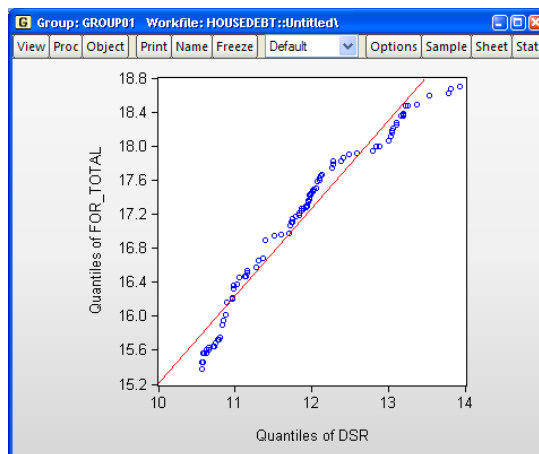
Our illustration uses the example workfile "Housedebt.WF1", containing quarterly data on household debt and financial obligations from 1980 to 2006. We show here the default QQ-plot for the debt service ratio series DSR against the financial obligation ratio series FOR_TO-TAL.

The settings accessed through the **Options** button are limited; you may specify a computation method, choose whether to display the fit line, and modify the legend settings. These settings are discussed in "Theoretical Distribution" on page 695 and "Quantile-Quantile (Theoretical)" on page 699.
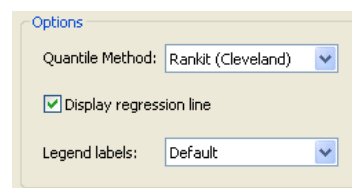
Note that unlike other distribution graphs, EViews does not allow you to add additional QQ-plots for a given pair of series; rarely will the choice of **Quantile Method** generate enough of a difference to make such a plot useful.

Boxplot

A boxplot, also known as a box and whisker diagram, summarizes the distribution of a set of data by displaying the centering and spread of the data using a few primary elements (McGill, Tukey, and Larsen, 1978).

The box portion of a boxplot represents the first and third quartiles (middle 50 percent of the data). These two quartiles are collectively termed the *hinges*, and the difference between them represents the *interquartile range*, or IQR. The median is depicted using a line through the center of the box, while the mean is drawn using a symbol. Note that EViews computes the median for boxplots using the Rankit method.

The *inner fences* are defined as the first quartile minus 1.5*IQR and the third quartile plus 1.5*IQR. The inner fences are typically not drawn in boxplots, but graphic elements known as *whiskers* and *staples* show the values that are outside the first and third quartiles, but within the inner fences. The staple is a line drawn at the last data point within (or equal to) each of the inner fences. Whiskers are lines drawn from each hinge to the corresponding staple.
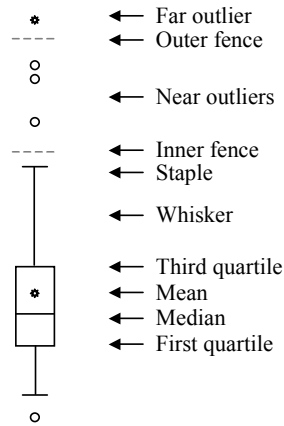
Data points outside the inner fence are known as *outliers*. To further characterize outliers, we define the *outer fences* as the first quartile minus 3.0*IQR and the third quartile plus 3.0*IQR. As with inner fences, outer fences are not typically drawn in boxplots. Data between the inner and outer fences are termed *near outliers*, and those outside the outer fence are referred to as *far outliers*. A data point lying on an outer fence is considered a near outlier.

A shaded region or notch may be added to the boxplot to display approximate confidence intervals for the median (under certain restrictive statistical assumptions). The bounds of the shaded or notched area are defined by the median +/- 1.57*IQR/$\sqrt{N}$, where

Shaded boxplot   Notched boxplot

N is the number of observations. Notching or shading is useful in comparing differences in medians; if the notches of two boxes do not overlap, then the medians are, roughly, significantly different at a 95 % confidence level. It is worth noting that in some cases, most likely involving small numbers of observations, the notches may be bigger than the boxes.

Boxplots are often drawn so that the widths of the boxes are uniform. Alternatively, the box widths can be varied as a measure of the sample size for each box, with widths drawn proportional to N, or proportional to the square root of N.
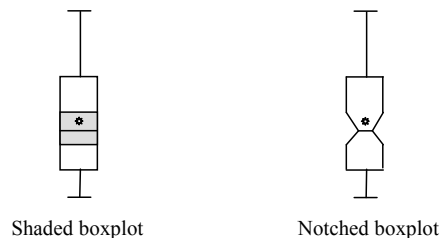
To display a boxplot for a single series or for each series in a group, select **View/Graph...** from the series or group menu, and then choose **Boxplot** in the **Specific** listbox.

(Note that specialized tools allow you to place box-plots along the axes of various graph types.)

The default boxplot for the three month CD rate data is presented here. Note that since we are displaying the boxplot for a single series, EViews changes the aspect ratio of the graph so that it is taller than it is wide. Typically, boxplots are displayed for multiple series; the aspect ratio will adjust accordingly.
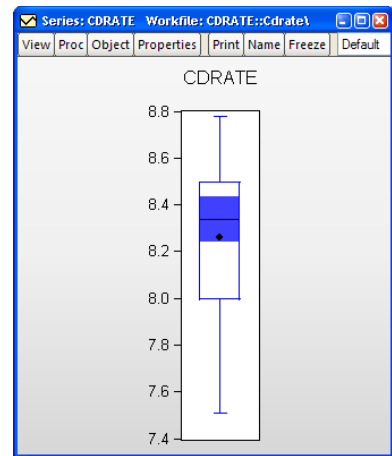
In addition to the **Orientation** option on the main page which allows you to rotate your box-plots, you may specify a number of display options in the **BoxPlots** dialog under the **Graph Elements** group.

The left-hand side of the **BoxPlot** dialog allows you to show or hide specific elements of the boxplot, to control the box widths, and to modify the appearance of the notching and shading.
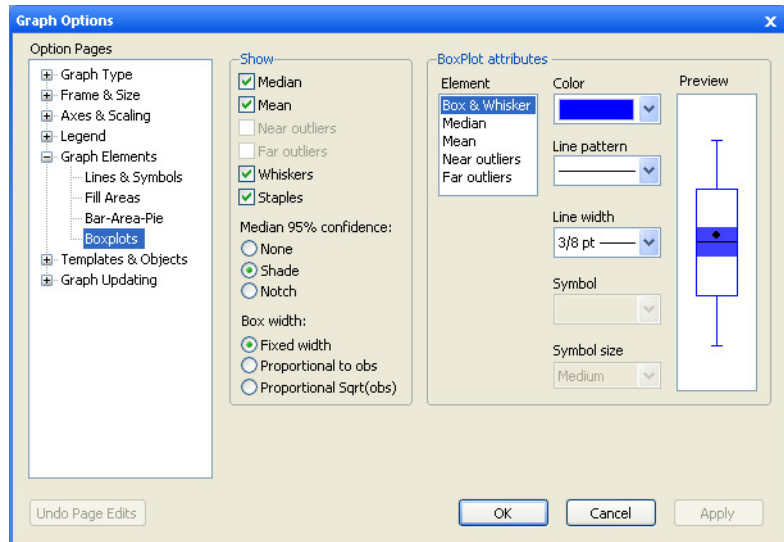
In the right-hand portion of the dialog, you may customize individual elements of your graph. Simply select an element to customize in the **Element** listbox or click on the depiction of a boxplot element in the **Preview** window, and then modify the **Color**, **Line pattern**, **Line/Symbol width**, and **Symbol** type as desired. Note that each boxplot element is represented by either a line or a symbol; the dialog will show the appropriate choice for the element that you have selected.

The **Preview** window will change to display the current settings for your graph. You may also click on elements within the **Preview** window to select them in the **Element** listbox, if you find this easier. To revert to the original graph settings, click on **Undo Page Edits**.

Auxiliary Graph Types

EViews can construct several analytical graphs that are only meant to be added to observation graphs; we term these graphs *auxiliary graphs.* Strictly speaking, auxiliary XY graphs should not be thought of as a distinct graph type, but rather as a class of modifications that may be applied to an observation plot.

At present, auxiliary graphs may be added on top of scatterplots and XY line graphs. When either **Scatter** or **XY Line** is selected in the **Specific** listbox, the right-hand side of the graph dialog changes to offer the **Fit lines** dropdown menu, where you can select one of the auxiliary types to be added to the graph. If you wish to add additional auxiliary graphs or if you wish to customize the settings of your specified type, you should click on the **Options** button to display additional settings.
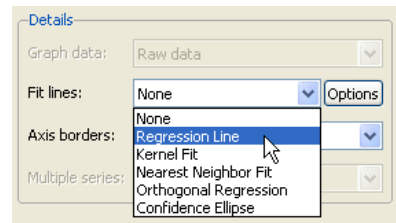
The following is a brief summary of the characteristics of each of these graph types. For illustration purposes, the examples generally use the familiar "Old Faithful Geyser" eruption time data considered by Simonoff (1996) and many others ("Geyser.WF1"). These data provide information on 222 eruption time intervals and previous eruption durations for the Old Faithful Geyser in Yellowstone National Park.

Regression Line

This graph uses data from two series, displaying the fit of a bivariate regression of the second series y on the first series x, and a constant. If desired, you may automatically perform various transformations of your data prior to performing the regression.

Our example uses the geyser data and considers the relationship between previous eruption length, and the interval to the next eruption. We create a group GROUP01 where
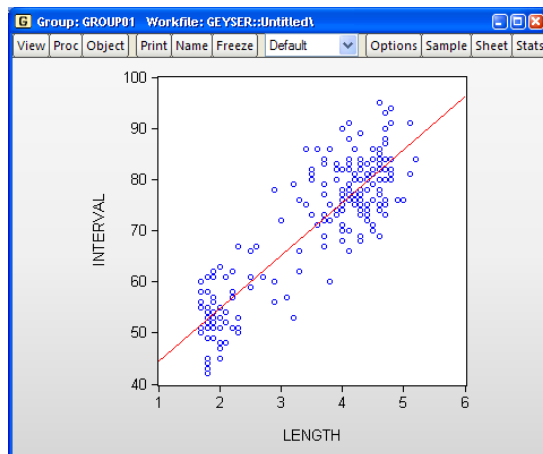
the first series, LENGTH, represents the duration of the previous eruption, and the second series, INTERVAL, measures the interval between eruptions.

In our illustration, the regression line is drawn on top of the scatterplot of points for the geyser data. Clearly there is a positive relationship between length of eruption and the time until the next eruption.

Clicking on the **Options** button displays the **Scatterplot Customize** dialog. The left-hand side of the dialog may be used to add additional auxiliary graphs; simply click on the **Add** button and select the type of element you wish to add.

The right-hand side of the dialog contains options specific to the selected element. In this case, we see the options for the regression line selection.
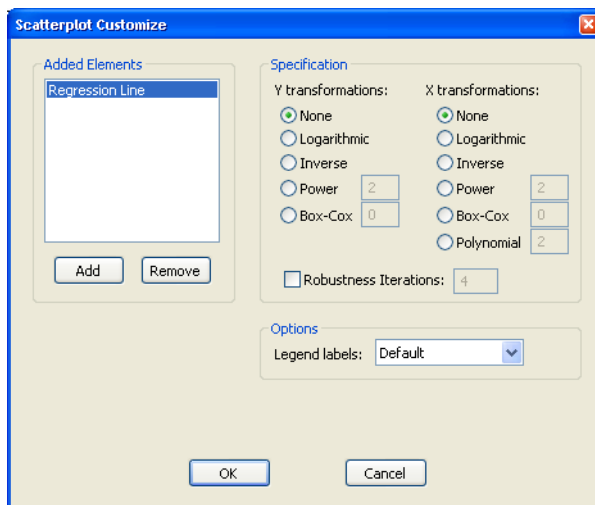
First, you may specify transformations of your dependent and independent variables using the radio buttons. The following transformations are available for the bivariate fit:

None	y	x
Logarithmic	$\log(y)$	$\log(x)$
Inverse	$1/y$	$1/x$
Power	y^a	x^b
Box-Cox	$(y^a - 1)/a$	$(x^b - 1)/b$
Polynomial	—	$1, x, x^2, ..., x^b$

where you specify the parameters a and b in the edit field. Note that the Box-Cox transformation with parameter zero is the same as the log transformation.

- If any of the transformed values are not available, EViews returns an error message. For example, if you take logs of negative values, non-integer powers of nonpositive values, or inverses of zeros, EViews will stop processing and issue an error message.

- If you specify a high-order polynomial, EViews may be forced to drop some of the high order terms to avoid collinearity.

Next, you may instruct EViews to perform robustness iterations (Cleveland, 1993). The least squares method is very sensitive to the presence of even a few outlying observations. The **Robustness Iterations** option carries out a form of weighted least squares where outlying observations are given relatively less weight in estimating the coefficients of the regression.

For any given transformation of the series, the **Robustness Iteration** option carries out robust fitting with bisquare weights. Robust fitting estimates the parameters a, b to minimize the weighted sum of squared residuals,

$$\sum_{i=1}^{N} r_i (y_i - a - x_i b)^2 \tag{13.4}$$

where y_i and x_i are the transformed series and the bisquare robustness weights r are given by:

$$r = \begin{cases} (1 - e_i^2/(36 m^2))^2 & \text{for } |e_i/6m| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{13.5}$$

where $e_i = y_i - a - x_i b$ is the residual from the previous iteration (the first iteration weights are determined by the OLS residuals), and m is the median of $|e_i|$. Observations with large residuals (outliers) are given small weights when forming the weighted sum of squared residuals.

To choose the number robustness iterations, click on the check box for **Robustness Iterations** and specify an integer for the number of iterations.

Lastly there is an option controlling the amount of information provided in legends. The EViews default displays a minimum of legend information; this default may be overridden using the **Legend labels** dropdown menu. In particular, if you wish to see the coefficients of your fitted line you should select **Full**. (Note that coefficient information is not available for some transformations).

Kernel Fit

Using data from two series, this kernel fit displays the local polynomial kernel regression fit of the second series y on the first series x. Extensive discussion may be found in Simonoff (1996), Härdle (1991), Fan and Gijbels (1996).

Both the nearest neighbor fit ("Nearest Neighbor Fit," on page 709), and the kernel regression fit are nonparametric regression methods that fit local polynomials. The two differ in how they define "local" in the choice of bandwidth.

The effective bandwidth in nearest neighbor regression varies, adapting to the observed distribution of the regressor. For the kernel fit, the bandwidth is fixed but the local observations are weighted according to a kernel function.

Local polynomial kernel regressions fit Y at each value x, by choosing the parameters β to minimize the weighted sum-of-squared residuals:

$$m(x) \;=\; \sum_{i=1}^{N} \left(Y_i - \beta_0 - \beta_1 (x - X_i) + -\ldots -\beta_k (x - X_i)^k \right)^2 K\!\left(\frac{x - X_i}{h}\right) \qquad (13.6)$$

where N is the number of observations, h is the bandwidth (or smoothing parameter), and K is a kernel function that integrates to one. Note that the minimizing estimates of β will differ for each x.

The default settings compute the local linear fit using the Epanechnikov kernel and an arbitrary, rule of thumb bandwidth rule. For efficient purposes, the kernel fit is evaluated using the linear binning method proposed by Fan and Marron (1994).

Our example shows the default kernel fit line drawn on top of the geyser scatterplot data. As with the regression line there is a positive relationship between the length of eruption and the time until the next eruption. There does appear to be some flattening of the slope of the relationship for long durations, suggesting that there may be a different model for short and long duration times.
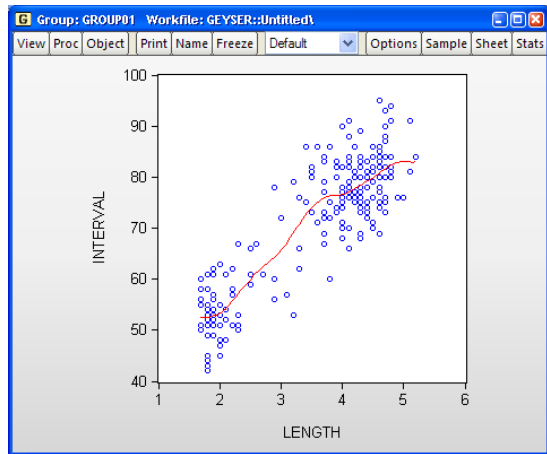
You may click on the **Options** button to display the **Scatterplot Customize** dialog. As always, the left-hand side of the graph may be used to add additional auxiliary graphs, while the right-hand side of the dialog provides options for the kernel fit.

You will need to specify the form of the local regression (**Nadaraya-Watson** constant, **Linear**, **Polynomial**), the kernel function, the bandwidth, and other options to control the fit procedure.
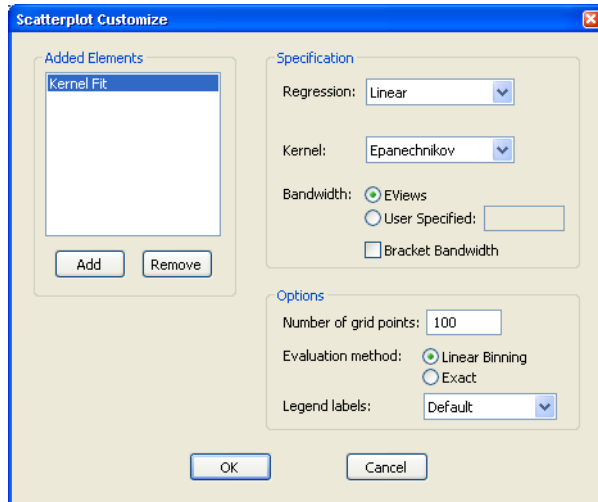
Regression

Here, you will specify the order of the polynomial k to fit at each data point. The **Nadaraya-Watson** option sets $k = 0$ and locally fits a constant at each x. **Local Linear** sets $k = 1$ at each x. For higher order polynomials, mark the **Local Polynomial** option and type in an integer in the field box to specify the order of the polynomial.
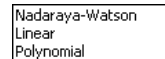
Kernel

The kernel is the function used to weight the observations in each local regression. Definitions are provided in the discussion of "Kernel Density," beginning on page 691.

Bandwidth

The bandwidth h determines the weights to be applied to observations in each local regression. The larger the h, the smoother the fit. By default, **EViews** arbitrarily sets the bandwidth to:

$$h = 0.15(X_U - X_L) \tag{13.7}$$

where $(X_U - X_L)$ is the range of X.

To specify your own bandwidth, mark **User Specified** and enter a nonnegative number for the bandwidth in the edit box.

The **Bracket Bandwidth** option fits three kernel regressions using bandwidths $0.5h$, h, and $1.5h$.

For nearest neighbor (variable) bandwidths, see

Number of grid points

You must specify the number of points M at which to evaluate the local polynomial regression. The default is $M = 100$ points; you can specify any integer in the field. Suppose the range of the series X is $[X_L, X_U]$. Then the polynomial is evaluated at M equi-spaced points:

$$x_i = X_L + i \cdot \left(\frac{X_U - X_L}{M} \right) \qquad \text{for } i = 0, 1, \dots M - 1 \tag{13.8}$$

Method

Given a number of evaluation points, EViews provides you with two additional computational options: exact computation and linear binning.

The **Linear Binning** method (Fan and Marron, 1994) approximates the kernel regression by binning the raw data X_j fractionally to the two nearest evaluation points, prior to evaluating the kernel estimate. For large data sets, the computational savings may be substantial, with virtually no loss of precision.

The **Exact** method performs a regression at each x_i, using all of the data points (X_j, Y_j), for $j = 1, 2, \dots, N$. Since the exact method computes a regression at every grid point, it may be quite time consuming when applied to large samples. In these settings, you may wish to consider the linear binning method.

Nearest Neighbor Fit

The nearest neighbor fit displays local polynomial regressions for two series with bandwidth based on nearest neighbors. Briefly, for each data point in a sample, we fit a locally weighted polynomial regression. It is a local regression since we use only the subset of observations which lie in a neighborhood of the point to fit the regression model; it may be weighted so that observations further from the given data point are given less weight.
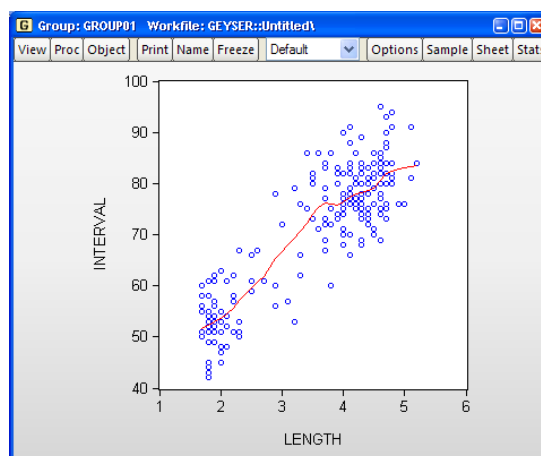
This class of regressions includes the popular *Loess* (also known as *Lowess*) techniques described by Cleveland (1993, 1994). Additional discussion of these techniques may be found in Fan and Gijbels (1996), and in Chambers, Cleveland, Kleiner, Tukey (1983).

The default settings estimate a local linear regression using a bandwidth of 30% of the sample. The estimates use Tricube weighting, and Cleveland subsampling of the data.

Our illustration shows results that are broadly similar to the results for the kernel fit. There is a positive relationship between the length of eruption and the time until the next eruption, with evidence of flattening of the slope of the relationship for long durations.

Clicking on the **Options** button displays the **Scatterplot Customize** dialog. The left-hand side of the dialog may be used to add additional auxiliary graphs, while the right-hand side of the dialog provides options for the nearest neighbor fit.

You will need to specify the form of the local regression, the bandwidth, and other options to control the fit procedure.
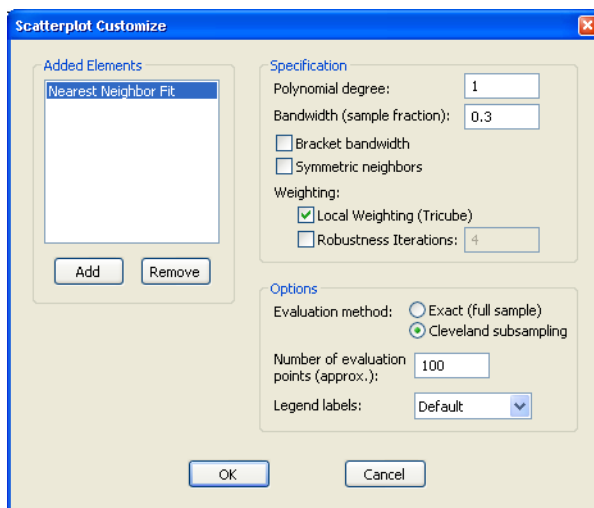
Specification

For each point in the sample selected by the **Evaluation Method** option, we compute the fitted value by running a local regression using data around that point. The **Specification** option determines the rules employed in identifying the observations to be included in each local regression, and the functional form used for the regression.

Polynomial degree specifies the degree of polynomial to fit in each local regression.

Bandwidth span determines which observations should be included in the local regressions. You should specify a number α between 0 and 1. The span controls the smoothness of the local fit; a larger fraction α gives a smoother fit. The fraction α instructs EViews to include the $\lfloor \alpha N \rfloor$ observations nearest to the given point, where $\lfloor \alpha N \rfloor$ is 100α % of the total sample size, truncated to an integer.

If you mark the **Bracket bandwidth span** option, EViews displays three nearest neighbor fits with spans of 0.5α, α, and 1.5α.

Note that this standard definition of nearest neighbors implies that the number of points need not be symmetric around the point being evaluated. If desired, you can force symmetry by selecting the **Symmetric neighbors** option. **Symmetric Neighbors** forces the local regression to include the same number of observations to the left and to the right of the point being evaluated. This approach violates the definition, but arguably not the spirit, of nearest neighbor regression. Differences between the two approaches will show up where the data are thin (there are relatively few observations in the region).

Weighting

Local Weighting (Tricube) weights the observations of each local regression. The weighted regression minimizes the weighted sum of squared residuals:

$$\sum_{i=1}^{N} w_i(y_i - a - x_i b_1 - x_i^2 b_2 - \ldots - x_i^k b_k). \tag{13.9}$$

The tricube weights w are given by:

$$w_i = \begin{cases} \left(1 - \left|\dfrac{d_i}{d(\lfloor \alpha N \rfloor)}\right|^3\right)^3 & \text{for } \left|\dfrac{d_i}{d(\lfloor \alpha N \rfloor)}\right| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{13.10}$$

where $d_i = |x_i - x|$ and $d(\lfloor \alpha N \rfloor)$ is the $\lfloor \alpha N \rfloor$-th smallest such distance. Observations that are relatively far from the point being evaluated get small weights in the sum of squared residuals. If you turn this option off, each local regression will be unweighted with $w_i = 1$ for all i.

Robustness Iterations iterates the local regressions by adjusting the weights to downweight outlier observations. The initial fit is obtained using weights w_i, where w_i is tricube if you choose **Local Weighting** and 1 otherwise. The residuals e_i from the initial fit are used to compute the robustness bisquare weights r_i as given in "Regression Line," beginning on page 704. In the second iteration, the local fit is obtained using weights $w_i r_i$. We repeat this process for the user specified number of iterations, where at each iteration the robustness weights r_i are recomputed using the residuals from the last iteration.

Note that LOESS/LOWESS is a special case of nearest neighbor fit, with a polynomial of degree 1, and local tricube weighting. The default EViews options are set to produce LOWESS fits.

Options

You should choose between computing the local regression at each data point in the sample, or using a subsample of data points.

- **Exact (full sample)** fits a local regression at every data point in the sample.

- **Cleveland subsampling** performs the local regression at only a subset of points. You should provide the size of the subsample M in the edit box.

  The number of points at which the local regressions are computed is approximately equal to M. The actual number of points will depend on the distribution of the explanatory variable.

Since the exact method computes a regression at every data point in the sample, it may be quite time consuming when applied to large samples. For samples with over 100 observations, you may wish to consider subsampling.

The idea behind subsampling is that the local regression computed at two adjacent points should differ by only a small amount. Cleveland subsampling provides an adaptive algorithm for skipping nearby points in such a way that the subsample includes all of the representative values of the regressor.

It is worth emphasizing that at each point in the subsample, EViews uses the entire sample in determining the neighborhood of points. Thus, each regression in the Cleveland subsample corresponds to an equivalent regression in the exact computation. For large data sets, the computational savings are substantial, with very little loss of information.

Orthogonal Regression

The orthogonal regression fit displays the line that minimizes the orthogonal (perpendicular) distances from the y data to the fit line. This graph may be contrasted with the regression fit ("Regression Line," beginning on page 704) which displays the line that minimizes the sum of squared vertical distances from the data to the corresponding fitted y values on the regression line.

Apart from adding other auxiliary graphs, the only option for orthogonal regression is the **Legend labels** dropdown menu. If you wish to see the properties of your fitted line you should select **Full**. EViews will display the mean of X, the mean of Y and the estimated angle parameter.
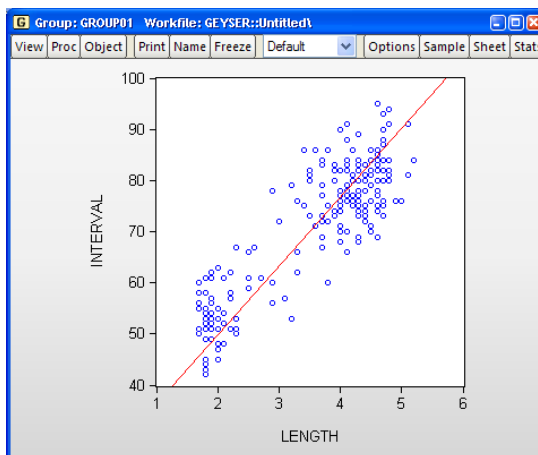
Confidence Ellipse

The confidence ellipse for a pair of series displays the confidence region around the means (Johnson and Wichern 1992, p. 189).

By default, EViews displays the 95% confidence ellipse around the means, computed using the F-distribution with 2 and $N-2$ degrees-of-freedom.

Our illustration shows the default confidence ellipse around the means of the geyser data. The effect of the positive correlation between the length of eruption and time until next eruption is apparent in the oval shape of the region.
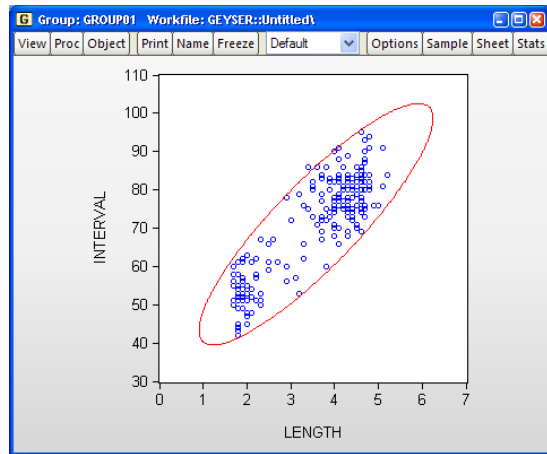
Pressing the **Options** button opens a dialog that allows you to specify additional auxiliary graphs to be added, or to modify the ellipse options.

The edit field at the top of the dialog is where you will enter the probabilities for which you wish to compute confidence regions. If you wish to compute more than one, simply provide a space-delimited list of values or put them in a vector and enter the name of the vector.
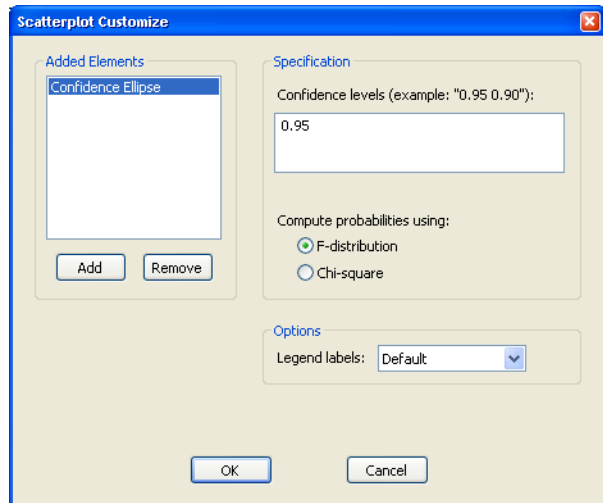
Next, you may change the method of computing the interval to use the $\chi^2(2)$ distribution instead of the F-distribution.

Lastly, you may use the **Legend labels** dropdown menu to change the amount of information provided. If you select **Full**, EViews will always display both the probability associated with each ellipse as well as the distribution used to compute values.

User-Defined Fit Lines

You may add custom fit lines to your scatterplot. Within a scatterplot, select **User-defined** from the **Fit lines** drop-down on the **Basic Type** page, and click the **Options** button to display **Scatterplot Customize** dialog. The left-hand side of the dialog may be used to add additional auxiliary graphs, while the right-hand side of the dialog provides options for the user-defined line.
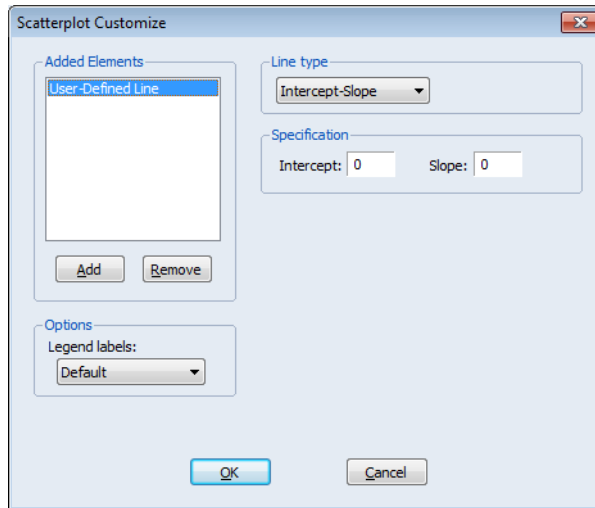
EViews offers three different ways to specify your line, which you can select via the **Line type** dropdown menu. The methods available are **Intercept-Slope** (setting values for the Y-intercept and slope), **Data pairs** (selecting two points of data to draw the line through), or **Custom**, which allows more complicated line specifications.

For example, let's start by creating a simple intercept-slope line, with an intercept of "5" and a slope of "0.005". We also select the **Detailed Legend labels** option, which results in the following graph:

URATE=5+0.005*M1
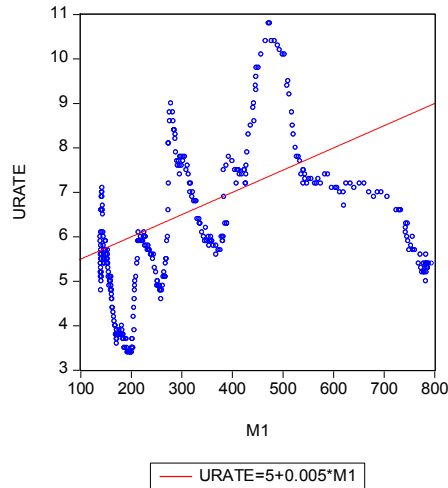
Similarly, we could specify two points through which to draw the line by selecting **Data pairs** as the line type. We use the points (180,6) and (750,9), which results in the following, similar, graph:

URATE=5.053+0.005263*M1
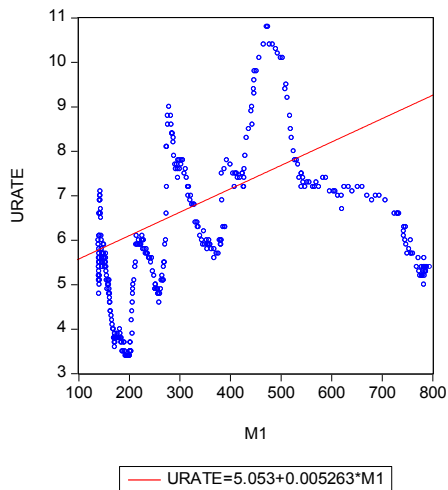
References

Brown, Lawrence D., T. Tony Cai, and Anirban DasGupta (2001). "Interval Estimation for a Binomial Proportion," *Statistical Science*, 16(2), 101-117.

Chambers, John M., William S. Cleveland, Beat Kleiner, and Paul A. Tukey (1983). *Graphical Methods for Data Analysis*, Murray Hill, NJ: Wadsworth & Brooks/Cole Publishing Company.

Cleveland, William S. (1993). *Visualizing Data*, Summit, NJ: Hobart Press.

Cleveland, William S. (1994). *The Elements of Graphing Data*, Summit, NJ: Hobart Press.

Conover, W. J. (1980). *Practical Nonparametric Statistics*, 2nd edition, New York: John Wiley & Sons.

Fan, J. and I. Gijbels (1996). *Local Polynomial Modelling and its Applications*, London: Chapman & Hall.

Fan, J. and J. S. Marron (1994). "Fast Implementations of Nonparametric Curve Estimators," *Journal of Computational and Graphical Statistics*, 3, 35–56.

Freedman, David and Persi Diaconis (1981). "On the Histogram as a Density Estimator: L_2 Theory," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*," 57, 453-476.

Hyndman, R. J. and Fan, Y. (1996). "Sample Quantiles in Statistical Packages," *American Statistician*, 50(4), 361–365.

Härdle, Wolfgang (1991). *Smoothing Techniques with Implementation in S*, New York: Springer Verlag.

Johnson, R. A., and D. W. Wichern (1992). *Applied Multivariate Statistical Analysis, Third Edition*, Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Jones, M. C., M. Samiuddin, A. H. Al-Harbey, and T. A. H. Maatouk (1998). "The Edge Frequency Polygon," *Biometrika*, 85(1), 235-239.

Marron, J. S. and D. Nolan (1989). "Canonical Kernels for Density Estimation," *Statistics and Probability Letters*, 7, 191–195.

McGill, R., J.W. Tukey, and W. Larsen (1978). "Variations of Boxplots," *The American Statistician*, 32(1), 12-16.

Scott, David W. (1979). "On Optimal and Data-Based Histograms," *Biometrika*, 66(3), 605-610.

Scott, David W. (1985a). "Frequency Polygons: Theory and Application," *Journal of the American Statistical Association*, 80(390), 348-354.

Scott, David W. (1985b). "Average Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions," *The Annals of Statistics*, 13(3), 1024-1040.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, London: Chapman & Hall.

Simonoff, Jeffrey S. (1996). *Smoothing Methods in Statistics*, New York: Springer-Verlag.

Simonoff, Jeffrey S. and Frederic Udina (1997). "Measuring the Stability of Histogram Appearance When the Anchor Position is Changed," *Computational Statistics & Data Analysis*, 23, 335-353.

Stock, James H. and Mark W. Watson (2007). *Introduction to Econometrics*, Boston: Pearson Education, Inc.

Wilson, Edwin B. (1972). "Probably Inference, the Law of Succession, and Statistical Inference," *Journal of the American Statistical Association*, 22(158), 209-212.

Chapter 14. Categorical Graphs

Suppose that you have a sample of individuals in the United States, with information on employment, earnings, and various demographic variables. Among other things, you may wish to:

- Display a bar plot comparing the mean incomes of individuals living in each state.

- Produce a scatterplot of wages and hours worked, where the subset of males is drawn using one plotting symbol, and the subset of females uses a different symbol.

- Show wage–education profiles for both male and female workers.

- Draw histograms and boxplots of wages for union and non-union workers in different industries.

These graphs are all examples of *categorical graphs.* Categorical graphs are observation or analytical graphs formed using subsets of the data, where the subsets are defined using the values of one or more categorical conditioning variables (which we refer to as *factors)*. In the examples above, state of residence, gender, years of education, and union status are factors that are used to form subsets of the data, which we then use to construct the graph.

Constructing these graphs by hand can be a difficult and time consuming-process. Fortunately, EViews provides powerful tools for constructing categorical graphs directly from your data. With these tools, you may quickly and easily define your categorization, specify the graph you wish to construct, and describe the basic graph layout; additional options provide detailed control over layout and labeling of the graph, if necessary.

The remainder of this chapter describes the construction of categorical graphs of data from a series or group object using the **View/Graph...** menu item.

Illustrative Examples

Starting from the premise that the most useful method of documenting categorical graphs is to work through examples, we begin by describing the construction of a few representative cases. We divide these examples into two broad categories: graphs which display categorical summaries of the data, and graphs which display the raw data with category identifying information.

Since there is considerable detail in many of the example graphs, we have saved the graphs and then imported them directly into the manual, rather than using the usual screen capture approach.

Category Summaries

Perhaps the most common form of categorical graph involves the display of summary information computed for subsets of observations. For this type of categorical graph, we plot summaries based on the classification, not the original data.

We consider three examples of summary graphs: the first example involves simple plots of descriptive statistics computed for each group; the second example produces line plots from categorical descriptive statistics; the third example constructs analytical graphs for each factor level (category).

Descriptive Statistics

The simplest categorical graph compares values for descriptive statistics for observations in each category.

For our first set of examples, we employ the workfile "Gulfcoast.WF1" which contains demographic information for counties located in the Gulf Coast region of the United States. The workfile consists of 234 observations; 117 counties measured at two different periods (July 2005 and January 2006). The latter measurement is from a special assessment taken by the Census Bureau to measure the impact of hurricanes Rita and Katrina on population in the region.

The series POP contains data on the population in each county (in thousands). The series YEAR identifies the period associated with each observation, while STATE_NAME and COUNTY_NAME are categorical series that identify the observation at the state and county level, respectively.

We begin by constructing a summary graph comparing total population in the two periods. There are three parts to specifying this graph.

First, open the series POP and select **View/ Graph...** to display the graph dialog, which should automatically open to the **Basic type**
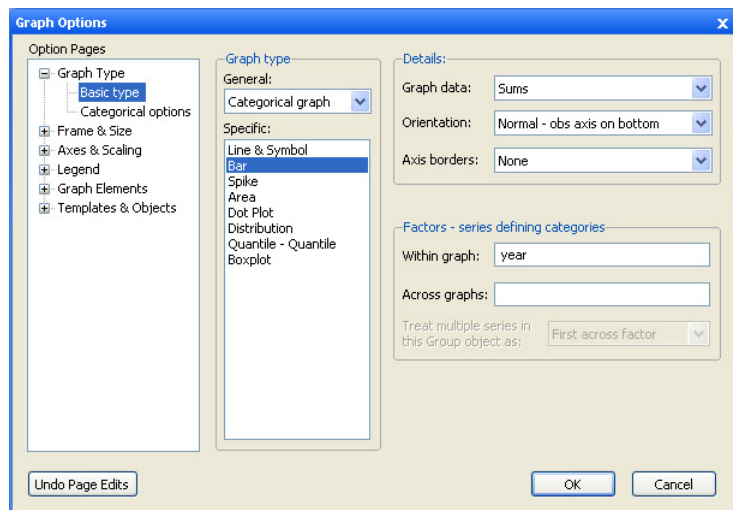
group. Select **Categorical graph** from the **General** dropdown and **Bar** from the **Specific** type in the middle of the dialog to identify the main graph type.

Next, select **Sums** in the **Graph data** dropdown on the right-hand side of the dialog. This setting instructs EViews to plot the sum of POP computed for each subset of the data.

Lastly, we enter "YEAR" in the **Within graph** edit field. EViews will construct categories using the two unique values in YEAR (YEAR = 2005 and YEAR = 2006), and will display the summary statistics *within* a single graph frame.

Sum of POP by YEAR

Click on **OK** to accept the settings. EViews will display a bar graph showing the total population for each year, computed by taking sums of POP over all 117 counties in the region for the given year. We see that total population in the first year was roughly 12 million, and that the total population in the region falls by roughly 250,000 over the periods.
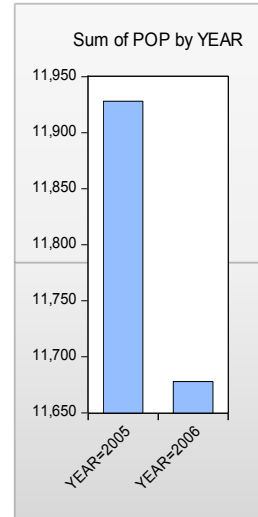
To gain additional insight into the composition of the population change, we may construct a categorical graph showing the sums of POP categorized using both YEAR and STATE_NAME. Double click on the graph window to display the dialog, edit the **Within graph** edit field to read "YEAR STATE_NAME", and click on **OK** to display the updated graph.

EViews computes the total population for each distinct combination of YEAR and STATE_NAME, and displays bar graphs of the results in a single graph frame. Note that the set of bars for YEAR = 2005 are displayed first, followed by the bars for YEAR = 2006. Also note that the bars for a given STATE_NAME are assigned the same color (*i.e.*, the bars for "Alabama" in the two years are both blue, the bars for "Louisiana" are both red, *etc.*) to facilitate comparison across years.

Sum of POP by YEAR, STATE_NAME
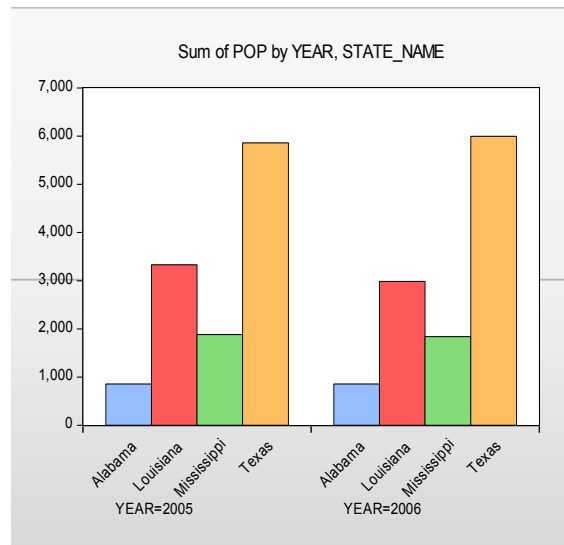
The ordering of the bars in the graph follows the order in which the categorical variables were entered; the factors entered first vary more slowly than later factors (for an apparent excep-

tion to the rule, see "Line Graphs" on page 722). Since YEAR is the first factor in the list, it varies more slowly, *i.e.*, the values for STATE_NAME are grouped within a given year.

While this particular ordering of bars has its merits, grouping by STATE_NAME makes more sense here since presumably, we want to compare population values for a given state across the two years. Rearranging the factor specification so that STATE_NAME precedes YEAR in the **Within graph** list, we now display the graph with the bars grouped by state.

It is considerably easier to visually assess the change in state populations. Not surprisingly, we see that the bulk of the population decrease occurs in Louisiana, and to a lesser extent Mississippi. Texas experi-

ences population growth over the period, in part due to relocations from neighboring states.
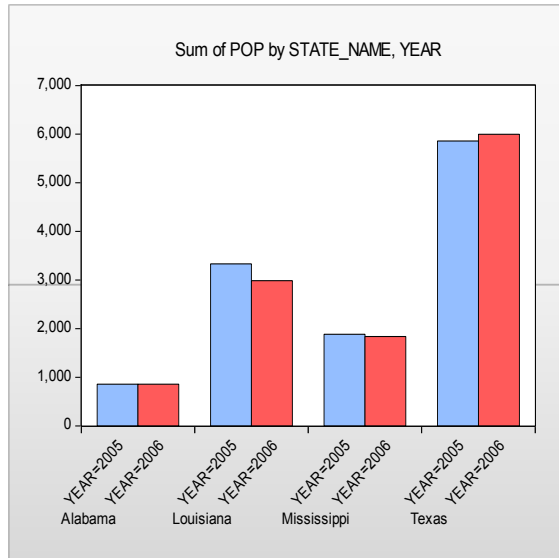
Up to this point we have displayed our categorical graphs within a single graph frame. To display graphs for each category in a separate frame, you should enter the factor name in the **Across graphs** edit field. For example, to display a graph comparing state population across years with each state in its own frame, we enter YEAR in the **Within graphs** and STATE_NAME in the **Across graphs** edit fields. Click on **OK** to display the graph. (We have rearranged the graph so that all four frames appear on a single line by right-clicking on the graph and selecting **Position and align graphs...**; see "Working with Multiple Graphs" on page 784.)
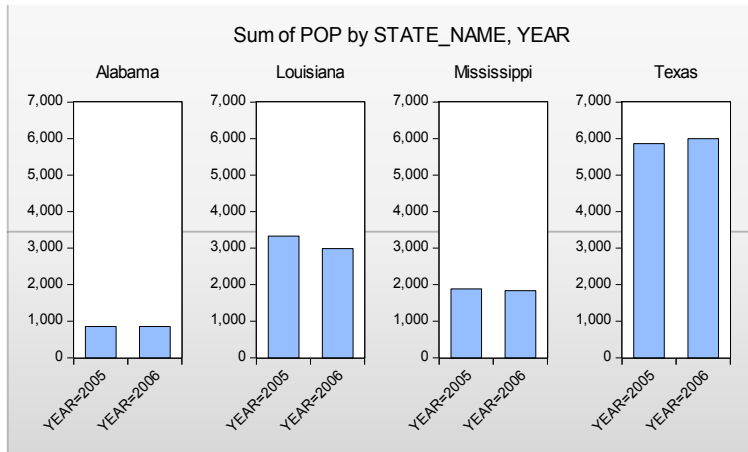
Note that by default, the multiple graph frames employ common vertical axes scales. This feature facilitates comparison of the series sums across states in different frames.

To turn off this feature, select **View/ Graph...** or double-click on the graph to open the **Graph Options** dialog, then select **Categorical options** under the **Graph Type** group on the left-hand side of the dialog.

The **Categorical options** dialog permits control of settings for category definitions and labels, as well as axis scaling. We will have much more to say about the category and label settings later ("Factor Display Settings," on page 737). For now, we focus on the **Axis scales** section.
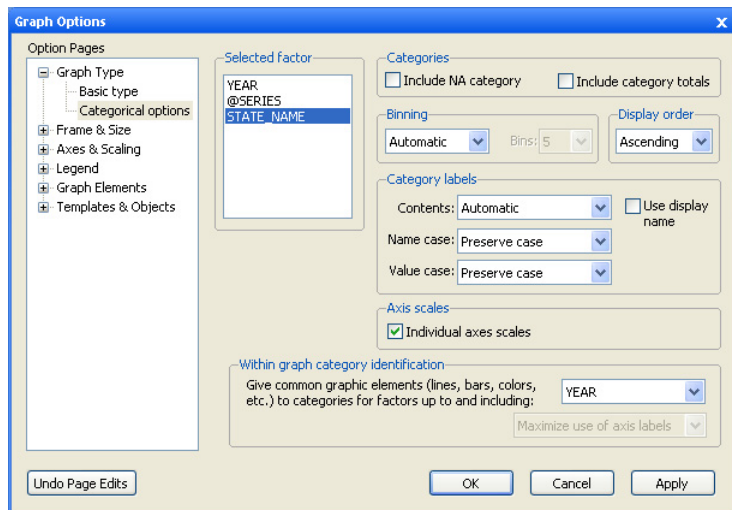
You will use the **Selected factor** list box to select the factor whose properties you wish to modify. In this case, we want each state to have its own scale, so we click on **STATE_NAME**, and select **Individual axes scales**. Click on **OK** to accept the changes and display the modified graph.

Sum of POP by STATE_NAME, YEAR
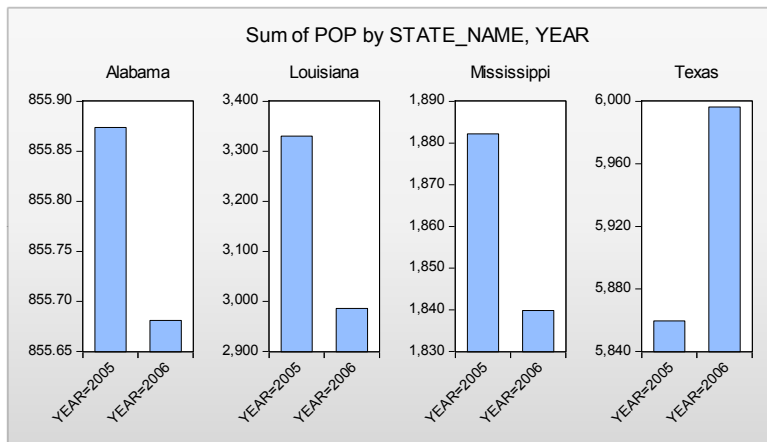
Each graph frame now has its own axis scale, making it easier to see the year-to-year changes, but more difficult to compare the changes across states. While the common scaling made it difficult to determine whether Alabama experienced an increase or decrease in population, the individually scaled graphs clearly show a small reduction in population in that state over the two years.

Line Graphs

One special case of categorical summary plots involves examining line graphs constructed from the summary statistics. While there is a general correspondence to the graphs described in "Category Summaries" on page 718, there are some important differences in the specification of these graphs which require some discussion.

We illustrate these techniques using data from the Panel Study of Income Dynamics (Cornwell and Rupert 1988), as discussed by Baltagi (2001). The data (provided in "Wages.WF1") consist of wage and demographic information for 595 individuals taken over 7 years from 1976–1982. For our purposes, we focus on three binary factors: FEM, a (0, 1) indicator for whether the individual is male (FEM = 0) or female (FEM = 1), UNION, a (0, 1) indicator for whether the wage is set by union contract, and EXPER, a measure of the number of years of full-time work experience.

Suppose, for example that we wish to examine the earnings-experience profiles for all of the individuals in our sample. Our approach will be to compute the average earnings at each experience level and then to display a line graph connecting the mean values. Note that a key feature of EXPER is that is numeric (cardinal), so that it does make sense to draw a line between summary values computed at different experience levels.

First, open the log-wage series LWAGE and select **View/Graph...** to display the graph dialog. Then, select **Categorical graph** from the **General** dropdown in the middle of the dialog and **Line & Symbol** from the **Specific** list to identify the main graph type.

Next, select **Means** in the **Graph data** dropdown on the right-hand side of the dialog to compute the means of LWAGE for each of our categories.

Lastly enter "EXPER" in the **Within graph** edit field and click on **OK** to accept the settings. EViews will display the average earnings-experi-ence profile computed across all of the observations in the workfile as depicted. The profile is generated by computing the mean of LWAGE for each level of the factor variable EXPER and plotting the category means against the category values using a line graph. Note that there is a dropoff in the profile at around 30 years of experience.
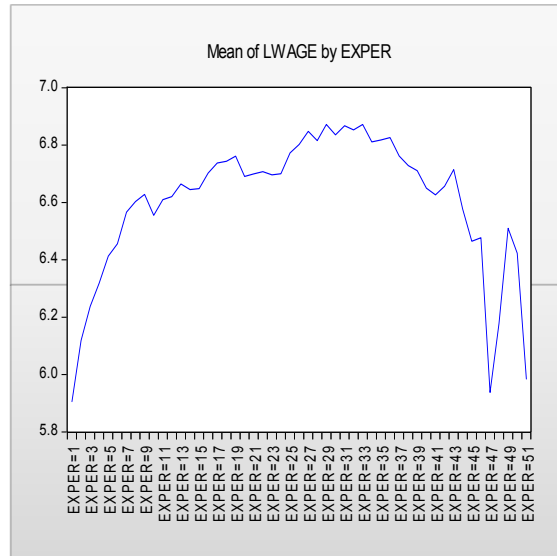
You may notice that the horizontal axis labels showing the category identifiers in this graph are not very attractive (*e.g.,* "EXPER = 20"). We will use the **Categorical options** page to modify these labels.

Double-click on the graph to open the **Graph Options** dialog, then select **Categorical options** under the **Graph Type** group on the left-hand side.
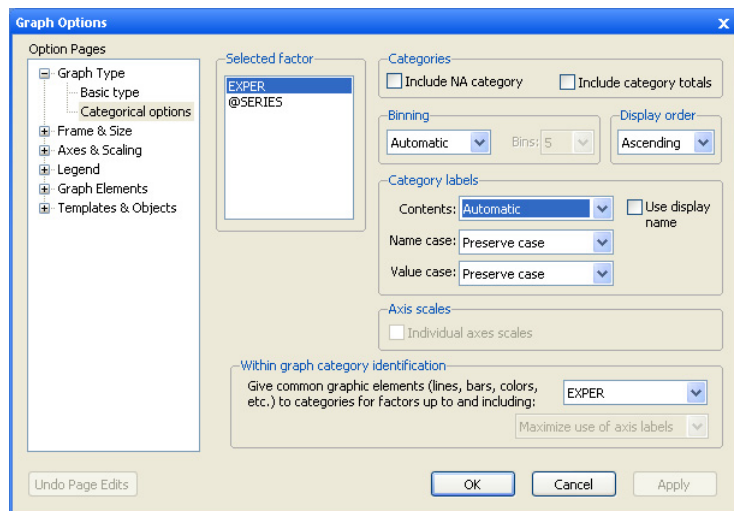
Since we want to change the labeling of the categories defined by levels of experience, we select **EXPER** in the **Selected factor** listbox. The settings of interest are in the section labeled **Category labels**.

The **Contents** dropdown menu provides three settings for the category labels. The default setting, **Automatic**, lets EViews choose the labels. In this example, **Automatic** is equivalent to the second setting, **Factor name & value**, where we form labels using both the name of the factor ("EXPER") and the value of the factor ("20"). In this case, we want to display only the factor value so you should select **Factor value only**. Click on **OK** to accept the updated graph settings.
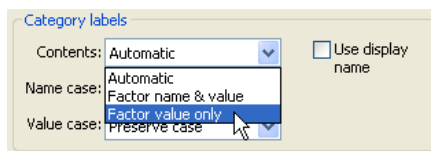
The factor labels in the graph are shorter and slightly easier to read now that they omit the factor name and use only the factor value.

If desired, you may make two additional customizations of the display by double-clicking on the horizontal axis to bring up the **Obs/Date axis** dialog page, and setting the **Observations to label** to **Custom**, starting at 0, with steps of 5. This setting automatically rotates the labels to horizontal. Then go to the **Grid Lines** group, also under the **Axes & Scaling** section, to turn on vertical gridlines. Select **Automatic grid placement** from the **Obs & Date axis grid lines** group, and click **OK**.
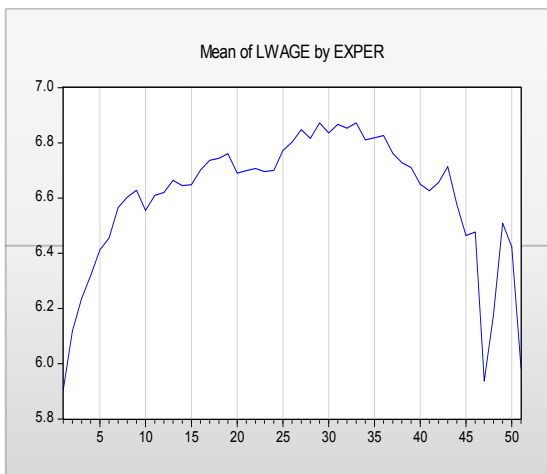
Next, suppose that we wish to compute separate profiles for males and females and to place them in different graph frames. Simply double-click on the graph to display the **Graph Options** dialog, and enter "FEM" in the **Across graphs** edit field. Click on **OK**, and EViews will display the two wage-experience profiles in separate graph frames. (Note that we have again rearranged the graphs so that they are next to each other by right-clicking on the graph and selecting **Position and align graphs...**. We will rearrange many of the following graph examples in this manner.)

Mean of LWAGE by FEM, EXPER
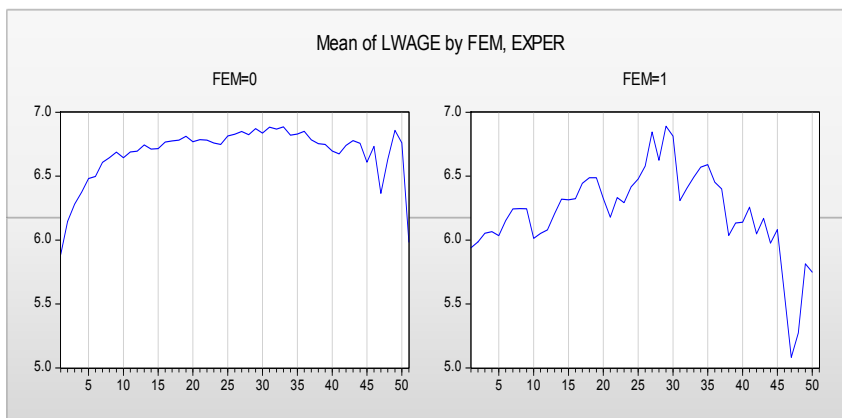
The shapes of the two graphs suggests that the bulk of the dropoff in the overall profile comes from the steep decline in the profile for women at 30 years of experience. (Note that the factor label and interval settings were retained when we added the FEM factor.)

Suppose instead that we wish to display the separate profiles in a single frame. Double click on the graph to bring up the dialog and move "FEM" from the **Across graphs** to the **Within graphs** edit field so that the latter reads "EXPER FEM". The resulting graph shows the wage-experience profile for both males and females in the same graph frame.

Mean of LWAGE by EXPER, FEM
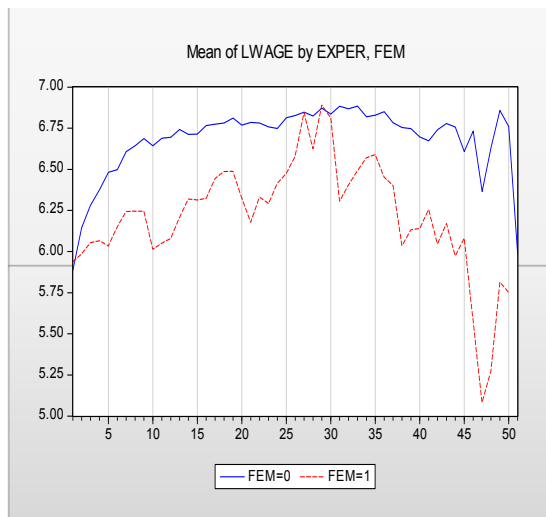
The order in which we enter the factors in this latter example requires some discussion. The rule-of-thumb is that factors should be entered from slowest varying to fastest varying, so that the values for the second factor are grouped within the first factor, and values for the third factor are grouped within the second factor, and so forth.

In this case, however, the first series, EXPER, *appears* to vary more rapidly than the second series, FEM (the variation in EXPER for a given level of FEM using a line), despite preceding it in the list of within series. The apparent reversal of ordering arises from the combined effect of two simple rules: (1) the slowest varying factor is placed along the observation axis,

and (2) line graphs connect data along the observation axis. Since we want to draw lines connecting levels of EXPER along the observation axis, it is entered first in the list.

We describe various implications of the rules for specifying factors in greater depth in "Specifying Factors," on page 734. For now, it is probably sufficient to note that the specified ordering is probably the most natural way of describing the problem since we would probably refer to this graph as displaying the "wage-experience profile, conditional on gender."

Analytical Graphs

You may display categorical graphs where, in place of computing simple descriptive statistics such as the mean or sum, we construct an analytic graph (Distribution, Quantile-Quantile, Boxplots) for each subset of observations.

We begin our example with a simple categorical histogram of the log-wage series from the PSID data described above ("Line Graphs" on page 722). Consider first a simple example showing a histogram of LWAGE with FEM as an across factor. The procedure is straightforward: select **Categorical graph** and **Distribution** for the **Graph type** from the middle of the dialog, select **Histogram** for the **Distribution** type, and place FEM in the across list. Click on **OK** and EViews will display the two histograms in individual graph frames.
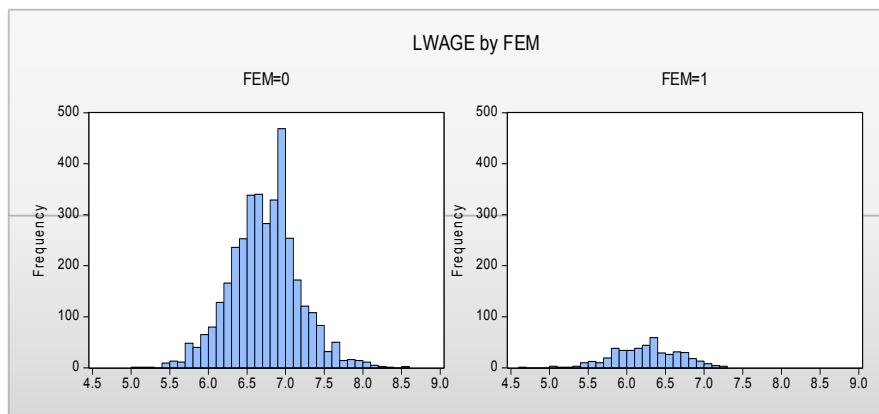
If desired, you may change the multiple graph axis scaling to allow for individual scales as described in "Descriptive Statistics," beginning on page 718.

Next, we consider slightly more complicated examples involving multiple series and multiple factors. We begin by displaying kernel density plots of two series, LWAGE and EXPER, using FEM as a within factor and UNION as an across factor. First, create a group containing LWAGE and EXPER, then select **View/Graph...** from the group menu to display the graph dialog. From this point, constructing our example graph is a four step process:

- First, select **Categorical graph** and **Distribution** as the **Graph type**.

- Select **Kernel Density** from the **Distribution** drop-down.

- Enter "FEM" in the **Within graph** edit field, and "UNION" in the **Across graph** field.

- Select **First across factor** (the default) in the **Treat multiple series in the Group object** dropdown.

The last setting, which is displayed only when graphing multiple series, may appear to be a bit obscure, but the basic idea is really quite simple.

Each series in the group may be viewed as a subset of the data in the workfile. Accordingly, we may define an implicit "series factor," which we denote @SERIES, that divides the workfile data into subsets corresponding to series. In our example, data in the first series of our group are said to be in the category defined by "@SERIES = LWAGE" while data in the second series are in category "@SERIES = EXPER".
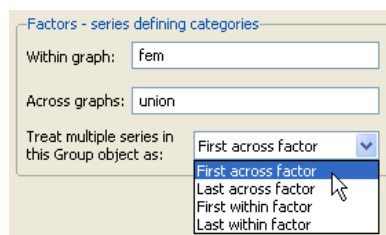
Since @SERIES is a factor, we may choose to have it vary within or across graphs. If it varies within graphs, data for both LWAGE and EXPER will be displayed in a single frame; if it varies across graphs, data for the two series will be displayed in different graph frames. (The choice between plotting the multiple series data in a single graph or in multiple graphs may sound familiar since it corresponds to the **Multiple series** option for basic graphs; see "Multiple Series" on page 632.)

The **Treat multiple series** dropdown menu allows us to insert the implicit @SERIES factor at the beginning or the end of the list of within or across factors. By default, EViews treats @SERIES as the **First across factor** (most slowly varying across factor), but you may move it to the end of the across list or the beginning or end of the within list.
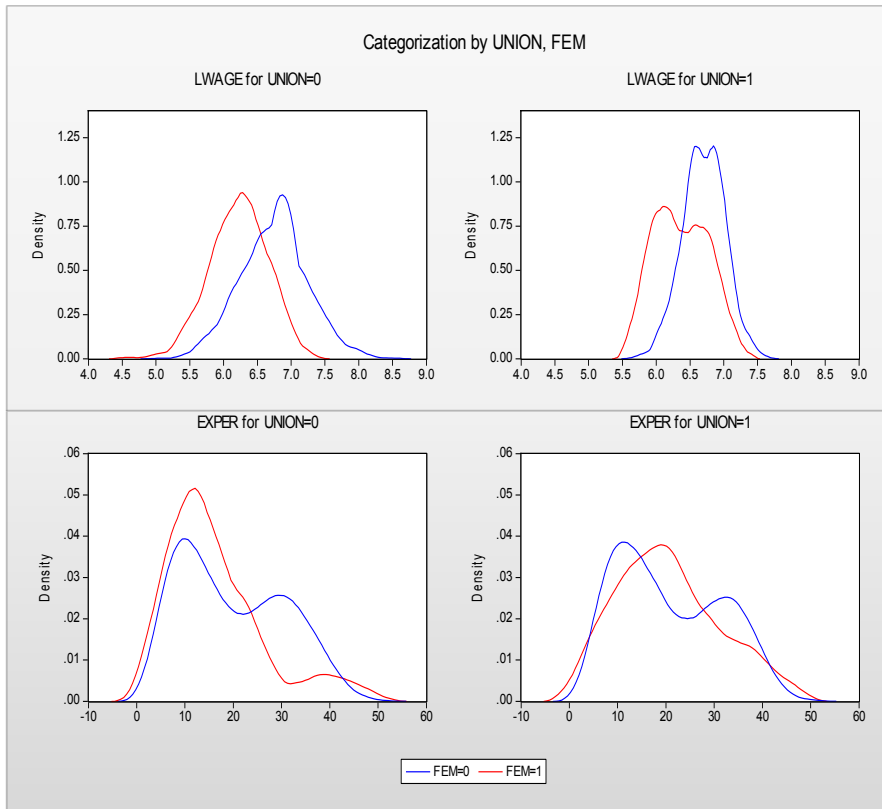
The current example specifies @SERIES as the first, and UNION as the second across factor. Since @SERIES varies more slowly, values of UNION will be grouped within @SERIES. We see the effect of this grouping since the first two frames are for data where "@SERIES = LWAGE" paired with "UNION = 0" and "UNION = 1", respectively, followed by "@SERIES = EXPER" for the two union values.

Similarly, we may display a categorical boxplot with FEM, UNION and @SERIES as within graph factors.

First, double click on the graph, being careful to select the entire graph, to display the dialog and change the **Specific** graph type to **Boxplot**. Next, move the UNION factor to the end of the **Within graph** edit field, and change the multiple series dropdown to **Last within factor**.

The resulting graph displays eight boxplots in a single graph frame. The implicit factor @SERIES has been placed at the end of the within list so that it varies fastest. We see that LWAGE and EXPER are displayed for each level of FEM and UNION, that the levels of UNION vary within each level of the first factor FEM.

Categorization by FEM, UNION
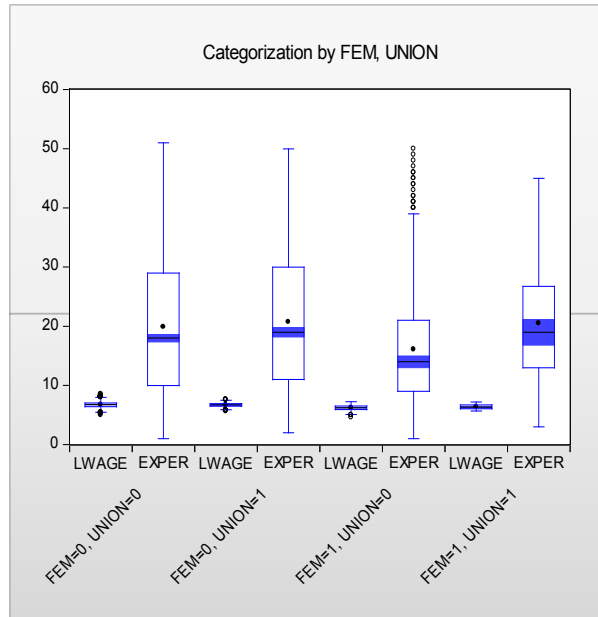
Identifying Categories

The second major type of categorical graph displays the raw data along with category identifying information.

We consider four representative examples of these graphs: a scatterplot, a spike plot, a line plot, and a dot plot. The first two examples, which involve multiple observations in each category, use the **Raw Data** setting for the **Graph data** dropdown; the last example, where there is a single observation in each category, uses the special **Unique values – error if not identical** setting.

Raw Data

We consider here two categorical graphs that employ the **Raw Data** setting in the **Graph data** dropdown. As you might expect given the name of the setting, these graphs all display the underlying (raw) data in the series.

One commonly employed raw data categorical graph is a scatterplot where observations in each category are displayed with a different symbol. Our first two examples use data in the

workfile "Mpg.WF1" on EPA reported miles-per-gallon and engine size (displacement) for a subset of 2006 model year automobiles.

We construct a categorical scatterplot of city miles-per-gallon (CITY_MPG) against engine size (DISPLACEMENT), using the number of cylinders (NUMB_CYL) in the engine as a within factor. First, create a group containing DISPLACEMENT and CITY_MPG. Simply specify **Categorical graph** and **Scatter** as the **Graph type**, leave the **Graph data** setting at **Raw data**, and enter "NUMB_CYL" in the **Within graph** edit field. To draw a set the linear regression lines through the points in each class, set the **Fit lines** dropdown to **Regression line**.

The resulting graph uses color and symbol choice to identify categories. Since we have selected **Raw data**, every valid observation in the sample is displayed using category specific colors and symbols. Not surprisingly, we see that engines with greater numbers of cylinders have a larger displacement. More interestingly, there appears to be a weaker relationship between DISPLACEMENT and
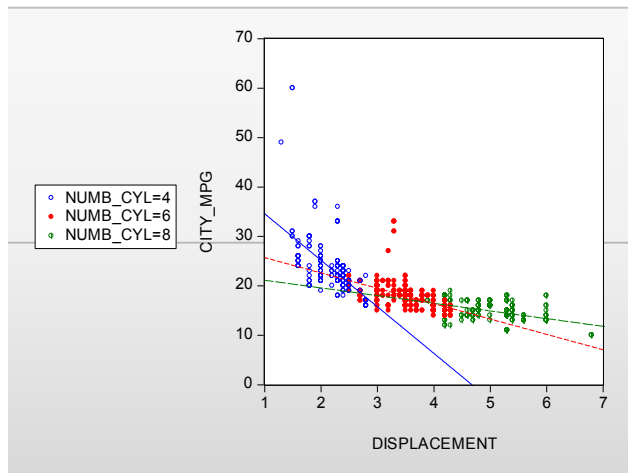
CITY_MPG for cars as the number of cylinders increases, though the two high MPG outliers may be unduly influential in that comparison.

We may compare this categorical graph to one in which we treat NUMB_CYL as an across factor:
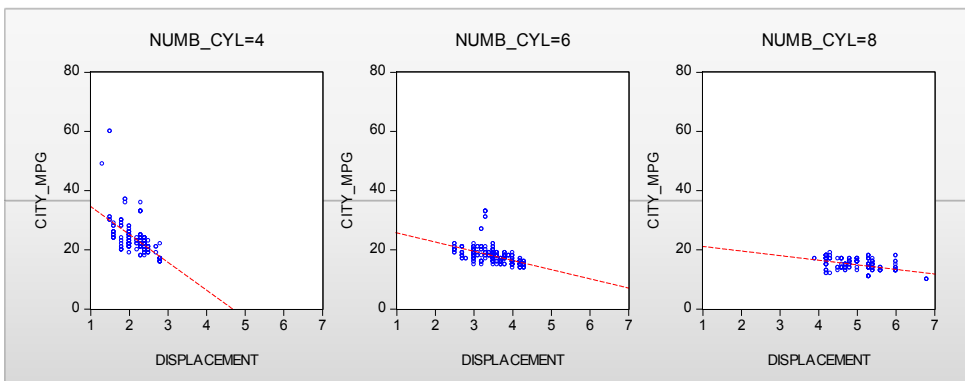
Here, each set of raw data points is displayed in its own graph frame, using common axis scaling. The points all use the same color and symbol since the graph frame titles are sufficient to identify the group displayed in the frame.

Our second example uses categorical raw data graphs to explore differences in the regression fit of CITY_MPG to DISPLACEMENT. We first estimate the linear regression of CITY_MPG on DISPLACEMENT then save the residuals from this equation to the series MPG_RESID. Note that the equation results in EQ01 assume a common slope coefficient on DISPLACEMENT; the scatterplots above suggest that this assumption is not valid.

As further evidence that the equation assumptions are not valid, we display a categorical bar plot of MPG_RESID using NUMB_CYL as the within factor. This graph shows each value of MPG_RESID, with observations in different classes drawn using different colored bars.
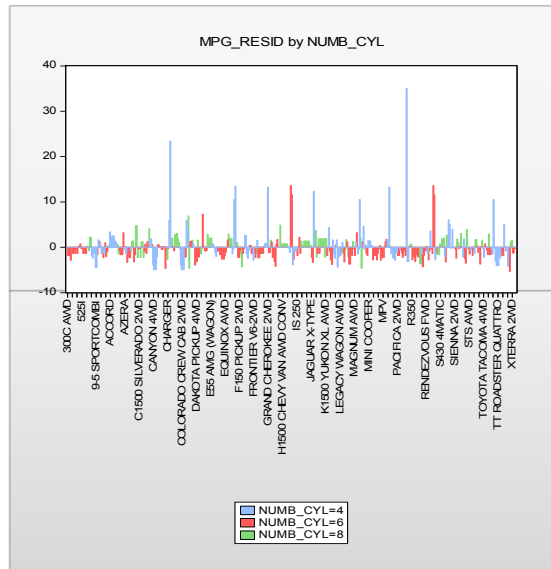
While it may be a bit difficult to see in the printed black-and-white form of the graph, the size of the equation residuals appears to be negatively related to the number of cylinders; in particular, almost all of the very large positive residuals are for 4-cylinder vehicles. The correlation between residuals and number of cylinders suggests that, at the very least, number of cylinders is an omitted variable in the equation. Note that EViews shows only observation indices since there is insufficient space to show observation labels.

The visual comparison of residuals for cars with different numbers of cylinders may be facilitated by treating NUMB_CYL as an across factor:
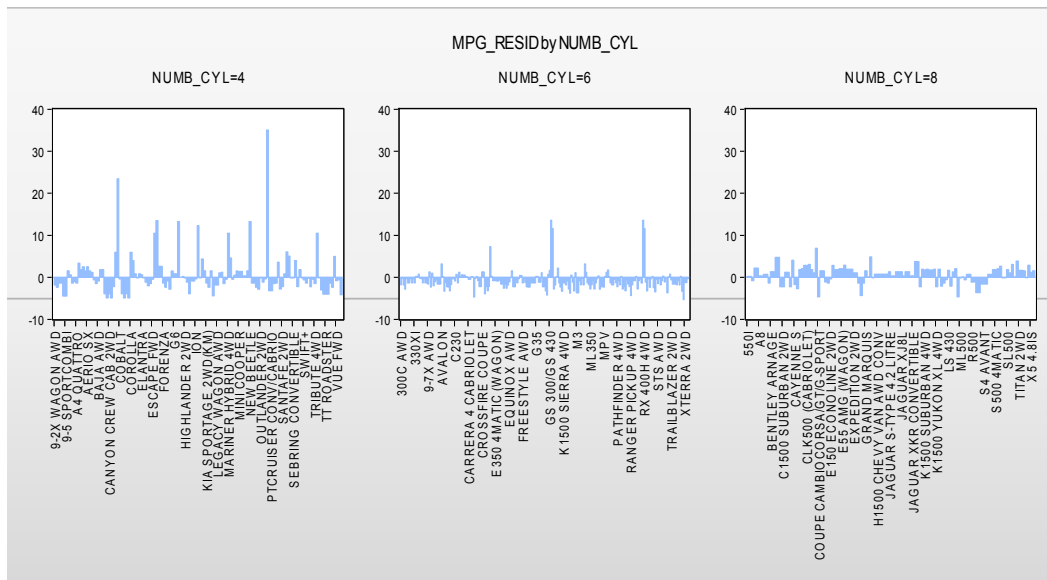
The negative relationship between number of cylinders and the size of residuals is readily apparent in this graph. Note that since there are fewer observations plotted in each of the graph frames, EViews switches to showing some of the observation labels from the workfile.

Unique Values

Our final example uses the **Unique values – error if not identical** data setting.

We again employ the workfile "Gulfcoast.WF1" containing population information for counties in the Gulf Coast region of the United States. For this example, we restrict ourself to displaying values for counties in Louisiana by setting the sample to only include observations where the STATE_NAME = "Louisiana".

We display the percentage change in population for counties in Louisiana in 2005 and 2006 using a categorical dot plot. The categorical plot uses the factors COUNTY_NAME and YEAR, with YEAR entered last in the within list since we want to compare population values in the two years for a each county.

A slightly customized version of the graph is depicted here. The filled circles represent the proportionate changes in population in 2006; the open circles represent the 2005 changes. We see that there is a large disparity in the effect of the hurricanes across counties, with three counties: St. Bernard, Orleans, Plaquemines, and to a lesser extent Cameron and Jefferson bearing the brunt of the impact.

Constructing this particular graph is straightforward, requiring only a few steps.

First, we display the graph dialog for PDIFF and set the **Graph Type** to **Categorical graph** and **Dot Plot**, and choose **Rotated – obs axis on left** from the **Orientation** dropdown.

Since we want to plot a graph grouping different years for each county together, we enter "COUNTY_NAME YEAR" in the **Within graphs** edit field.

PDIFF by COUNTY_NAME, YEAR
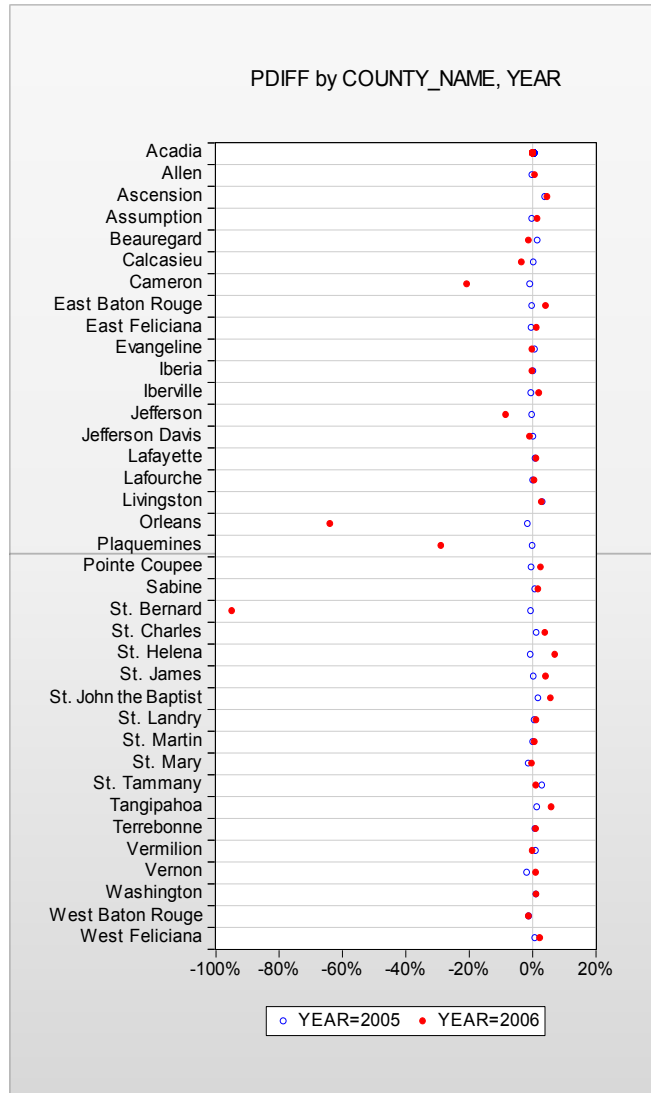
Next, despite the fact that we wish to plot every observation in the sample, we set the **Graph data** setting to **Unique values – error if not identical**. Note that this is a change from previous examples where we used the **Raw data** setting.

This latter choice requires a bit of discussion. Since we are displaying a plot of every observation (every county and period) in the sample, you might at first think of selecting **Raw data** for this setting. Recall, however, that using **Raw data** will produce a plot with each

observation identified in some way as belonging to a category. In this case, since every observation is in a different category (county and period), selecting **Raw data** will produce a dot plot that uses a separate row and symbol for *every observation*. This is obviously not the desired effect.

Selecting **Unique values – error if not identical** tells EViews that (using the default settings in the **Categorical options** dialog) despite the fact that we are plotting every observation, we want to plot both year values for a single COUNTY_NAME on a single row, and that we want to use unique graph elements across years, but not across counties. Thus, different YEAR observations are given different symbols within a county, but the set of symbols used to identify the two years is the same across different counties.

If all of this seems rather abstract or mysterious, we will examine this issue in greater depth in "Specifying Factors," on page 734. For now, you may follow a simple rule-of-thumb: if your factors define groups containing only one observation each, you generally should select **Unique values – error if not identical** to obtain the desired graph.

Lastly, we use the **Size & Indents** dialog under the **Frame** section to change the height and aspect ratio to 6 and 0.50, respectively. Then use the **Grid Lines** dialog under **Axes & Scaling** to turn on horizontal gridlines by selecting **Custom** from **Obs & Date axis grid lines**, with a **Step** of 1. In addition, we employ the **Data axis labels** dialog, also under **Axes & Scaling**, to display the bottom axis scale in **Units** of **Percent**, with a "%" suffix, and to draw a zero line.

Specifying Factors

Categorical graphs use factor variables to define subsets of data. In the simplest case, a categorical graph is based on a a single factor variable containing a small number of discrete values; subsets of the data are defined for observations with each of these values. In this basic setting, specifying the factors for the graph involves little more than providing the name of the factor variable and indicating whether it should vary within or across graph frames.

More complicated situations can be constructed involving multiple factors or non-categorical factor variables. These cases raise a number of issues associated with how to define the categories for the factor and how to organize the subsets of the data for display. How these issues are resolved has a profound impact on the appearance of the categorical graph.

Accordingly, the factor specification for a categorical graph may involve much more than simply providing a list of factors. While the EViews defaults will generally produce the desired graph, you may need to customize the factor specification in more complicated settings. The remainder of this section outlines the default rules that EViews uses for specifying and organizing factors, and describes rules for customizing the factor specification.

Defining a Factor Categorization

In most cases, you will specify a factor variable that contains a small number of discrete values. These discrete values will be used to define a set of categories associated with the factor.

Suppose, for example, that we have the factor variable, FEM, indicating whether the individual is a 0 (Male) or 1 (Female). The two distinct values 0 and 1 will be used to define the categories for the factor and each individuals in a sample will be categorized on the basis of whether they are 0 or 1.

You may also specify a factor variable that is non-categorical, or one with a large number of distinct values. For example, suppose you propose the use of the series INCOME, which measures individual incomes, as a factor variable. The use of this variable creates difficulties since income does not have a small number of categories; indeed, every observation will be in its own category.

By default, EViews tries to avoid this situation by analyzing each factor to determine whether it appears to be categorical or continuous. If EViews determines that the variable is continuous, or if there is a large number of categories associated with the factor, EViews will define a new categorization by automatically binning the factor into five categories defined by the quintiles of the series.

You may override the EViews default categorization settings using the **Categorical options** dialog under the **Graph Type** section (note that this dialog is only available when **Categorical graph** has been selected for the **Graph type**). Simply select the factor whose options you wish to change in the left-side list box, then select the desired entry in the **Binning** dropdown menu. The default setting, **Automatic**, uses **Quantile bins** if there are a large number of distinct values for the factor, and **No binning** otherwise. You may choose either of the latter two methods directly, or tell EViews to create **Value bins** by grouping data on the basis of equal width intervals. For both **Quantile bins** and **Value bins**, EViews will prompt you for the number of bins to use. The default number of bins is 5.

For example, we again consider the workfile "Mpg.WF1" which reports EPA reported miles-per-gallon and engine size (displacement) for 468 automobiles. We first display the categorical bar graph of the mean of CITY_MPG using the categorical variable DISPLACEMENT as a within factor.

Mean of CITY_MPG by DISPLACEMENT

There are 35 distinct values in the DISPLACEMENT series. EViews automatic binning settings allow DISPLACEMENT to be used as an unbinned factor.

By default, EViews attempts to label all 35 categories in the resulting graph. The graph may be a bit busy for some tastes, even if we only show the factor levels. One alternative is to display a binned version of this graph where we define categories based on intervals of the DISPLACEMENT values.
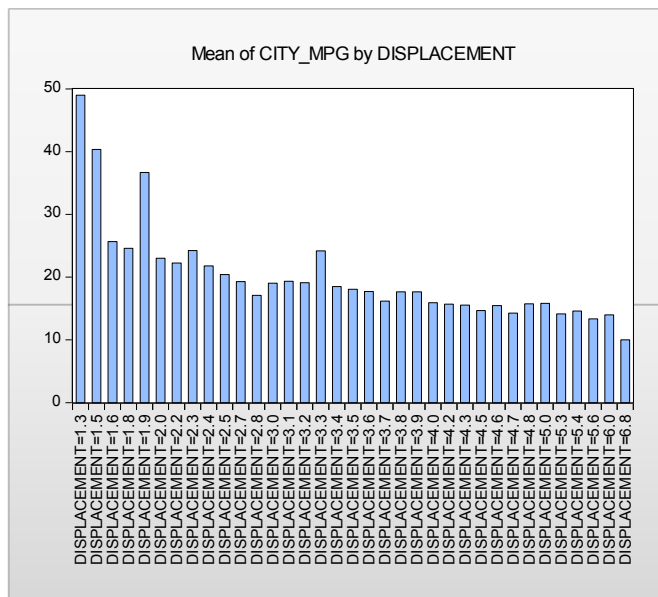
Click on **Categorical options** under **Graph Type** in the main dialog, select DISPLACEMENT in the left-hand side list box, then change the **Binning** dropdown to **Value bins**. For these settings, EViews will create a factor using (at most) 5 equal-width bins based on the values of DISPLACEMENT. Click on **OK** to accept the options.

The resulting graph shows that EViews categorizes observations into one of four DISPLACEMENT ranges: [0, 2), [2, 4), [4, 6), [6, 8). The mean MPG for cars with engine size under 2 liters is roughly 29, while the mean value for engines from 6 to 8 liters is under 14. While the negative relationship between engine size and miles-per-gallon can be seen in the earlier graph, it is more apparent in the binned version.
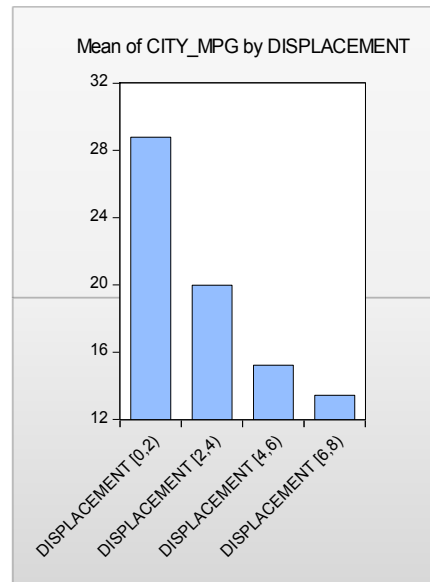
It is worth noting that binning on the basis of custom thresholds is not directly supported in graphs. If you wish to define custom bins, you should use the series classification Proc to define a new categorical variable (see "Stats by Classification" on page 405 for details), and then use the new variable as your factor.

Factor Display Settings

Having defined factor categories for one or more factors, there are several basic settings that will control the appearance of your graph: whether to display factor levels within or across graph frames, the ordering of factor levels, the ordering of multiple factors, and for summary graphs, the assignment of graph elements to factor levels and the method of labeling factor categories.

Within vs. Across

You should enter your factor names in the **Within graph** and **Across graphs** edit lists on the main graph options page. Each level of a factor entered in the **Across graphs** factor list will be displayed in a separate graph frame, while levels of factors in **Within graph** will be displayed in single frames.

In addition, if you are plotting multiple series in a group, you will be prompted for whether to treat the different series as an across or a within factor, and to specify the factor ordering (whether the factor should be placed at the beginning or end of the list).

A number of the case studies in "Illustrative Examples," beginning on page 717 demonstrate the effects of these choices.

Factor Levels Ordering

By default, EViews orders the categories formed from each factor from lowest to highest value. Categories formed from numeric values will be sorted numerically while categories formed from alphanumeric factors will be sorted alphabetically. The order of categories is then used in constructing the graph.
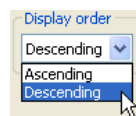
To change the ordering of levels for a given factor, click on **Categorical options** to display the options dialog, select a factor in the left-hand side list box, then change the **Display order** dropdown from the default **Ascending** to **Descending**.

We may, for example, modify our categorical graph for CITY_MPG using the binned values of DISPLACEMENT. Double click on the graph to open the main graph dialog, click on **Categorical options** to show the options dialog, and change the display order.

Note that changing the ordering of the levels changes the order in which they are displayed in the graph. The categories for DISPLACEMENT now start at the largest level for the factor and continue on through the smallest.
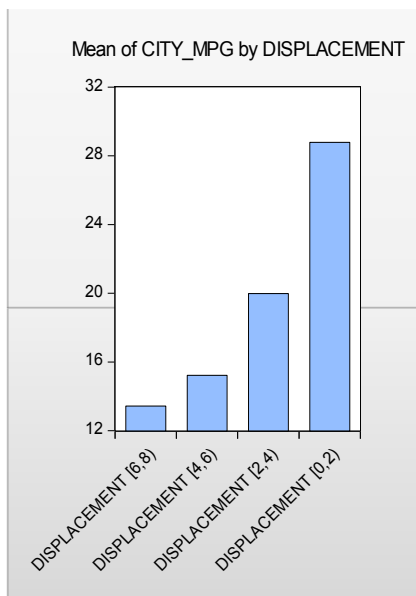
Multiple Factor Ordering

You may specify more than one factor variable, thereby forming a set of categories defined by each combination of the distinct factor values. In this case, the order in which the factors vary has an important effect on the final display.

Suppose, in addition to the FEM variable, you have a second factor variable UNION representing whether the individual is in "Union" or "Non-union" employment. Then the four categories for these two factors are: {("Male," "Non-union"), ("Male," "Union"), ("Female," "Non-union"), ("Female," "Union")}.

Note that in this list, we have arranged these factors so that:

Order	FEM	UNION
1	"Male"	"Non-union"
2	"Male"	"Union"
3	"Female"	"Non-union"
4	"Female"	"Union"

with the "Male" categories coming first, followed by the "Female" categories, and with the UNION status categories varying within the FEM categories. We say that the FEM factor varies more slowly in this ordering than the UNION category since the latter varies within each level of FEM.

Alternately, we can reverse the ordering so that the FEM factor varies more rapidly:

Order	FEM	UNION
1	"Male"	"Non-union"
2	"Female"	"Non-union"
3	"Male"	"Union"
4	"Female"	"Union"

so that the GENDER values vary for each level of UNION.

We may extend this notion of ordering to more than two categories. Suppose we have a third factor, YEAR, representing the year the individual is observed, with three distinct values 1980, 1981, and 1982. Then if FEM varies most slowly, UNION next most slowly, and YEAR most rapidly, we have:

Order	FEM	UNION	YEAR
1	"Male"	"Non-union"	1980
2	"Male"	"Non-union"	1981
3	"Male"	"Non-union"	1982
4	"Male"	"Union"	1980
5	"Male"	"Union"	1981
6	"Male"	"Union"	1982
7	"Female"	"Non-union"	1980
8	"Female"	"Non-union"	1981
9	"Female"	"Non-union"	1982
10	"Female"	"Union"	1980

11	"Female"	"Union"	1981
12	"Female"	"Union"	1982

The first three cells correspond to {"Male," "Non-union"} workers in each of the three years, while the first six cells correspond to the "Male" workers for both union and non-union workers in each of the three years.

When specifying factors in the main **Graph Options** page, you will enter the factors in the **Within graph** or **Across graphs** list. Within each list, factors should be ordered from slowest to fastest varying. Factors listed in the **Across graphs** list are always more slowly varying than those in the **Within graphs** list since each across graph category is displayed in a separate graph frame.

The first example in this section uses the ordering:

```
fem union
```

so that FEM varies more slowly than UNION. The second example reverses the ordering of the two factors so that UNION varies more slowly:

```
union fem
```

The last example orders the factors so that FEM varies most slowly, and YEAR most rapidly:

```
fem union year
```

Various examples of the effect of reversing the ordering of factors are provided in "Illustrative Examples," beginning on page 717.

Assigning Graph Elements to Categories

One of the most important decisions you will make a within categorical summary graph is choosing the elements for displaying data for different categories. While EViews provides you with reasonable defaults, there are useful features for customizing these choices that you may find useful.

(The choices described in this section are not relevant for non-summary categorical graphs specified by selecting **Raw data** in the **Graph data** dropdown on the main graph dialog).

To understand the basic issues involved in these choosing graph elements, we must first divide our within factors into two groups: *primary* and *secondary factors*. Primary within factors are a subset of most slowly moving factors whose levels share common graph elements (*e.g.*, colors, line patterns, shades). The remaining secondary factors display different levels with different graphic elements.

You may think of the primary factors as defining the set of categories that yield summary "observations" so that they are arrayed along the axis, with the secondary factors defining subsets within these categories (much in the same way that one may draw minor ticks

between the major ticks on a graph axis). We then apply the general rule that primary factors share common graph elements across levels, while secondary factors use different graph elements for different categories. The interpretation of primary factors as being categories displayed the axis with secondary factors specified as subsets of the primary factors is an important one that we will explore further.

As is often the case, some examples will best illustrate the basic ideas. We return to the earlier example of constructing a binned categorical graph for mean of CITY_MPG divided into ranges of DISPLACEMENT. We begin by displaying a bar graph showing the categorical means:
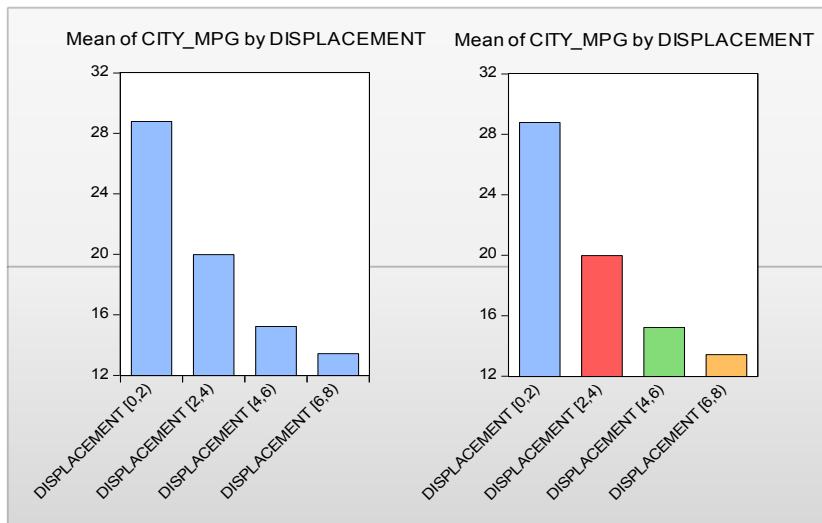
On the left is the graph using default settings where DISPLACMENT is treated as a primary factor, while on the right is a graph with DISPLACMENT treated as a secondary factor. Note that on the left, the levels of the primary factor DISPLACEMENT use the same graph element (bar color), while on the right, the levels of the secondary factor DISPLACEMENT use different bar colors.

Before examining examples of the more complex settings, let us first see how we modify the default settings of the graph on the left to obtain the graph on the right. Click on **Categorical options** to display the options dialog. At the bottom of the dialog is the descriptively titled **Within graph category identification** which provides control over the assignment to major and minor factor categories, and as we will see later, the labeling of these categories.

As the name suggests, the verbosely labeled dropdown menu **Give common graphic elements (lines, bars, colors, etc.) to categories for factors**
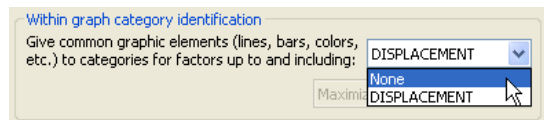
up to and including, selects the set of factors are to be given common graphic elements. Since the primary factors must be the most slowly varying factors, assigning factors to the primary and secondary groups is the same as choosing a cutoff such that factors up to and including the cutoff are primary factors, and factors following the cutoff are secondary factors.

In the single factor case setting, the dropdown default is set so that factor is primary so that all graph elements are common; in this example, the dropdown is set to **DISPLACEMENT**. The graph above on the left, with all bars displayed using the same color, shows the default setting. Changing the dropdown to read **None** indicates that there are no primary factor, only the single secondary factor, as in the graph with different colored bars on the right.

While informative, our bar graph example hides one very important difference between the two graphs. Recall that one interpretation of the difference between primary and secondary factors is that the levels of the primary factors are placed along the axis, with secondary factors defining subsets within these major categories. In our example, there are four distinct categories along the axis in the left bar graph and only one category on the axis in the right graph. The different numbers of categories along the axis is hidden in bar graphs; since the latter always offset bars drawn for different categories it is difficult to tell the difference between the primary and secondary factor categories.

We may, see the importance of this difference when switching from a bar graph to a dot plot:
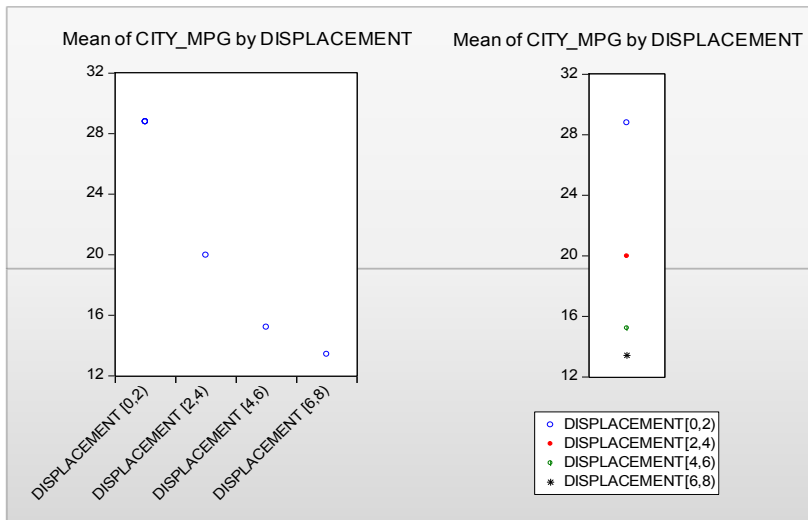
In the graph on the left, DISPLACEMENT is a primary factor so that each level of the factor is displayed as a separate "observation" along the axis using a common symbol and color for the dot. In the graph on the right, DISPLACEMENT is a secondary factor that is displayed

using different symbols and colors for each level of the primary factor. Since there is no primary factor in this case there is only a single observation on the axis, and all four symbols are lined up on that single observation.

For a slightly more complicated example, we again use the "Wages.WF1" workfile containing information on log wages for a sample of 4165 individuals. We will use the three series FEM, UNION, and YEAR as within factors, entered in that order, and will display a dot plot of the means for this categorization using the default settings.

For more than one within factor, the default is to designate only the last listed factor as a secondary factor. At the default setting, the dropdown menu in our example is set to **UNION**

so that FEM and UNION are primary factors for the graph, while YEAR is as secondary factor.
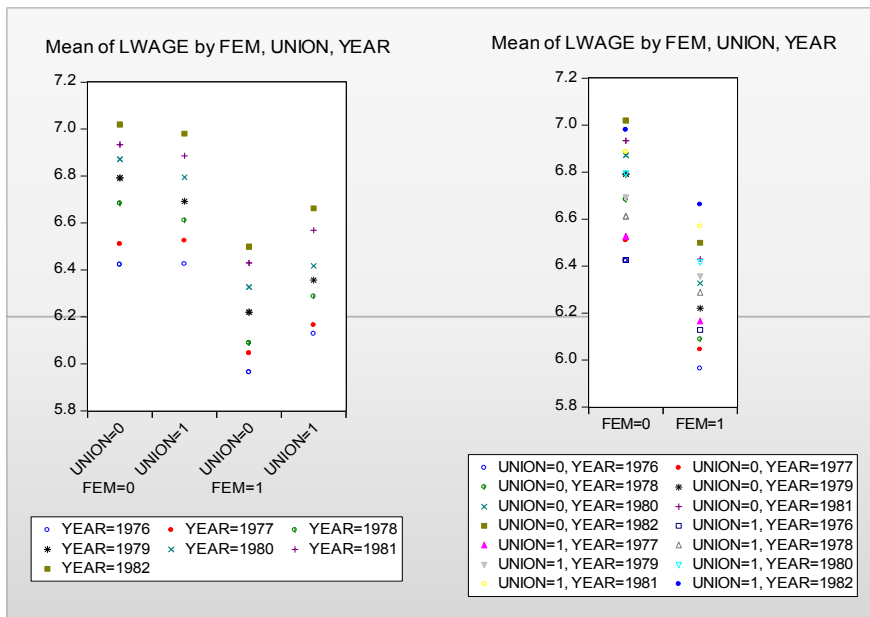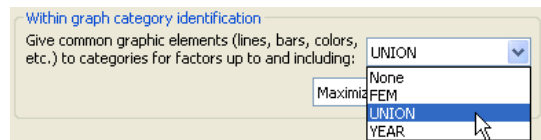
The resulting graph, shown on the left, has several notable features. First, the four distinct categories formed from the primary factors FEM and UNION are each assigned to the graph axis. Within each level of the primary factors, we see distinct symbols representing the various levels of the secondary YEAR factor. Lastly, the set of symbols is common across primary factor levels (*e.g.*, all four of the "YEAR = 1976" symbols are blue circles).

Changing the dropdown menu to FEM produces the graph on the right. Since FEM is the sole primary factor, EViews assigns the two levels for FEM to the graph axis, with the remaining factors treated as secondary factors.

For our next example, we consider the group object GROUP01 containing the series EXPER and LWAGE. We display scatterplots of the categorical means for these two series given the three within factors FEM, UNION, and YEAR, along with regression fit lines.
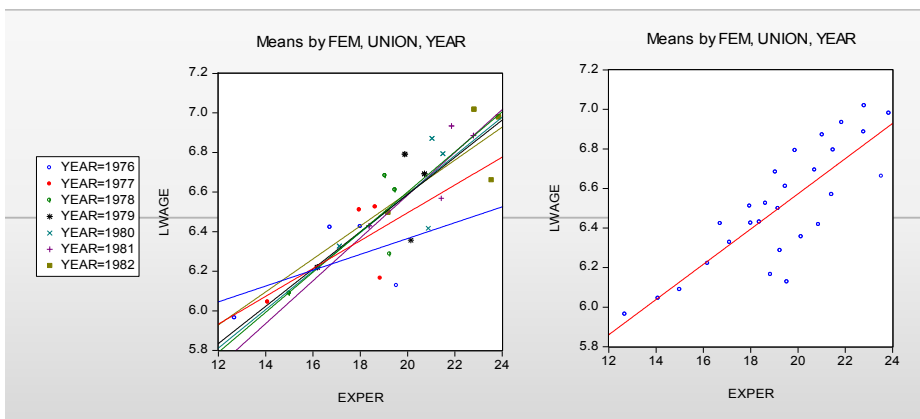
The scatterplot on the left uses the default setting so that FEM and UNION are primary categories, and YEAR is a secondary category. Mean values are plotted for each category, with different symbols used for different levels of YEAR. Following the principal that primary factors define observations, regression fit lines are computed for each level of the secondary category across levels of the primary factor. Thus, the fit line for YEAR = 1977 shows the regression fit obtained using the four mean values of LWAGE and EXPER in the categories defined by levels of FEM and UNION.

In contrast, setting the dropdown to YEAR so that all factors are primary yields the plot on the right. All of the points use the common symbols, and the fit line is fitted across all of the primary factor levels.

The basic principle here is that if you wish to draw fit lines for summary statistics across categories, those categories should be specified as primary factors.

Parenthetically, we are now in a position to explain the apparently anomalous ordering of factors in our wage-experience profiles above ("Line Graphs," on page 722). Recall that the displaying separate average wage-experience profiles for men and women in a single graph frame required that we use the within factor list "EXPER FEM" despite the fact that EXPER appears to vary more rapidly that FEM.

An examination of the default settings for the graph reveals that EXPER is a primary factor, while FEM is a secondary factor. Since the levels of EXPER are observation identifiers that are displayed along the axis, line graphs connect the EXPER levels, making it appear that EXPER varies rapidly, even though the points are with FEM varying for each level of EXPER.

Here, we see the dot plot corresponding to the earlier line graph. FEM clearly varies more rapidly as both the FEM = 0 and FEM = 1 points are plotted for each level of EXPER. The line graph version of this graph simply connect points across observations (experience

levels) for each level of FEM and turns off the symbols, making it appear as though EXPER is varying more rapidly.
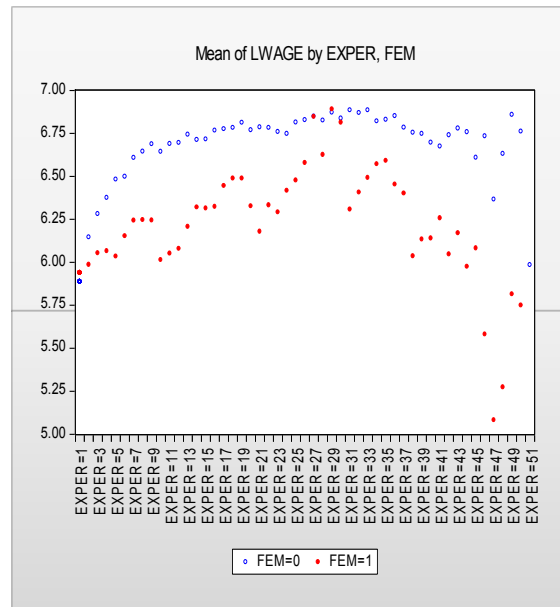
Our last example ties together all of the various concepts. Suppose that we were to plot the average log wage against year using FEM and UNION as our factors. There are two distinct approaches to constructing this graph. In the first approach, we specify a single observation scale using YEAR and draw four different wage-year profiles, one for each category formed by FEM and UNION. In the second approach, we adopt a "panel" style graph in which divide the factor scale into two panels, with the first panel representing a YEAR scale for males, and the second panel representing a YEAR scale for females. We show the two cases below:
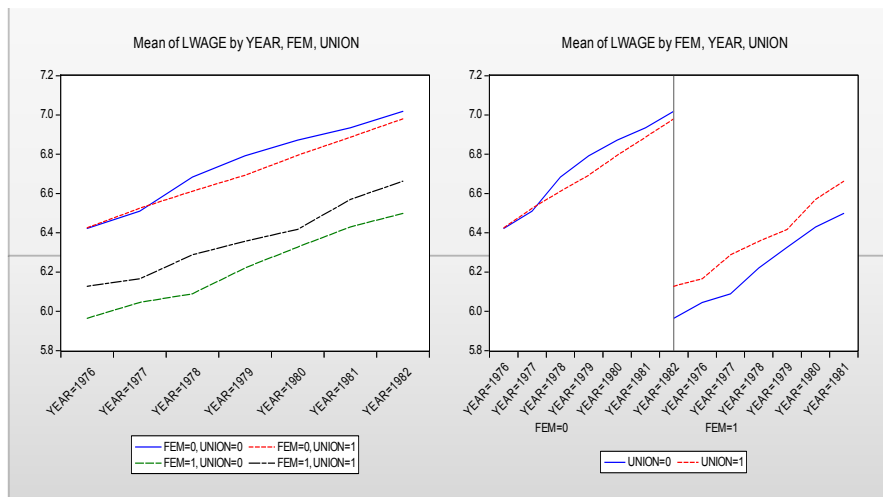
The graph on the left specifies the within factor list as "YEAR FEM UNION", with YEAR the sole primary factor, and FEM and UNION the secondary factors. The axis scale uses YEAR to identify observations, and for each secondary factor category draws a line connecting the observations for that category. In contrast, the graph on the right uses the within factor list "FEM YEAR UNION", with FEM and YEAR as the primary factors. The axis scale uses FEM and YEAR for observations, with YEAR varying for each level of FEM, and for each level of the secondary factor connects the lines across the observations for each factor. Note that EViews knows not to connect lines across levels of the FEM factor.

(Note: we have customized the graph on the right slightly by freezing the graph, and turning on **Segment with lines** in the **Sample breaks** section of the **Basic type** page.

The rule-of-thumb to remember here is that the factor that you wish to connect using a line graph or XY line graph, should be specified as the last primary factor. Specifications with one primary factor will have a set of lines for each secondary factor factory; specifications with more than one primary factor will be displayed in paneled form.

Factor Labeling

By default, EViews will label factor levels in summary graphs using some combination of axis labels and legend entries. For line graphs and XY graphs, the EViews choices are the only possible way to identify the levels. For other types of summaries, we may choose to display the bulk of the label information along the axis, or we may choose to display most of the information in legend entries.
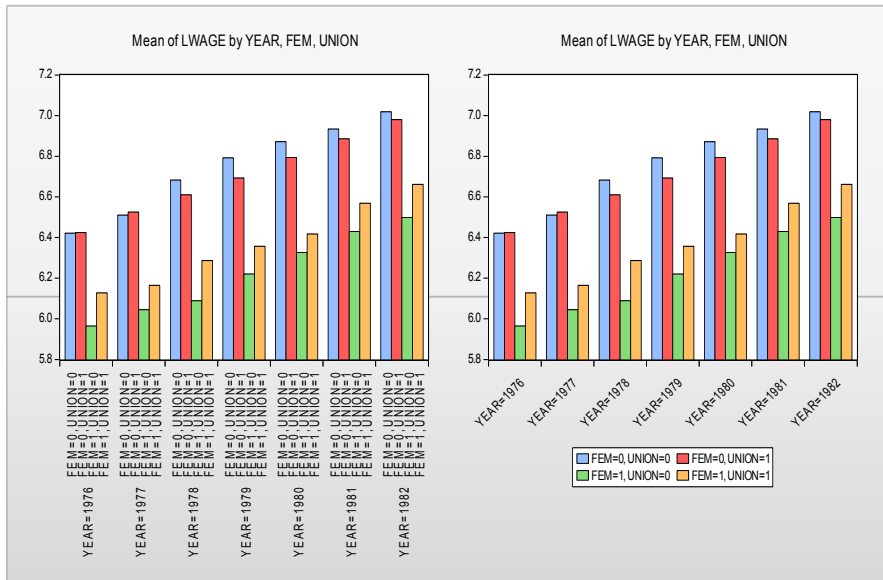
Both of the graphs displayed here are summary bar graphs of LWAGE categorized by YEAR, FEM and UNION. In the graph on the left, we display all of the category information using two-level labels along the axis, while in the graph on the right, we display the information using a single level axis label combined with legend entries.

By default, EViews will, if possible, place the category information along the axis. You may choose to override this default using the **Categorical options** dialog. At the bottom of the options dialog, in the **Within graph category identification** section, there is a dropdown menu which allows you to choose between the default, **Maximize use of axis labels**, or the alternative, **Maximize use of legends**, which encourages the use of legend information. The graph on the left above was obtained using the default setting, while the graph on the right was obtained by encouraging the use of legend information.
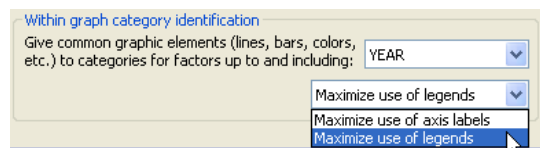
We emphasize again that this dropdown menu does not affect the category labeling for Line & Symbol, Scatter, and XY Line plots.

Part III. Customizing Output

EViews objects (series, groups, equations, and so on) display their view and (sometimes) procedure output in the form of graphs, tables, and text. You may, for example, display the descriptive statistics of a set of series, or the regression output from an equation as a table, or the impulse responses from a VAR as a graph. We will term these displays *object views*.

While object views may be customized in a variety of ways, they are generally transitory; when you close the object and subsequently redisplay or switch between views, many of the customized settings are lost. And in cases where the views are dynamic, the view is regenerated automatically when the underlying object or the active sample changes, resulting in the loss of any custom settings.

Fortunately, you may preserve the current object view, along with any customization, so that it does not change when the object changes. We refer to this action as *freezing* the view. Freezing a view will create a new *output object* containing a "snapshot" of the current contents of the view window. The type of object created when you freeze a view depends on the original view—freezing a graphical view creates a graph object, freezing a tabular view creates a table object, and freezing a text view creates a text object.

Graph, table, and text objects form the basis of presentation output, and EViews provides sophisticated tools for customizing the appearance of these objects. EViews also offers a *spool object* which allows you to manage collections of output objects. Spool objects may be used for creating a log of the output created during a project or an EViews session, or for gathering together graph, table, and text output for a presentation.

The following chapters describe the use of graph objects, table objects, and spool objects in greater detail, showing how you may use these objects to form presentation output:

Chapter 15. Graph Objects

In Chapter 13. "Graphing Data," on page 617 and Chapter 14. "Categorical Graphs," beginning on page 717 we described in detail EViews tools for producing graphical displays of your data. The topics considered range from simple observation line graphs, to analytical graphs such as histograms, to even more complex categorical graphs.

These graphical displays are what we term *graph views* since they are simply a different way of viewing the data in the underlying data object. While graph views may be customized in a variety of ways, they are generally transitory; when you close the series or group object and subsequently redisplay or switch views, many customized settings are lost. And in cases where the views are dynamic, the view is regenerated automatically when the underlying object or the active sample changes, resulting in the loss of any custom settings.

Fortunately, you may create a *graph object* by *freezing* a graph view. Freezing a view will create a new graph object containing a "snapshot" of the current graph view, allowing you to perform permanent customization of the graphical output. Graphs may also be placed into a spool object along with table and text objects for use in custom presentations (see Chapter 17. "Spool Objects," on page 801).

This remainder of this chapter describes the options available for customizing the appearance of a graph object. This chapter does not offer a comprehensive examination of all of the possible customizations you may perform; we encourage you to experiment with various settings to see the effect on your output.

Creating Graph Objects

Graph objects are usually created by freezing an object view. Simply press the **Freeze** button in an object window containing a graph view.

It is important to keep in mind the distinction between a graphical view of an object such as a series or a group, and a graph object created by freezing that view.

For example, suppose you wish to create a graph object containing a line graph of the series LPASSENGER from the workfile "Airline.WF1". To display the line graph view of the series, select **View/ Graph...** from the LPASSEN-GER series menu and select **Line & Symbol** from the list-box in the middle of the dialog. Notice the "Series: LPASSENGER" designation in the window titlebar that shows this is a view of the series object.
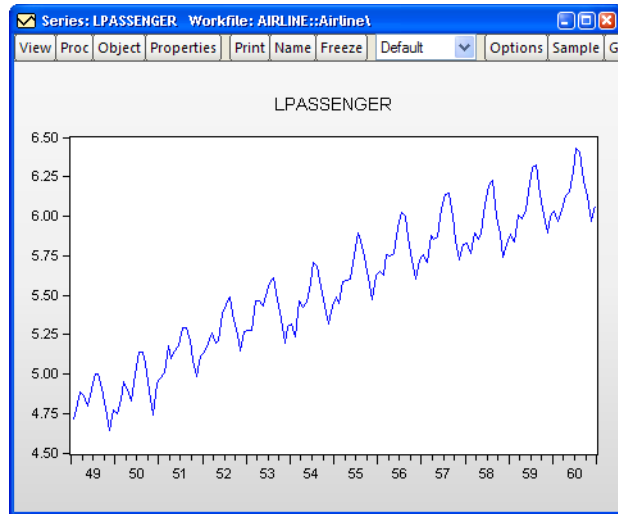
You may customize this graph view in any of the ways described in "Customizing a Graph" on page 38, but many of these changes will be lost when the view is redrawn, *e.g.* when the object window is closed and reopened, when the workfile sample is modified, or when the data underlying the object are changed. If you would like to keep a customized graphical view, say for presentation purposes, you should create a graph object from the view.

Freezing a Graph View

To create a graph object from the view, click on the **Freeze** button. EViews opens the **Auto Update Options** dialog, prompting you to chose whether to make your new graph updating, and if so, to specify a few update options. Let's discuss the difference between an updating and a non-updating frozen graph.

Selecting **Off - Graph is frozen and does not update** will create an untitled graph object containing a snapshot of the view. (This option mimics the behavior of the **Freeze** button in EViews prior to version 7.) The data has been copied from the
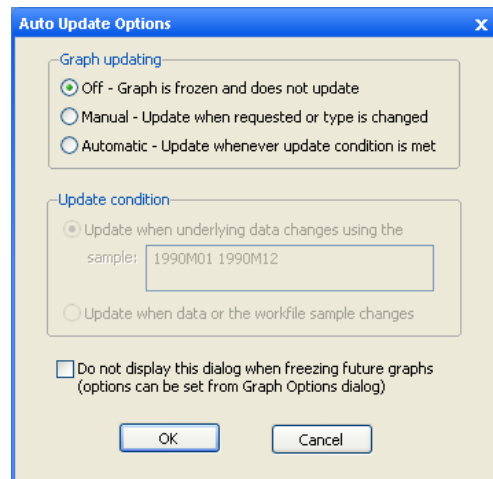
series and is stored in its current state in the graph object. Changes to the series data will not affect the graph.

Alternately, choosing **Manual - Update when requested or type is changed** or **Automatic - Update whenever update condition is met** will create an updating graph. This means that the underlying graph data is still tied to the original series, so when the series values change, the graph has the opportunity to update as well.

The choice between **Manual** and **Automatic** is based on when you would like the graph to respond to changes in the data. Selecting **Manual** allows you to control when the updates occur. The graph will remain the same until you push the **Update** button on the graph toolbar, or until the graph type is changed. Choose **Automatic** if you'd like EViews to keep the graph current with changes to the underlying data.

Whether you are working with a manually or automatically updating graph, you can specify how you would like changes in the data or workfile sample to affect the graph when it is updated. Selecting either **Manual** or **Automatic** will enable the **Update condition** section of the dialog. If you'd like the graph to plot over a constant sample that may or may not match the current sample of the workfile, select the first option, **Update when underlying data changes using the sample**. Enter the desired sample in the edit field, or simply accept the current default sample. The graph will now only update data that fits within the specified sample period. In addition, changes to data outside of this sample period will not trigger an update for automatically updating graphs.

Alternately, select **Update when data or the workfile sample changes** to accept all changes to the data and workfile sample when the graph is updated. Note this means that a change in the workfile sample will be reflected in the graph, whereas the **Update when underlying data changes using the sample** option keeps the graph sample constant unless you change it in the main **Graph Options** dialog.

For example, let's look at freezing the line graph view of the series LPAS-SENGER. The workfile contains monthly data from 1949 through 1960. First, we'll concentrate on a portion of the data by changing the workfile sample period to 1949 through 1952. Then freeze this view, selecting **Automatic** updating and **Update when underlying data changes using the sample:** in the **Update condition** group. Leave the default sample in the edit field to restrict the graph to changes within the first few years of the workfile data. Click **OK** to create an untitled graph object.
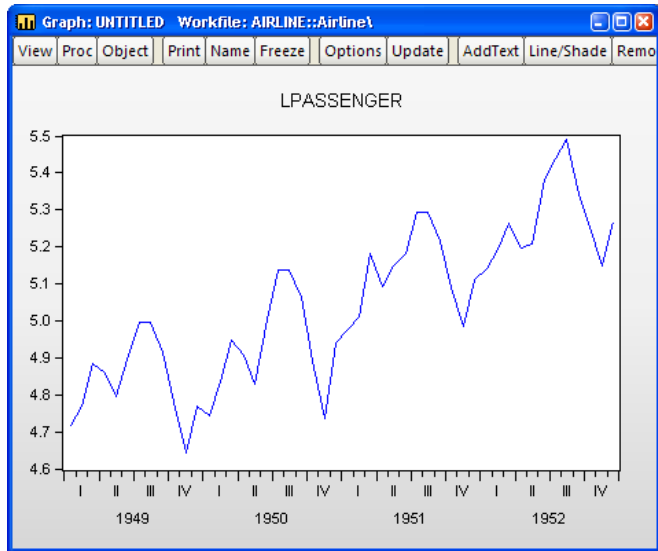
Now, let's see how changes to the series data affect the frozen graph. If we change the value at 1949M01, we'll see the graph plotting the new value. However, modifying the value at 1956M01 has no effect on the graph, since this date is outside the graph's sample range.

The real difference between the two options in the **Update condition** group can be seen when we modify the workfile sample. Since we have specified an update sample for the current graph, changes to the workfile sample will not have an effect on the graph.

Alternately, if we had selected **Update when data or the workfile sample changes** in the **Auto Update Options** dialog when freezing the graph, we would see different behavior. Create a graph with the second **Update condition**, and then expand the workfile sample back to 1949 through 1960. Once updated, the graph will expand its sample to match that of the workfile.
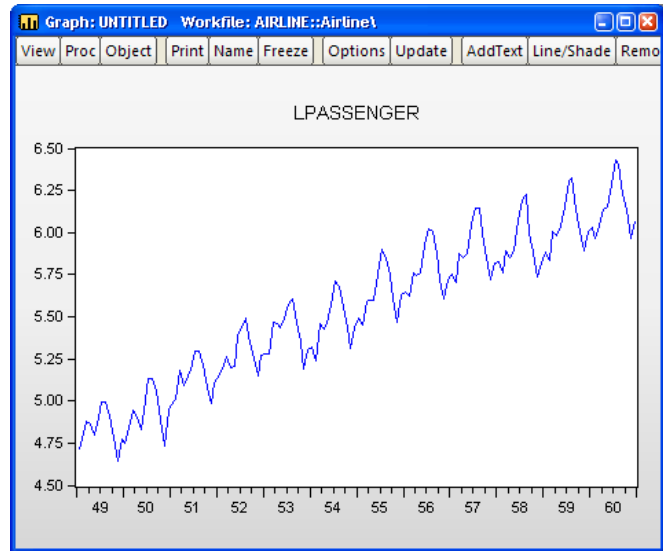
Don't worry if you're not sure which choice to make for any of these options. You can always make changes after the graph has been created from the main **Graph Options** dialog, under the **Graph Updating** section ("Graph Updating" on page 782). Even if you've chosen to create a non-updating graph, you can always switch the graph to update from this dialog, provided the original series or group still exists in the workfile.

Now, let's talk about the graph objects we've been creating.

Here, the titlebar shows that we have an untitled graph object. Notice also that since we are working with a graph object, the menu bar provides access to a new set of views and procedures which allow you to further modify the contents of the graph object.
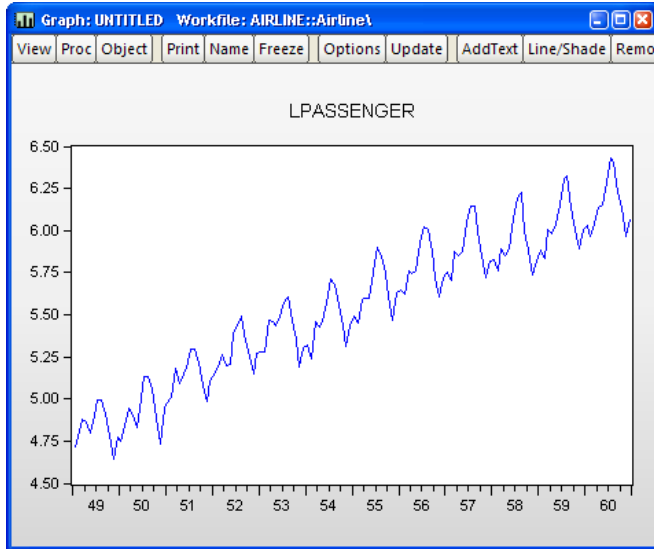
As with other EViews objects, the UNTITLED graph will not be saved with the workfile. If you wish to store the frozen graph object in your workfile, you must name the graph object; press the **Name** button and provide a name.

There's one other thing to notice in the titlebar. A graph with updating turned off will have a green icon in the upper left-hand corner, while a graph with updating turned on will have an orange icon. These colored icons will also be seen in the workfile window, once the graph has been named. You may also verify (or modify) the updating status using the **Graph Updating** section of the graph options dialog ("Graph Updating" on page 782).

Combining Graphs

You may also create a graph object by combining two or more existing named graph objects. Simply select all of the desired graphs and then double click on any one of the highlighted names. EViews will create a new, untitled graph, containing all of the selected graphs. An alternative method of combining graphs is to select **Quick/Show…** and enter the names of the graphs.

Note that combined graphs may not be made auto-updating.

Customizing Graphs

EViews allows you to perform extensive customization of your graph object. You may add text, lines and shades, edit or remove existing elements such as legends or titles, or change a wide variety of display settings for the graph.

A graph object is made up of a number of elements: the plot area, the axes, the graph legend, and possibly one or more pieces of added text or shading. To select one of these elements for editing, simply click in the area associated with it. A blue box will appear around the selected element. Once you have made your selection, you can click and drag to move the element around the graph, or double click to bring up a dialog of options associated with the element.

Alternatively, you may use the toolbar or the right mouse button menus to customize your graph. For example, clicking on the graph and then pressing the right mouse button brings up a menu containing entries for customizing, copying and saving the graph.

C_opy to clipboard...	Ctrl+C
_Options...	
Add _text...	
Add _lines & shading...	
_Templates...	
_Sort...	
_Remove selected	
_Save graph to disk...	
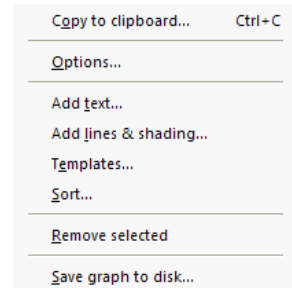
Adding and Editing Text

You may customize a graph by adding one or more lines of text anywhere in the graph. Adding text can be useful for labeling a particular observation or period, or for adding titles or remarks to the graph. To add new text, simply click on the **AddText** button in the graph object toolbar or select **Proc/Add text…** from the main graph menu.

To modify an existing text object, simply double click on the object. The **Text Labels** dialog will be displayed.

Enter the text you wish to display in the large edit field. Spacing and capitalization (upper and lower case letters) will be preserved. If you want to enter more than one line, press the **Enter** key after each line.

- The **Justification** options determine how multiple lines will be aligned relative to each other.

- **Font** allows you to select a font and font characteristics for the text.

- **Text in Box** encloses the text in a box.

- **Box fill color** controls the color of the area inside the text box.

- **Frame color** controls the color of the frame of the text box.
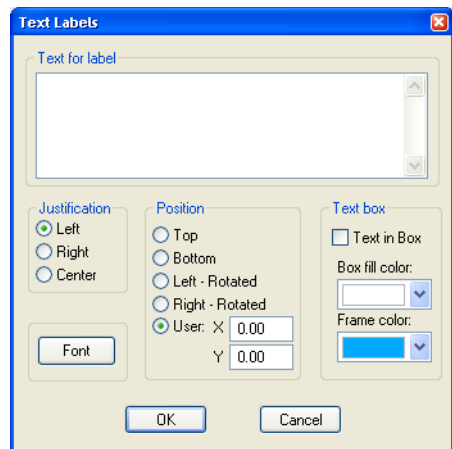
The first four options in **Position** place the text at the indicated (relative) position outside the graph. You can also place the text by specifying its coordinates. Coordinates are set in virtual inches, with the origin at the upper left-hand corner of the graph.

The X-axis position increases as you move to the right of the origin, while the Y-axis increases as you move down from the origin. The default sizes, which are expressed in virtual inches, are taken from the global options, with the exception of scatter diagrams, which always default to 3×3 virtual inches.

Consider, for example, a graph with a size of 4×3 virtual inches. For this graph, the X = 4, Y = 3 position refers to the lower right-hand corner of the graph. Labels will be placed with the upper left-hand corner of the enclosing box at the specified coordinate.

You can change the position of text added to the graph by selecting the text box and dragging it to the position you choose. After dragging to the desired position, you may double click on the text to bring up the **Text Labels** dialog to check the coordinates of that position or to make changes to the text. Note that if you specify the text position using coordinates, the *relative* position of the text may change when you change the graph frame size.

Adding Lines and Shades

You may draw lines or add a shaded area to the graph. From a graph object, click on the **Lines/ Shade** button in the toolbar or select **Proc/Add shading…**. The **Lines & Shading** dialog will appear.

Select whether you want to draw a line or add a shaded area, and enter the appropriate information to position the line or shaded area horizontally or vertically. EViews will prompt you to position the line or shaded area by providing an observation or data value.

You should also use this dialog to choose a line pattern, width, and color for the line or shaded area, using the drop down menus.
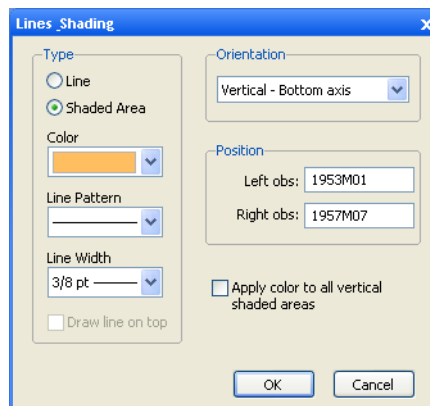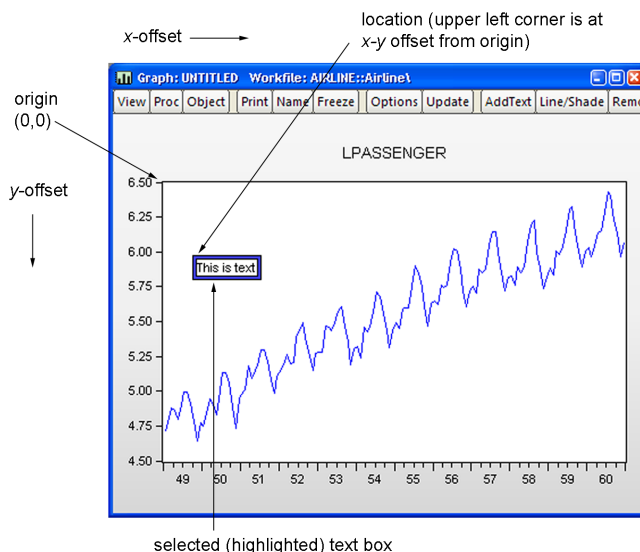
When adding a line, you may check **Draw line on top** to instruct EViews to draw the line after drawing the data. This can be useful with filled in areas, such as area or bar graphs.

If you check the **Apply color...** checkbox, EViews will update all of the existing lines or shades of the specified type in the graph.

Here we have drawn a vertical shaded area defined by the dates 1953M01 and 1957M07:
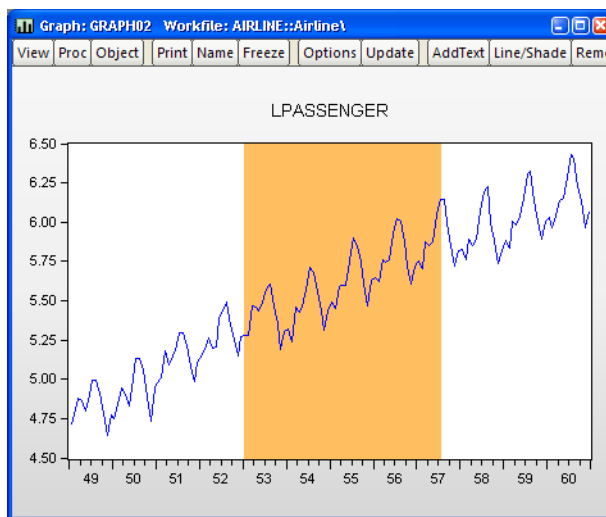
To modify a single existing line or shaded area, simply double click on it to bring up the dialog.

Drawing Lines and Arrows

It's may be useful to accentuate a data point in a graph or draw a comparison between two points. EViews allows you to embellish your graph by drawing custom straight lines anywhere in a graph object window. You may also choose from multiple designs for the arrowheads, including none (plain line), filled arrow, and open arrow.

To add an arrow or line to a frozen graph, place the cursor over the graph, right-click on the mouse, and select **Draw Arrow**. An "X" will appear over the graph. Place the "X" over the starting point for your line, left click the mouse to lock in this point, continue to hold down the mouse button, and move the cursor to the line's end-point and release the mouse button.

You can change either end-point by placing the cursor over the end, at which point the cursor will change to a white double-ended arrow, then left click the mouse and the cursor will change to a "X", continue to hold down the mouse key and move the cursor to the line's new end-point and release the mouse button.

If you select the middle of the arrow the "X" will appear, left-click the mouse and the arrow is selected allowing you to move it without changing the angle and only the position of the line.
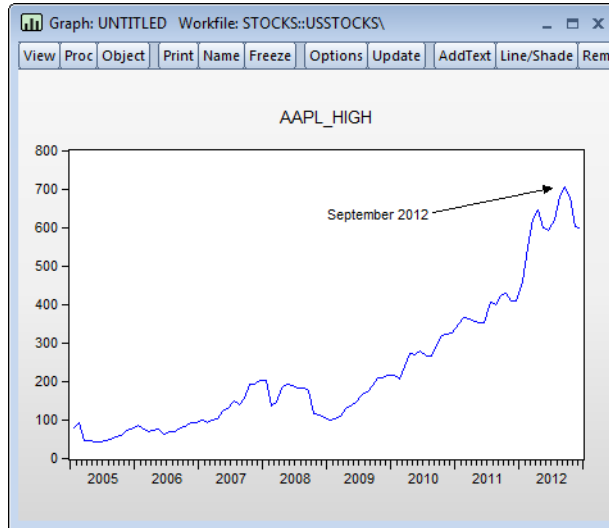
The line appearance can be modified in several ways. If you double-click anywhere on the line, the **Draw Object** menu will open, allowing you to choose your starting and ending types. You can also change the color, pattern, width, or arrow appearance. There is also an option that allows you to apply all these settings globally to add continuity to your project in one step.
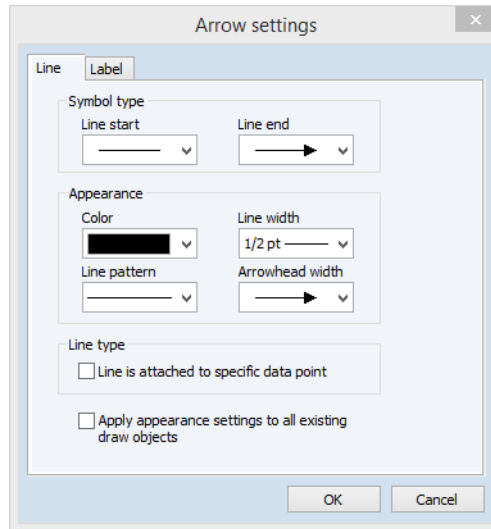
The undo command is disabled for any changes you make to the graph. To remove an arrow, click on it, and select **Delete**.

The **Line type** checkbox allows you to choose whether to anchor in virtual inches relative to the top-left corner of the graph, or whether to anchor the arrow to a data point. See See "Data-based Anchoring," beginning on page 763 for additional discussion.

Drawing Rectangles and Ellipses

You may also draw rectangles and ellipses in frozen graph objects.
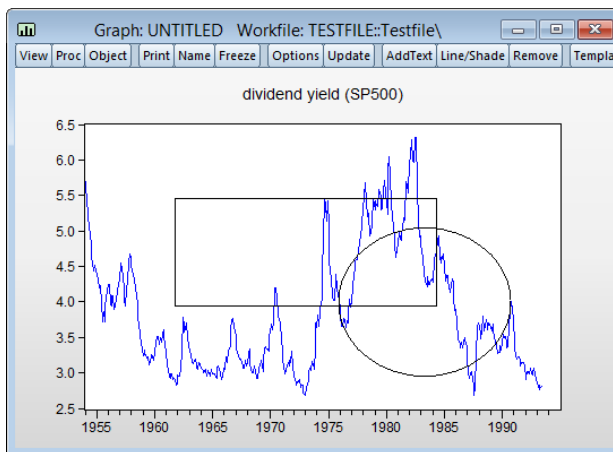
When the graph object is active, right-mouse click, and select **Draw/Draw Ellipse** or **Draw/Draw Rectangle**. The EViews cursor will change to a cross-hair. Click to set a location and then drag to form an object of the desired size.

Once created, you can move and resize the shape interactively. To move, hover over the shape to display the dragging hand cursor. Click and drag to the desired position.

To resize, click once to select the shape and display the positioning nodes. The cursor will change to a double arrow. Drag the node to a new position and release the cursor to resize the shape.

The position, anchor type, size, appearance and angle of the shape may be controlled via dialog. Double-click on the shape to display the dialog:
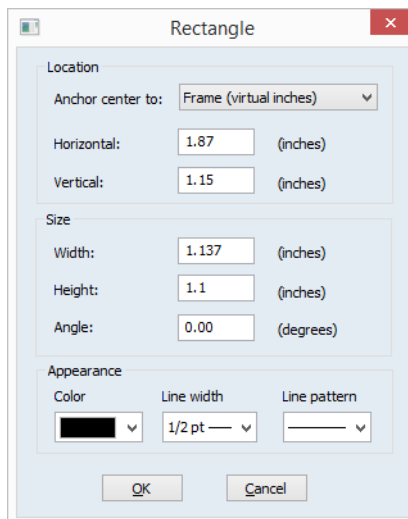
Here, we see the rectangle dialog which allows you to change the location, size, angle, color, width, and pattern of the rectangle.

The top dropdown allows you to choose whether to anchor the rectangle to the **Frame** (absolute position in virtual inches relative to the top-left corner of the graph), or whether to anchor the rectangle to **Data** (position in data and observation units in the graph). See See "Data-based Anchoring," beginning on page 763 for additional discussion.

Data-based Anchoring

By default, the positions of arrow and text objects in a graph are specified in virtual inches relative the top left corner of the graph. While useful for some types of drawing, this approach is problematic when one wishes to place objects at particular data points or dates in the graph as in, for example, identifying an outlier or the observation value at a notable date. For one, placing the object is quite difficult programmatically as there is no clear way of translating from data values or observations to virtual inch offset. Moreover, if one changes the axes ranges, sample, or aspect ratio of the graph, the absolute positioning method breaks down.

You can attach arrows and the new rectangle and ellipse drawing objects to fixed data/date points within the graph. You can, for example, choose to place your object at the point defined by the observation for 1991q1 and the maximum value of X.

Similarly, sizes of objects may also be expressed in terms of observations and data units, so that you can draw a rectangle that is "2 years wide" and "10.5" data units tall.
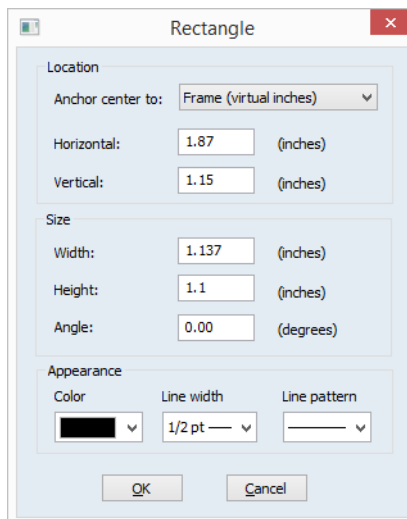
In addition to being easier to specify programmatically, if, you then alter the graph frame or axes, the object will move with the data in the graph.

By default, when you place an arrow, rectangle, or ellipse in a graph, the object placement will use the standard approach of measuring virtual inches relative to the top left corner of the graph.
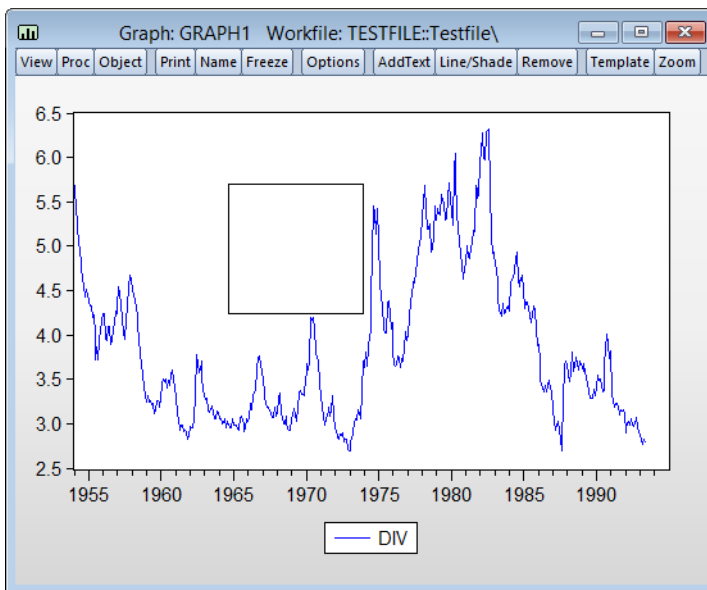
Here we have drawn a rectangle on a graph view of a series.
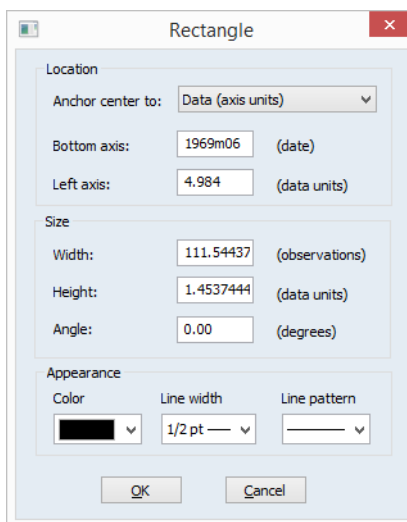
To modify the anchor point, resize the rectangle, or otherwise change its appearance (size, angle, color, width, and pattern), double click on the shape to bring up the dialog:

You may use the **Anchor to** dropdown to choose between the absolute position **Frame (virtual inches)** method, and the **Data (axis units)** approach. Select **Data (axis units).**

Notice that the units labels for the axes and sizes change and that EViews translates the frame units into observation and data units. We'll leave the position as is and click on **OK** to accept the changes.

Next, double click on the graph to display the **Graph Options** dialog, and select the F**rame & Size/Size & Indents** node and change the aspect ratio to 1:3 by entering "3" in the **Auto aspect ratio/Default** edit field. Click on **OK**.
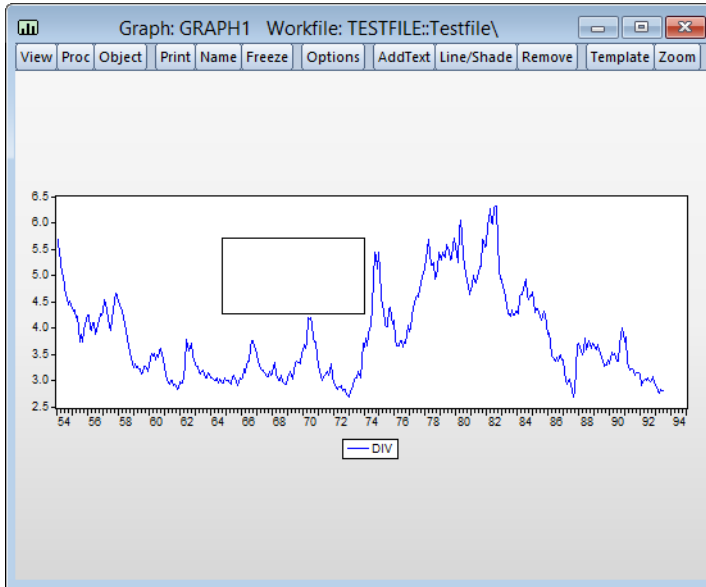
(Note: text drawing objects currently do not support the placement using data-based coordinates but you produce a similar effect by creating a data-based graph arrow with label, and making the label of minimal length.)

Sorting Graphs

Selecting **Sort...** from the **Proc** menu or the right mouse-button menu from the brings up the **Sort Order** dialog. Providing one or more sort keys will reorder the observations in the graph on the basis of the values of the keys. You may choose to reorder the data in ascending or descending values of the keys.

Note that sorting reorders the data in the graph object, not the underlying data in the original series or matrices.

Sorting the graph in ascending order yields:
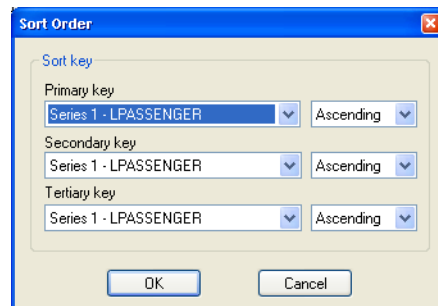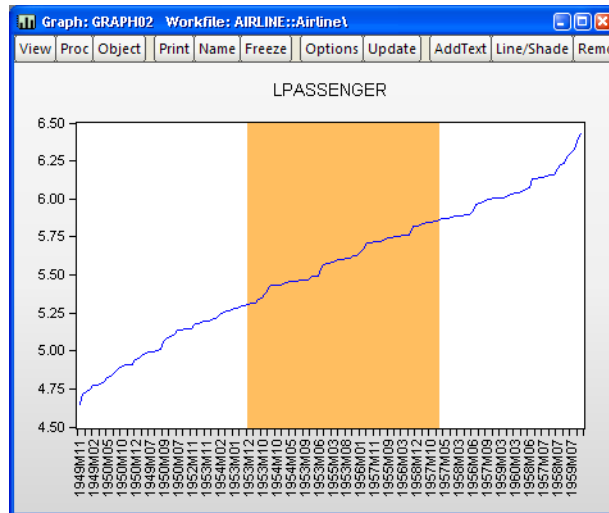
Notice that EViews displays as much axis label information as possible since the observations are no longer described using a single time scale. Note also that the existing shade is associated with observation numbers and remains in the sorted graph, albeit with a very different interpretation.

Removing Graph Elements

To remove a graph element, simply select the element and press the **Delete** key. Alternately, you may select the element and then press the **Remove** button on the graph toolbar. For example, to remove the vertical axis in your graph simply click on the axis. A border will appear around the axis indicating that it is selected.
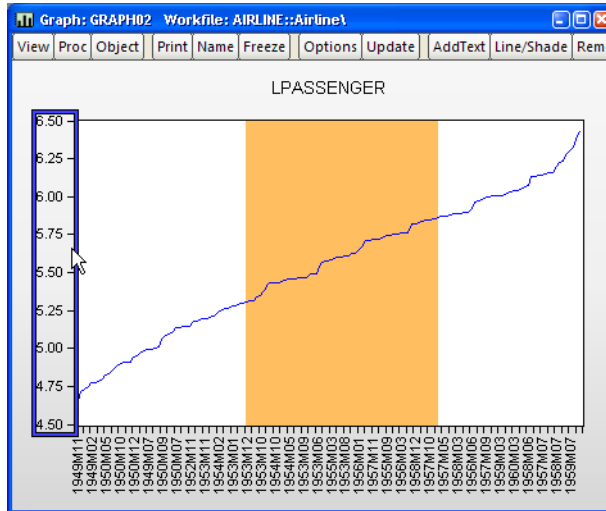
Simply press **Delete** or click on the **Remove** button to delete the scale.

(Double clicking will open the **Graph Options** dialog and will show the settings for the vertical axis.)

You may also remove legends, as well as any text, lines or shading which have been added to the graph.

Graph Options

The main **Graph Options** dialog controls the basic display characteristics of your graph. The dialog may be opened by selecting **Options...** from the right mouse menu. You may also double click anywhere in the graph window to bring up the **Graph Options** dialog. If you double-click on an applicable graph element (the legend, axes, *etc.*), the dialog will open to the appropriate page.
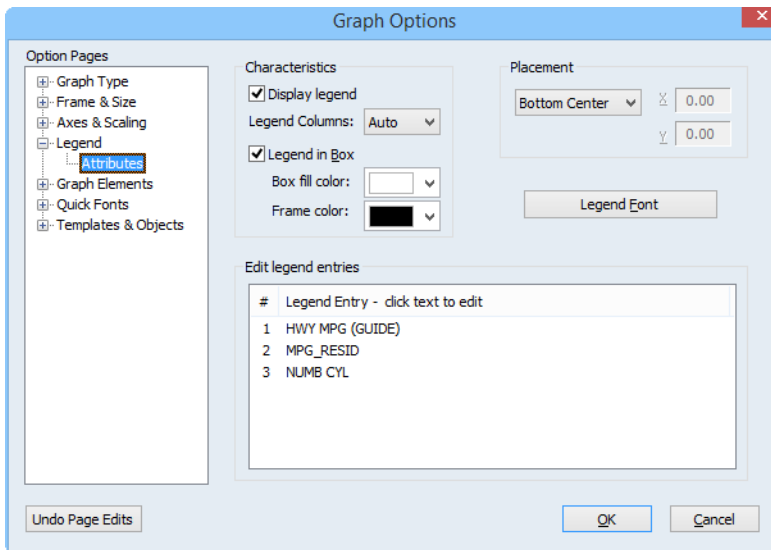
Here we have double clicked on the graph's legend, so the dialog opens to the **Legend** section. First, note the tree structure on the left-hand side of the dialog. EViews offers a lot of graph options, and to make them more accessible, they have been broken up and organized into groups. Within each group, you will find specific sets of options. Let's go through these groups one at a time.

Graph Types

The **Graph Type** section allows you to change the graph type. Begin by selecting a type from the **Basic type** page:
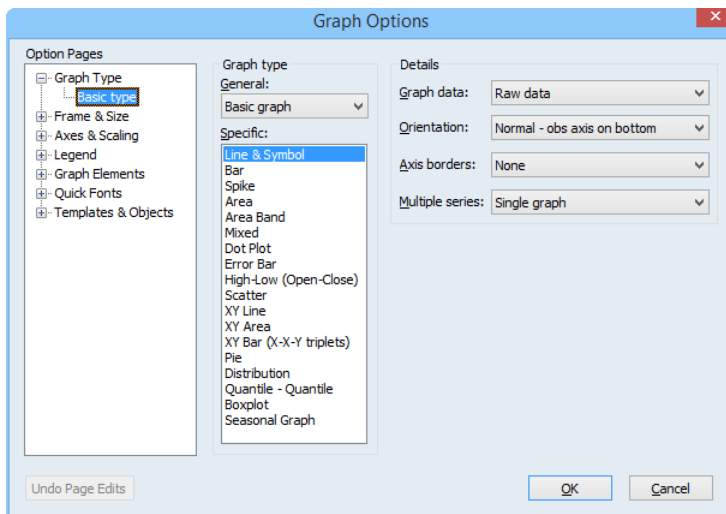
The listbox on the left-hand side of the page provides access to the fundamental graph types. The graph types that are available depend on whether the graph uses data from a single series (or column of data, *e.g.*, a vector) or more than one series (or more than one column of a matrix). For example, the **Area Band**, **Mixed with Lines**, and **High-Low (Open-Close)**, **Scatter**, **XY Line**, and **XY Area** types are only available for graphs containing multiple series or matrix columns.

Depending on the nature of your graph, there are a number of additional settings that may be displayed on the right-hand side of the **Basic type** page:

- **Multiple series** – When plotting line, bar, or area graphs with multiple series, EViews displays an option for producing a stacked graph. Simply select **Stack lines, bars, or areas** to display a stacked graph (see "Single Series Graphs," beginning on page 632 for details).

- **XY series handling** – In cases where there is potential ambiguity concerning the handling of multiple series in XY graphs (Scatter, XY Line, XY Area, XY Bar), EViews will display a dropdown menu prompting you for whether you want to plot the data using **First vs. All** or using **XY pairs** (see "Pairwise Graphs" on page 636 for discussion).

- **Mixed** - **First series type** – The mixed graph displays multiple series in a single graph frame, with the first series shown as a bar, spike, or area graph, or with the first two series displayed as an area band graph, with the remaining series depicted using lines.

  If you select **Mixed with Lines** as your graph type, the dialog will change to offer you a choice for the graph type for the first series type. The default setting is **Bar**. See "Mixed" on page 673.

- **Sample Breaks & NA Handling** – If your data involve sample breaks or missing values, EViews will display additional settings allowing you to control the appearance of your graph. See "Sample Break & NA Handling" on page 626 for further discussion.

Categorical graphs

If you have selected **Categorical graph** from the **General** drop-down on the **Basic type** page, a **Categorical options** page will also be available under the **Graph Type** section:
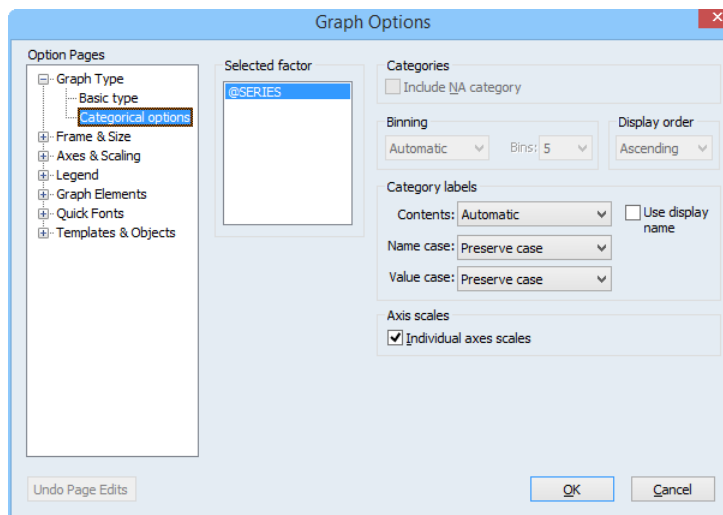
For a discussion of categorical graphs and use of the **Categorical options** page, see Chapter 14. "Categorical Graphs," on page 717.

Frame & Size

The **Frame & Size** section controls basic display characteristics of the graph, including color usage, framing style, indent position, and grid lines.
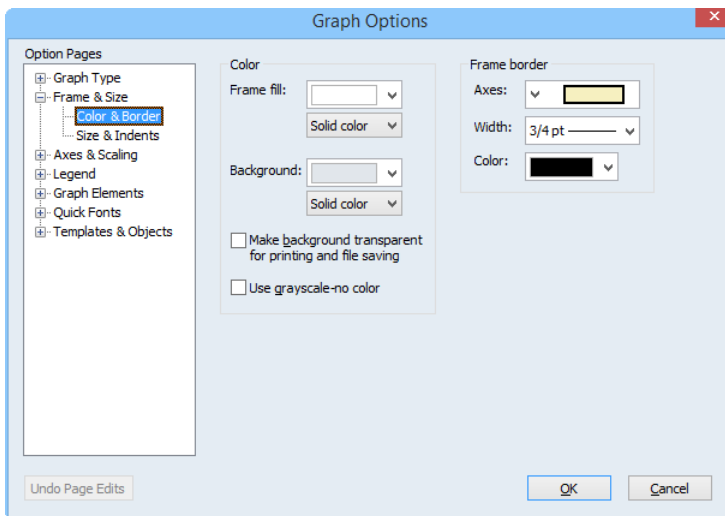
For discussion of each of these settings, see "Frame" on page 649.

Axes and Scales

To change or edit axes, select the appropriate page from the **Axes & Scaling** group. The first two sections in this group, **Data scaling** and **Data axis labels**, relate to settings for the data scales. The third section, **Obs/Date axis**, sets options for the workfile (observation) scales. The fourth section, **Grid Lines**, can be used to add or modify grid lines on any axis.

Depending on its type, a graph can have up to four axes: left, bottom, right, and top. Each series is assigned an axis as displayed in the upper right listbox on the **Data scaling** page:
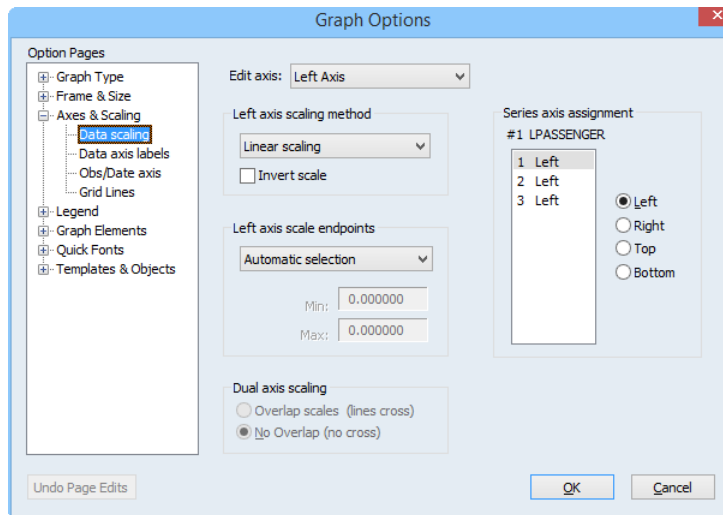
You may change the assigned axis by first highlighting the series and then clicking on one of the available axis buttons. For example, to plot several series with a common scale, you should assign all series to the same axis. To plot two series with a dual left-right scale, assign the two series to different axes. To edit characteristics of an axis, select the desired axis from the drop down menu at the top of the dialog.

Adding Custom Labels

EViews offers one of two methods for add custom labels depending on the type or status of the graph and the workfile structure.

In the **Obs/Date axis dialog**, you will see *either* a button or an edit field at the bottom of the **Axis labels** group.
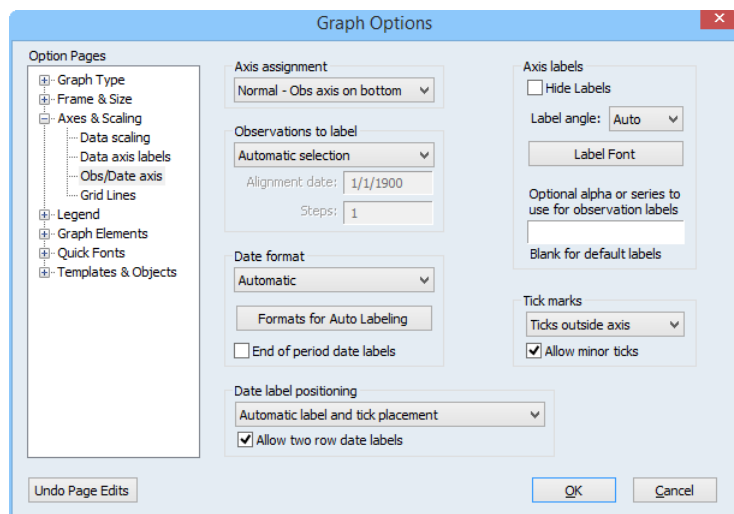
For graphs that are actively updated, such as a graph view or a frozen graph with updating turned on, an edit field will be displayed. Entering an alpha or series name in the field instructs EViews to label each observation with the text or value of the alpha or series.

Optional alpha or series to use for observation labels

Blank for default labels

The labels are updated as the values of the alpha or series change. To remove the labels and return to EViews default labels, leave this field blank.
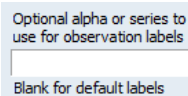
The second approach to adding custom labels allows you to use alpha or numeric series values in a static fashion. The labels are initialized with the current values in the series, but will not be updated if the underlying data change. You may also choose to initialize labels of this type with observation numbers, or simply leave them blank and assign your own text. The labels may be edited as you please, and your changes will not be replaced when the graph is refreshed or the type is changed.

This static method of labeling is used for a graph whose data is no longer changing, such as a frozen graph with updating off. If stored labels can be used, you will see the **Add custom obs labels** or **Edit custom obs labels** button in place of the edit field.

Add custom obs labels

To initialize the labels, click the **Add custom obs labels** button to bring up the **Axis label initialize** dialog. Select either **Standard observations labels (dates)**, **Observation numbers** (taken from the workfile), **Uninitialized (empty strings)**, or **Series in workfile**, providing an alpha or series name if you chose the latter. Click on **OK** to accept your selection.

A dialog will open allowing you to edit the label associated with each observation. Here we have initialized the custom labels with the workfile labels.

Axis label initialize

Graph currently uses default axis labels.

Initialize custom labels with:

◉ Standard observation labels (dates)
◯ Observation numbers
◯ Uninitialized (empty strings)
◯ Series in workfile: series,alpha

OK   Cancel
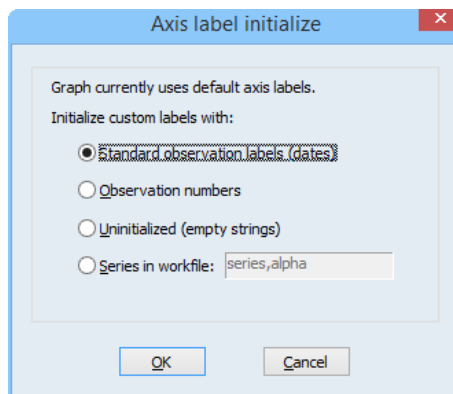
If you wish to return to the default axis labels, click the **Remove Labels** button in the **Edit custom obs labels** dialog. All custom labels will be deleted and replaced with the default labels.

EViews does its best to preserve labels you have assigned to a Graph. For example, say you have created a graph view and assigned labels based on an alpha series in the workfile. When you're ready to turn off updating and obtain a snapshot of the current graph, EViews will hold on to the labels from the alpha series. you may now move the graph to a different workfile page or delete the alpha series completely, and the graph labels will be preserved.

See for additional detail.

Legend

To edit the graph legend characteristics, select the **Attributes** page of the **Legend** group.

You may change the basic characteristics of the legend (number of columns, enclose it in a box with specified fill and frame color, change the font, and change the placement) using the settings on the top half of the **Attributes** page.

Use the bottom half of the dialog to change the legend text. Simply click on the specified item in the list box, and then click in the edit field and alter the text.
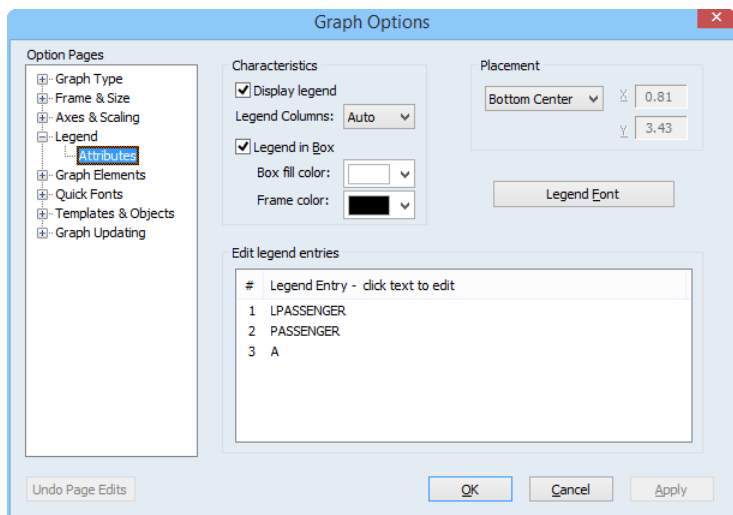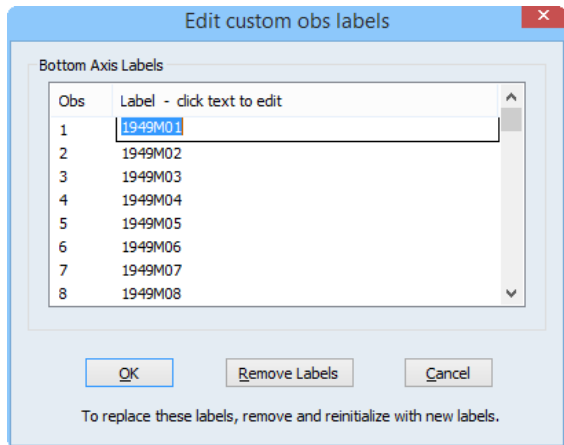
Note that if you place the legend using user-specified (absolute) positions, the relative position of the legend may change if you change the graph frame size.

Lines and Symbols

The **Lines & Symbols** page under the **Graph Elements** group provides you with control over the drawing of all lines and symbols corresponding to the data in your graph.

The current line and symbol settings will be displayed in the listbox on the right-hand side of the dialog. You may choose to display lines, symbols, or both, and you can customize the color, width, pattern, and symbol usage. Once you make your choices, click on **Apply** to see the effect of the new settings.
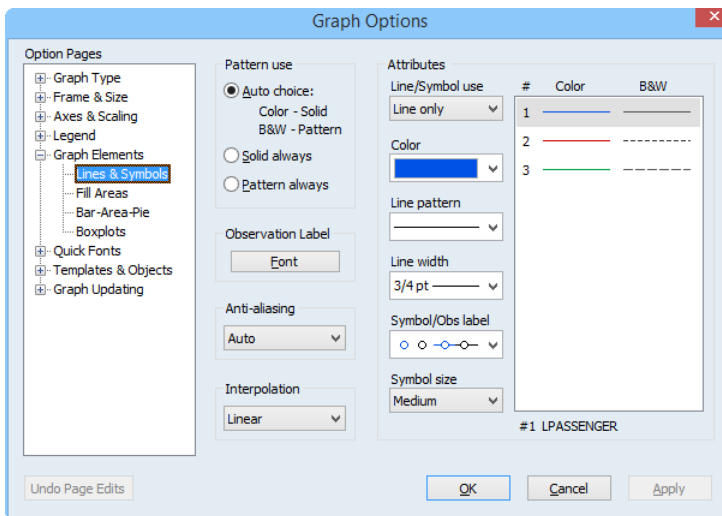
See "Lines and Symbols" on page 665 for additional detail.

Fill Areas

The **Fill Areas** page under the **Graph Elements** group allows you to control the display characteristics of your area, bar, or pie graph. Here, you may customize the color, shading, and hatching pattern of the graph elements.
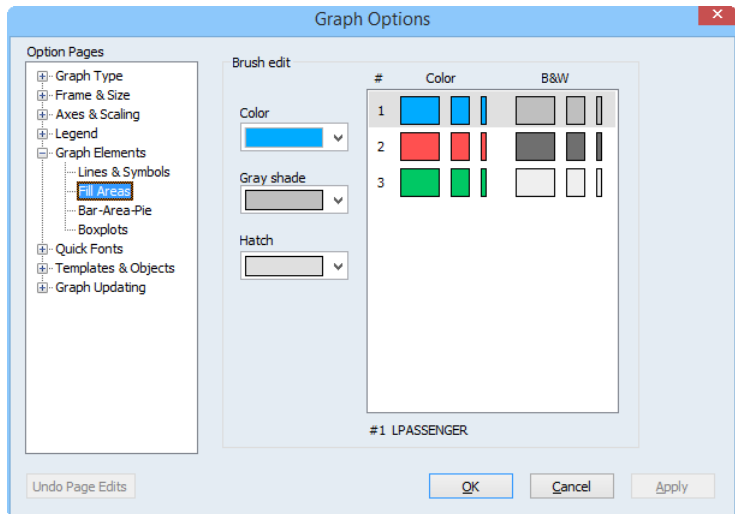
Bar-Area-Pie

The **Bar-Area-Pie** page under the **Graph Elements** group provides additional options for bar, area, or pie graphs. You may add data labels to bar graphs, as well as modify the bars visually by adding shading or a 3D appearance. You may also specify if fill areas should be outlined.
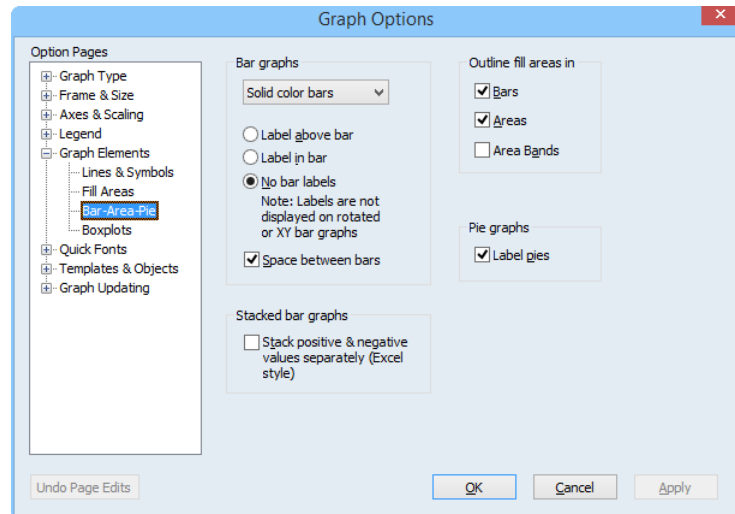
"Fill Areas" on page 667 provides additional discussion.

Boxplots

The final page of the **Graph Elements** group, **Boxplots**, provides control over all elements of a boxplot. As with the line and fill area dialogs, these options are only applicable if you have chosen boxplot as your graph type.
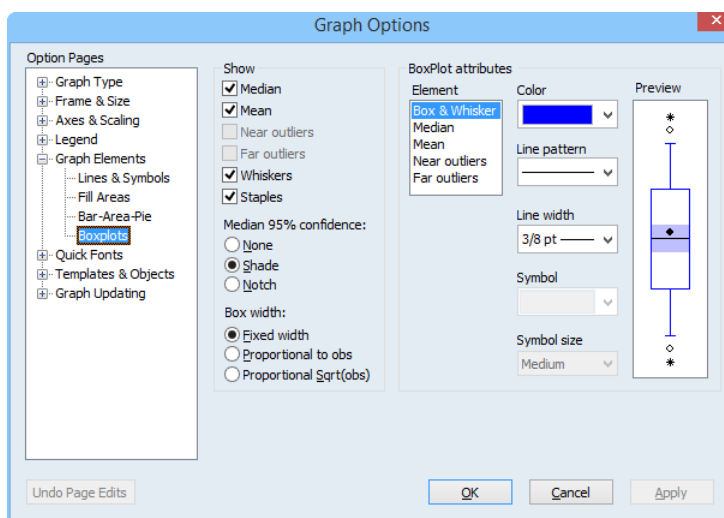
The left-hand side of the dialog allows you to select which boxplot elements should be visible, and to specify which method should be used to display the confidence interval. On the right-hand side of the dialog, you may modify the characteristics of each boxplot element individually. Each element may have a unique color, symbol, line pattern, or width.

See "Boxplot" on page 701 for a discussion of boxplots.

Quick Fonts

The **Quick Fonts** page lets you specify multiple fonts simultaneously, overriding the settings on other pages.

Note that changes to text objects cannot be undone after you press the **Apply** button.

Templates

Having put a lot of effort into getting a graph to look just the way you want it, you may want to use the same options in another graph. EViews allows you to use any named graph as a *template* for a new or existing graph. You may think of a template as a graph style that can be applied to other graphs.

In addition, EViews provides a set of predefined templates that you may use to customize the graph. These predefined templates are not associated with objects in the workfile, and are always available. The EViews templates provide easy-to-use examples of graph customization that may be applied to any graph. You may also find it useful to use the predefined templates as a foundation for your own graph template creation.

To update a graph using a template, double click on the graph area to display the **Graph Options** dialog, and click on **Apply template** under the **Templates & Objects** group. Alternatively, you may right mouse click, and select **Templates...** to open the desired page of the dialog.
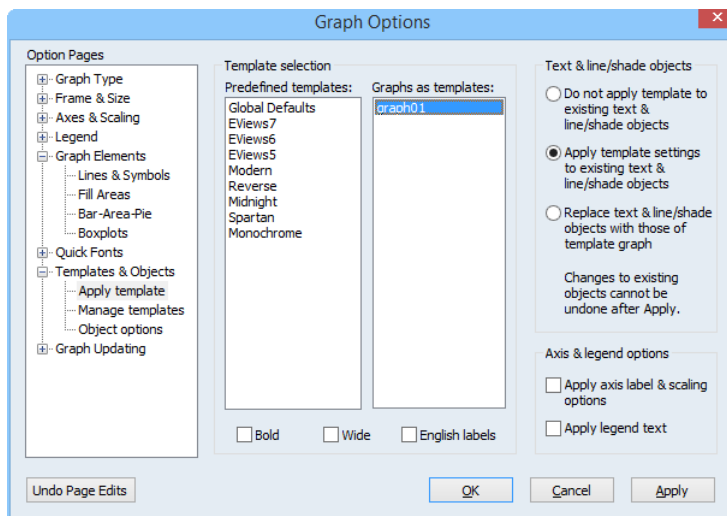
On the left-hand side of the dialog you will first select your template. The left-hand list box contains a list of the EViews predefined templates. The right-hand box contains a list of all of the named graphs in the current workfile page. In this dialog, we have selected the graph object GRAPH01 for use as our graph template.

If you select one of the templates, you will be given the choice of applying the **Bold**, **Wide**, or **English labels** modifiers to the base template. As the name suggests, the **Bold** modifier changes the settings in the template so that lines and symbols are bolder (thicker, and larger) and adjusts other characteristics of the graph, such as the frame, to match. The **Wide** modifier changes the aspect ratio of the graph so that the horizontal to vertical ratio is increased.

The **English labels** modifier changes the base template's settings for auto labeling the date axis. When **English labels** are applied, labels using month formatting will default to English month names. For example, with **English labels** turned on, the axis may read: "Jan", "Feb", "Mar", while otherwise it may read: "M1", "M2", "M3".

Applying the **Midnight** template to the example graph yields:
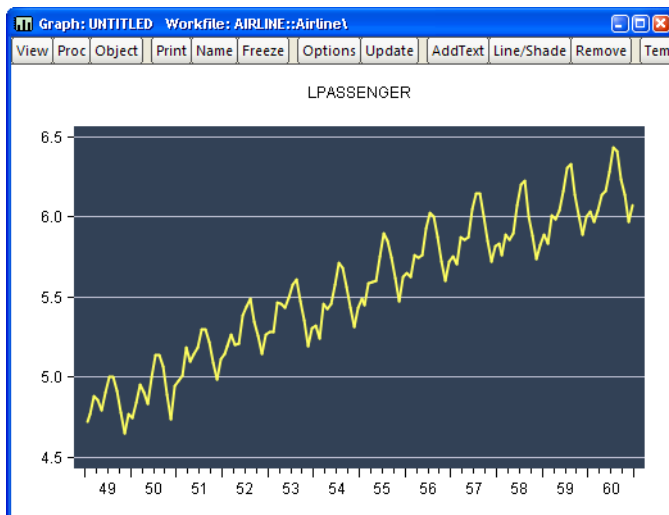
Applying **Midnight** with the **Bold** and **Wide** modifiers selected yields:
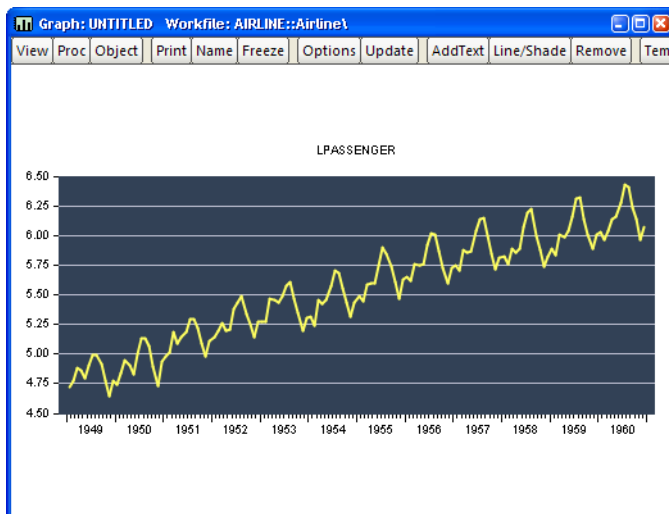
You may reset the dialog by clicking on the **Undo Page Edits** button prior to clicking on **OK**.

When you click on the **OK** or **Apply** button, EViews will immediately update all of the basic graph settings described in "Graph Options" on page 768, including graph size and aspect ratio, frame color and width, graph background color, grid line options, and line, symbol, and filled area settings.

In contrast to the basic graph settings which are always updated when you click on **OK** or **Apply**, the effects of using the template on the characteristics of existing text, line, and shade objects in the graph is controlled by the choices on the right-hand side of the dialog. There are three possibilities:

- **Do not apply template to existing text & line/shade objects** – instructs EViews to use the text, line, and shade attributes in the template or template graph only for the purpose of updating the default settings in the graph. If you select this option and select **Apply**, *subsequently* added text, line, and shades will use the updated settings, but existing objects will retain their existing characteristics.

- **Apply template settings to existing text & line/shade objects** – will update both the settings for existing text, line, and shade objects, and the defaults used for newly added objects.

- **Replace text & line/shade objects with those of the template graph** – will first remove any added text label, line, or shading objects in the existing graph, and then copy to the graph any such objects in the template.

If you have a template graph that you wish to apply to graphs in other workfiles, you may add it to the global template list so it will be accessible from any workfile. Click on **Manage templates** under the **Templates & Objects** group. The listbox in the top middle of the dialog shows all existing global templates, including the EViews predefined templates. These templates are stored globally and may be accessed from any workfile. The listbox on the bottom of the dialog shows all named graphs in the current workfile page that have not already been added to the predefined template list.
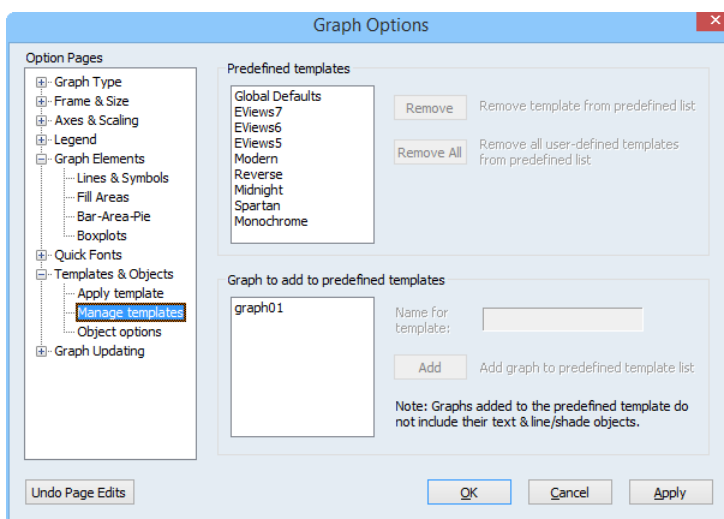
To add a graph to the predefined template list, select it from the listbox on the bottom and click the **Add** button. The graph will then move into the predefined list on the top of the dialog and will be removed from the list on the bottom. Before clicking the **Add** button, you

may wish to give the graph template a different name by editing its name in the edit field to the right of the listbox.

Similarly, to remove a graph from the predefined list, select it from the listbox on the top of the dialog and click the **Remove** button. This will remove the graph from the template list, but will not affect the graph object itself. Note that the EViews predefined templates cannot be removed.

Objects

Also under the **Templates & Objects** group, the **Object options** dialog allows you to control the default characteristics of *new* text, shade, or line drawing objects later added to the graph, or to update the characteristics of the existing objects.

You may select colors for the shade, line, box, or text box frame, as well as line patterns and widths, and text fonts and font characteristics.

By default, when you apply these changes to the graph object options, EViews will update the default settings in the graph, and will

use these settings when creating *new* line, shade, or text objects. Any existing lines, shades or text in the graph will not be updated. If you wish to modify the existing objects to use the new settings, you must check the **Apply to existing line/shade objects** and **Apply to existing text objects** boxes prior to clicking on the **OK** button.

Note that you may change the default settings for any of these options by selecting **Options/Graphics Defaults...** from the main EViews menu. Any new graph views or objects will use the updated options as the default settings.

See "Adding and Editing Text" on page 757 and "Adding Lines and Shades" on page 758.

Graph Updating

For graphs which may still be tied into underlying data, you will be offered the opportunity to specify auto updating settings. The **Graph Updating** section allows you to change options
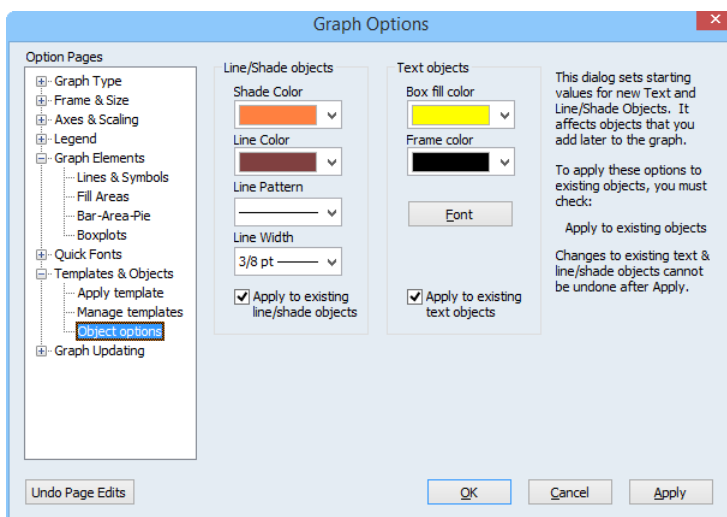
for how and when the graph is updated when its underlying data changes or the workfile sample is modified.
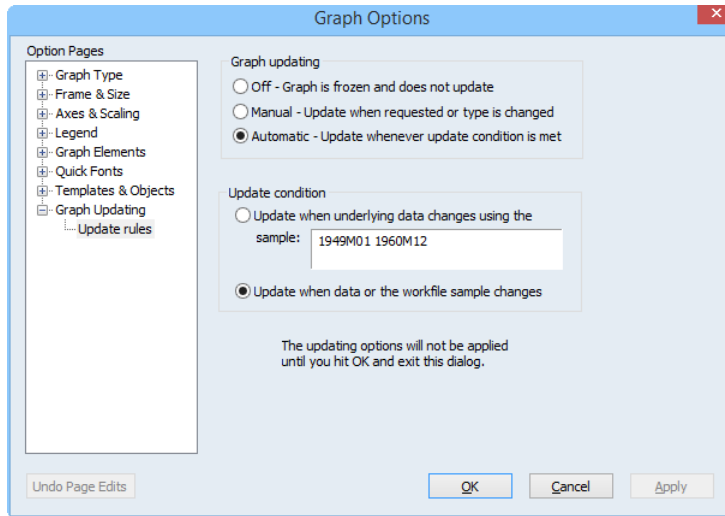
See *"Freezing a Graph View" on page 752* for details.

Modifying Multiple Graphs

Some views are made up of multiple graphs. Like single graph views, these multiple graph views may be turned into graph objects by freezing. For example, the impulse response view of a VAR can display multiple graphs in a single view.

You may also create a graph object containing multiple graphs by combining existing named graph objects. Simply select the desired graphs and then double click on any one of the highlighted names. An alternative method of combining graphs is to select **Quick/Show…** and enter the names of the graphs.

There are two ways to work with a multiple graph. You may change the settings for the multiple graph as a whole, or you may work with an individual graph component of the multiple graph.

Working with Multiple Graphs

EViews makes it easy to work with all of the graphs in a multiple graph. Simply select **Proc** from the graph menu or click on the background of the graph and display the right mouse-button menu. EViews will display a menu prompting you for additional choices.

C̲opy to clipboard...	Ctrl+C
O̲ptions on all graphs...	
Position and align graphs...	
A̲dd lines & shading to all graphs...	
Add t̲ext...	
T̲emplates...	
S̲ave graph to disk...	

Note that selecting **Proc** will display a menu with additional options related to auto updating of graphs. See "Freezing a Graph View" on page 752 for a discussion of graph updating.

These menu items set options that apply to all graphs in the graph object.

- **Copy to clipboard...** copies the graph to the clipboard.

- To set a common graph attribute to all graphs, select **Options on all graphs…**. EViews will display the multiple graph version of the **Graph Options** dialog, with each page initialized using the settings for the first of the multiple graphs. Here we see the dialog open to the **Color & Border** page of the **Frame & Size** section:
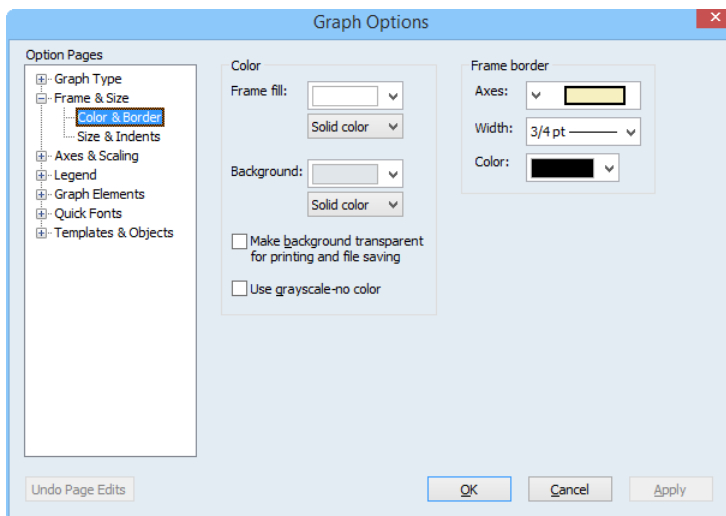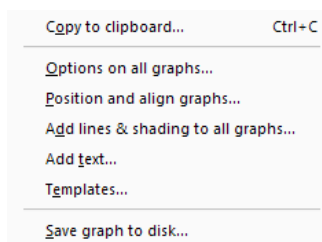
Click on **OK** to accept and apply the changes to each of the graphs.

- Each single graph in a multiple graph may be positioned freely by dragging the graph. Alternately, you may wish to align graphs in columns and control the overall spacing between graphs; for quick positioning all of your graphs, select **Position and align graphs...** to open the graph alignment dialog.

You may choose the number of columns and blocksize, as well as the horizontal and vertical spacing around individual graphs.

Here, we instruct EViews to display the graphs in blocks of 5, positioned using 3 columns per row. In this example, the first row of a block will contain 3 columns while the second row will have 2 columns; the pattern is repeated as necessary. If the specified blocksize is less than the number of columns, the "effective" blocksize will be the smallest multiple of the blocksize greater than the number of columns. Specified 3 columns with a blocksize of 2 is equivalent to 3 columns with a blocksize of 4.

Note also that when the dialog opens, the current spacing settings will be specified in the manual alignment edit fields. To have EViews automatically reposition the graphs (which may be useful if, for example, you have changed aspect ratios or font settings), select **Automatic** and click on **OK**.
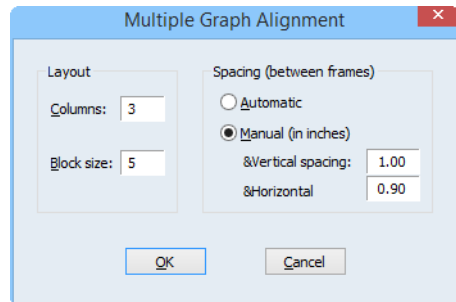
- If all of your graphs share a common axis, you can draw lines or add shading to each graph in the object, by selecting **Add lines & shading to all graphs…**.See "Adding Lines and Shades" on page 758.

- Selecting **Add text…** allows you to annotate your multiple graph. Note that adding an item to the multiple graph differs from adding it to an individual graph since it will not move as you move individual graphs within the multiple graph. See "Adding and Editing Text" on page 757.

- Selecting **Templates…** allows you to apply a template graph to each individual graph in your multiple graph or to reset the graph to use the global defaults. See "Templates" on page 778 for a discussion of templates.

- **Save graph to disk…** brings up the **File Save** dialog, as described in "Saving Graphs to a File" on page 787.

Working with Individual Graphs

You may change the options for a single graph within a multiple graph in the usual fashion by double clicking on the single graph to display its options dialog.

You can also perform various operations on individual graphs. Click on the individual graph and EViews will confirm the selection by surrounding the graph with a blue border. Select **Proc** or right mouse click to display a menu that combines the individual and multiple graph choices.

Most of the menu items are taken from the multiple graph menu, and apply to the entire graph. For example, selecting **Copy to clipboard...** allows you to copy the entire graph to the clipboard, not the individual graph.

The middle set of items provide tools for working with the selected graph. You may use these to change options, to add lines, shading, or text, draw rectangles or ellipses, to sort the data, or to extract or remove the selected graph.
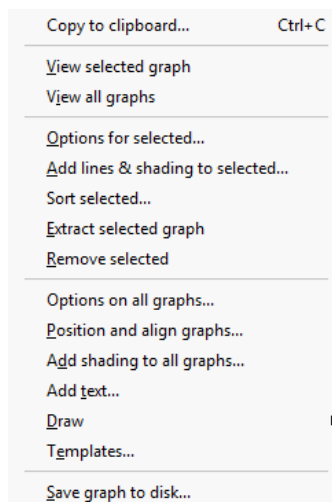
Printing Graphs

Clicking on the **Print** button on the graph view or graph object window toolbar will open the **Print** dialog, allowing you to override the various global settings for graph printing.

The top section of the **Print** dialog may be used to select a printer and print options, or to redirect the print job to an RTF file, graph object, or spool object (see "Print Setup" on page 880).

Most of the remaining options are self-explanatory. If you wish to print your graph in color using your color printer, make certain that the **Print in color** box is checked. Conversely, if you are printing to a black and white printer, you should make certain that this box is not checked so that EViews will substitute line patterns for colors.

Make background transparent should be used to ignore the background color of the graph (typically used when printing to a black-and-white printer).

See "Print Setup" on page 880 for additional details.
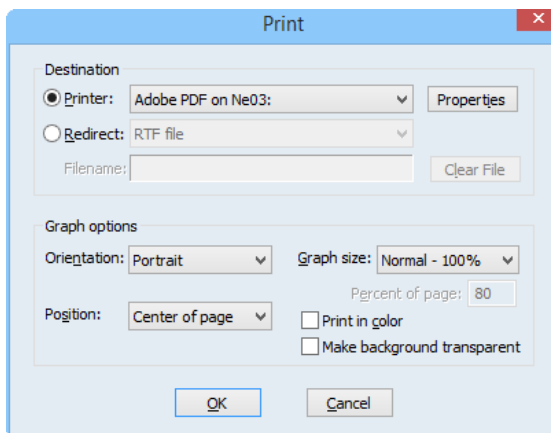
Saving Graphs to a File

EViews allows you to save your graphs to a file in a variety of popular graphics formats (Windows Metafile, PostScript, PDF, bitmap, GIF, JPEG, PNG) or to save the graph as a image file with a LaTeX wrapper.

Simply select **Proc** from the graph menu or click on the background of the graph and right-click to display the menu, then select **Save graph to disk...** to bring up the **Graphics File Save** dialog.

In the top portion of the dialog, you should provide the name of the file you wish to create. EViews will automatically append an extension of the proper type to the name (here, ".EPS" since we are saving an Encapsulated PostScript file).

Next, select the **File type**, and any options associated with the output type. You may select **Metafile - Win 3.1**, **Enhanced Metafile**, **Encapsulated Post-Script**, **Bitmap**, **Graphics Interchange Format** (also known as GIFs), **Joint Photographic Experts Group** (better known as JPEG), **Portable Network Graphics** (PNG), **Portable Document Format** (PDF), or **LaTeX**. You may elect to save the graph in color or not, whether to make the background transparent, and, for PostScript files, elect to include a bounding box or choose the graph orientation.

Lastly, you should select the **Output graph size**. The size may be specified in inches, centimeters, printer points, picas, or pixels. If the **Lock aspect ratio** checkbox is selected, changes to the **Width** or the **Height** will generate corresponding changes in the other dimension. If you wish to scale your graph in a non-proportionate fashion, you should uncheck this box.

When saving raster images (bitmap, GIF, JPEG, PNG) with sizes expressed in anything but pixels, EViews offers an additional setting, **Dots per Inch**, which specifies the output resolution. The total number of pixels written in the output file will depend on both the number of inches and the number of dots per inch. If, for example, you export a 5 by 4 inch GIF at 300 dots per inch, the final output file will be 1500 by 1200 pixels. At 100 dots per inch, the output file will be 500 by 400 pixels.
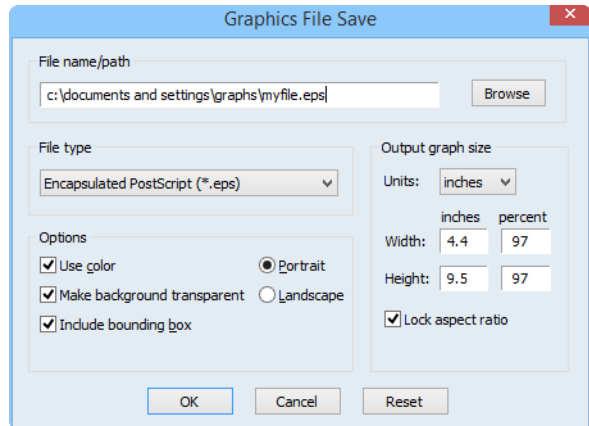
For LaTex output, you have the additional option to output full LaTeX document specification. Alternatively, you can disable the **Include document specification** option and the minimal amount of code will be generated for you to insert into your preexisting document.

The default graph file saving options may be set in the global options dialog by selecting **Options/Graphics Defaults....** and opening the **Exporting** page (see "Graphics Defaults" on page 878).

Graph Commands

For those of you who wish to automate these procedures, for example to produce a regular report, EViews allows you to perform extensive graph customization from the command line or using programs.

See Chapter 2. "Working with Graphs," on page 33 in the *Command and Programming Reference*, and "Graph" on page 236 in the *Object Reference*, for additional detail.

Chapter 16. Table and Text Objects

Table and text objects form the basis of presentation output for, and EViews provides sophisticated tools for customizing the appearance of these objects.

Table Objects

Freezing views that contain formatted text or numbers that are aligned in columns or rows produces table objects. The following sections describe the basics of working with table objects, including creating, formatting, printing, and exporting of tables.

Creating Tables

In EViews, a table may be an object view or a table object. Table views are object views that contain formatted text or numbers that are aligned in columns and rows. Examples of table views are the spreadsheet views of a series and the estimation output views of an equation. There are a limited set of customizations that are available for table views.

A table object is an independent object that contains formatted text or numbers. Table objects may be created directly, by issuing a table declaration, or indirectly, by freezing a table view. As with graph objects, table objects are "not live" in the sense that they do not reflect the current contents of the underlying object, but are based instead upon the contents of the object at the time the object was frozen. Table objects also allow for a full set of customizations.

While many of the features described here apply to both table views and table objects, the remainder of our discussion focuses on customization of table objects. Working with table views is described elsewhere (see, for example "Changing the Spreadsheet Display" on page 124).

Table Basics

The most basic operations in a table involve selecting cells and editing cell values.

Selecting Cells

Selecting one or more cells is one of the most common tasks in working with table views and table objects. For the most part, you will find that cell selection works as it does everywhere else in Windows, but a brief review may prove useful.

The simplest selection is for a single cell. Simply click on the cell you wish to select. If the table is not in edit mode, the cell will be shaded. If the table is in edit mode, the cell will be surrounded by a black border, and the contents of the cell will be displayed in the table edit window.

For the selection of multiple cells, EViews uses the concept of an *anchor cell* to determine a selection region. The anchor cell is used to mark the start of the selection region

C	4.341774	1.491847	2.910334
DMR	-5.784553	1.758724	-3.289063
SIG	2.199932	1.104584	1.991638
SIG(-1)	2.881466	1.063034	2.710605

and is used to indicate how a selection will change as you move the mouse or use keystrokes.

When edit mode is off, the anchor cell is marked as the cell with the black square in one of the four corners of the cell. When edit mode is on, the anchor cell is marked with a black border around the cell. You may toggle between edit mode on and edit mode off by clicking on the **Edit +/-** button on the object toolbar, or alternately, by right mouse clicking and selecting **Edit +/-**.

The easiest way to highlight a region is to (left) click in a cell to set an anchor point, then, while holding down the mouse button, move the mouse to select additional cells. In

C	4.341774	1.491847	2.910334
DMR	-5.784553	1.758724	-3.289063
SIG	2.199932	1.104584	1.991638
SIG(-1)	2.881466	1.063034	2.710605

addition, cell selection shortcuts allow you to select rows and columns by clicking on row and column headers, and to select rectangular regions by clicking on a cell to set an anchor cell, then SHIFT-click to select the rectangular region defined by the anchor and ending cells. You may enter CTRL-A to select all of the cells in a table.

Some of the more frequently used selection tools include:

To select	Action
Text in a cell	If edit mode is turned on, select the cell, double-click in it, and then select the text in the cell. Or select the cell and then select the text in the edit field.
A single cell	Click the cell, or use the arrow keys to move the anchor cell.
A range of cells	Click the first cell of the range, and then drag to the last cell. Or click in a cell to set the anchor, then SHIFT-click in the last cell you wish to include. Or set the anchor, and then SHIFT and press the arrow keys until desired cells are selected.
All cells in a table	Click the corner cell shared by the column and row header (the corner cell is not visible in some output views). Or press CTRL + A.
An entire row	Click the row heading.
An entire column	Click the column heading.

Adjacent rows or columns	Click and drag across the row or column headers. Or select the first row or column; then hold down SHIFT and click the last row or column heading.
More or fewer cells than the current selection	Hold down SHIFT and click the last cell you want to include in the new selection. The rectangular range between the active cell and the cell you click becomes the new selection. Or hold down SHIFT and press the arrow keys until selection is correct.

Note that row and column header selection is not always available in table views since the headers are not always displayed. For example, the estimation output view of an equation is a table that does not contain header lines. Freezing the view creates a table object that allows for cell selection using the visible headers.

Editing Cell Values

To enter or change the data in a table, you must first display the table edit window by enabling edit mode, and selecting a cell to be modified. Here, we see a table that is in edit mode, with the contents of the A1 cell displayed in the edit window just below the toolbar.

To modify the contents of the cell, simply type in the edit window. Alternately, you may double click in the cell to edit the contents. EViews will then allow you to edit the cell in place.

You may provide either alphanumeric or numeric input. If your text may be interpreted as a number, EViews will interpret the input and store the value as a number. Since the table value is stored as a number it may later be formatted using the numeric formatting tools. You may, for example, change the display of the number to scientific notation, or you may display numbers with 4 digits of precision (see "Content Formatting" on page 795).

Note that you may enter numeric expressions and have EViews evaluate them prior to placing them in the table. To evaluate a numeric expression into a cell, type " = " before the expression. For example, entering the text " = 4*5" will result in a cell value of "20". Entering an invalid numeric expression will set the cell to a numeric NA.

This latter example raises a minor issue associated with entering missing values into a table. If the text "NA" is entered into a table cell, the cell value will be set to the string "NA", not to the missing value NA. To enter a numeric missing value, you should enter the string " = NA" into the cell. We point out that the choice between entering the "NA" string or the NA value into a cell has consequences for auto-justification, or when saving values to a file.

Sorting Rows

You may easily sort the rows of a table using values in one or more columns. To sort, you must be in table edit mode. Highlight the data you wish to sort (note that the sort keys must also be selected) then right-click and select **Sort...** in the pop-up menu:
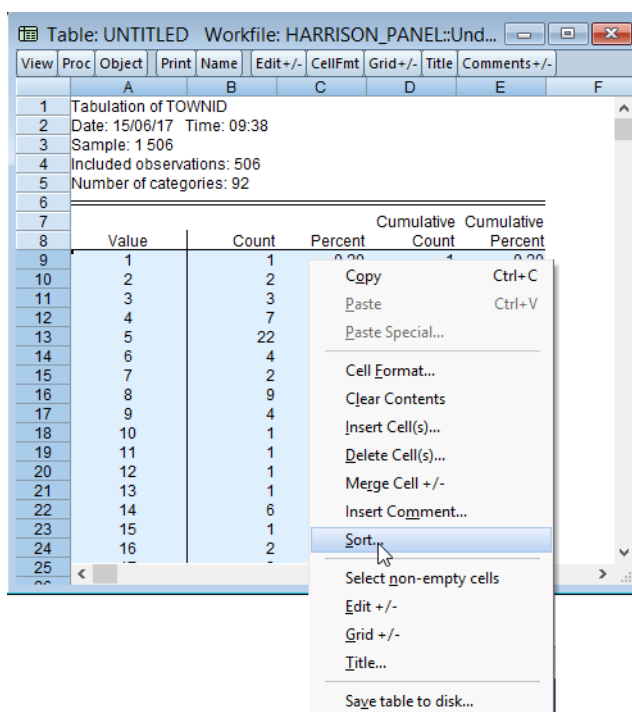
EViews will display the sort dialog prompting you to specify your sort keys and direction
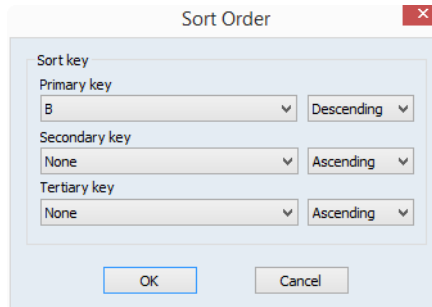
You may use the dialog to choose the columns on which to sort, and whether to use ascending or descending order. The rows will first be sorted on the basis of the primary key, and ties will be broken using the secondary and tertiary keys. Note that you may remove any existing sorting using the **None** option. Click on **OK** to sort the selection.

Note when sorting both numeric and alphabetic data in a given column that numbers will be treated as smaller than letters.

Basic Customization

You may perform basic customization of a table object by attaching a title, by adding or hiding the grid lines, or by resizing the rows or columns.

Table Title

To add a header title to the top of a table object, you should select **Proc/Title...** from the table menu, or you may click on the **Title** button on the toolbar. EViews will display a dialog prompting you to enter your title. When you enter text in this dialog, EViews displays a header title at the top center of the table. Note that the table title is different from the table name, which provides the object name for the table in the workfile.

To remove the table title, display the title dialog, then delete the existing title.

Grid Lines

To toggle on or off the grid marking the cells in the table object, click on the **Grid + /–** button on the table toolbar, or select **Proc/Grid + /-** from the main table menu.

Resizing Columns and Rows

Column widths may easily be resized in both table views and in table objects. Simply place your cursor over the separator lines in the column header. When the cursor changes to the two-sided arrow, click and drag the column separator until the column is the desired size. If you wish to resize more than one column to the same size, first select the columns you wish to resize, then drag a single column separator to the desired size. When you release the mouse button, all of the columns will be resized to the specified size.

Row heights may only be resized in table objects. Place your cursor over the separator lines in the row header and drag the separator until the row is the desired height. If you wish to resize more than one row, first select the rows you wish to resize, then drag a separator to the desired size. All of the rows will be resized to the specified size.

Double clicking a column/row edge in the header will resize the row or column to the minimum height or width required so that all of the data in that row or column is visible.

Table Cell Customization

EViews provides considerable control over the appearance of table cells, allowing you to specify content formatting, justification, font face, size, and color, cell background color and borders. Cell merging and annotation are also supported.

Cell Formatting

You may select individual cells, ranges of cells, or the entire table, and apply various formatting tools to your selection. To format the contents of a set of cells, first make certain that the table is in edit mode. Next, select a cell region, then click on **CellFmt** in the toolbar, or right mouse click within the selected cell region and select **Cell Format...**. EViews will open the **Table Options** dialog containing three tabs: **Format**, **Font/Color**, and **Borders/Lines**.

Content Formatting

The **Format** tab allows you to apply display formats to the contents of cells in table objects. Formatting of table objects may be cell specific, so that each cell may contain its own format.

You may also modify the display of numeric values, set column widths and row heights, and specify the justification and indentation.

Bear in mind that changing the height of a cell changes the height of the entire row and changing the width of a cell changes the width of the column. Column widths are expressed in unit widths of a numeric character, where the character is based on the default font of the table at the time of creation. Row height is measured in unit heights of a numeric character in the default font.

For additional discussion of content and cell formatting, see the related discussion in "Changing the Spreadsheet Display" on page 124.
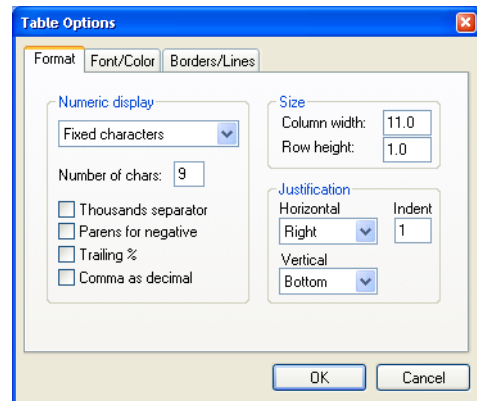
Fonts and Fill Color

The **Font/Color** tab allows you to specify the font face, style, size and color for text in the specified cells. You may also add strikeout and underline effects to the font. This dialog may also be used to specify the background fill color for the selected cells.

Where possible, the **Sample** window displays a preview of the current settings for the selected cells. In cases where it is impossible to display a preview (the selected cells do not have the same fonts, text colors, or fill colors) the sample text will be displayed as gray text on a white background.
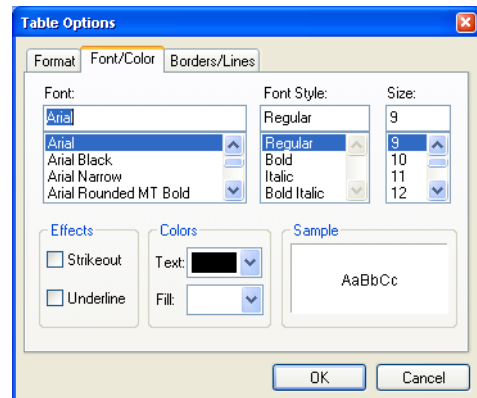
Note also that EViews uses the special keyword **Auto** to identify cases where the selection region contains more than one text or fill color. To apply new colors to all of the selected cells, simply select a **Text** or **Fill** color and click on **OK**.

Borders and Lines

The last tab, labeled **Borders/Lines** is used to specify borders and lines for the selected table cells.

Simply click on any of the **Presets** or **Border** buttons to turn on or off the drawing of borders for the selected cells, as depicted on the button. As you turn on and off border lines, both the buttons and the display on the right will change to reflect the current state of your selections. Note also that there is a checkbox allowing you to draw double horizontal lines through the selected cells.
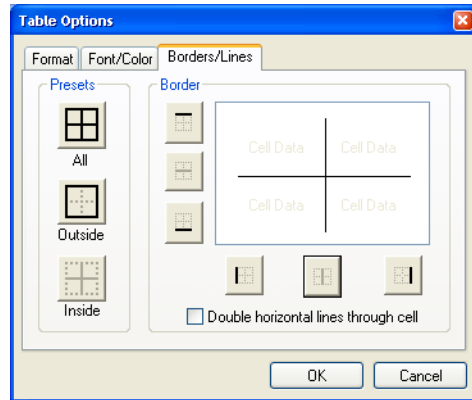
It is worth noting that the appearance of the **Borders/Lines** page will differ slightly depending on whether your current selection contains a single cell or more than one row or column of cells. In this example, we see the dialog for a selection consisting of multiple rows and columns. There are three sets of buttons in the **Border** section for toggling both the row and column borders. The first and last buttons correspond to the outer borders, and the second button is used to set the (between cell) inner border.

If there were a single column in the selection region, the **Border** display would only show a single column of "Cell Data", and would have only two buttons for modifying the outer vertical cell borders. Similarly, if there were a single row of cells, there would be a single row of "Cell Data", and two buttons for modifying the outer horizontal cell borders.
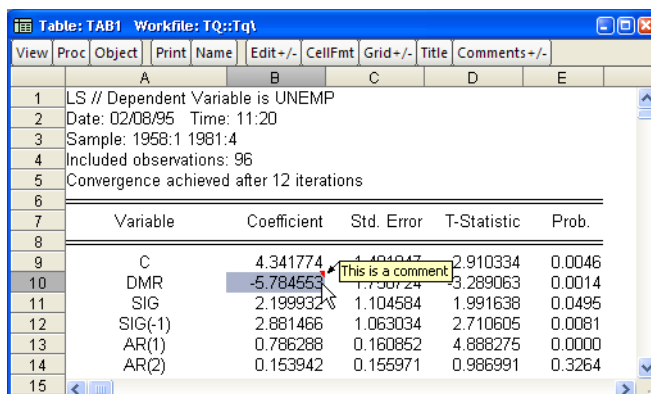
Cell Annotation

Each cell of a table object is capable of containing a comment. Comments may be used to make notes on the contents of a cell without changing the appearance of the table, since they are hidden until the mouse cursor is placed over a cell containing a comment.

To add a comment, select the cell that is to contain the comment, then right mouse click and select **Insert Comment...** to open the **Insert Cell Comment** dialog. Enter the text for your comment, then click **OK**. To delete an existing comment, just remove the comment string from the dialog.

If comment mode is on, a cell containing a comment will be displayed with a small red triangle in its upper right-hand corner. When the cursor is placed over the cell, the comment will be displayed.

If comment mode is off, the red indicator will not be displayed, but the comment will still appear when the cursor is placed over the cell.

Use the **Comments + /-** button in the tool bar to toggle comment mode on and off. Note that the red triangle and comment text will not be exported or printed.

Cell Merging

You may merge cells horizontally in a table object. When cells are merged, they are treated as a single cell for purposes of input, justification, and indentation. Merging cells is a useful tool in customizing the look of a table; it is, for example, an ideal way of centering text over multiple columns.

To merge several cells in a table row, simply select the individual cells you wish to merge, then right click and select **Merge Cell + /-**. EViews will merge the cells into a single cell. If the selected cells already contain any merged cells, the cells will be returned to their original state (*unmerged*).

Here, we begin by selecting the two cells B1 and C1. Note that B1 is the anchor cell, as indicated by the edit box surrounding the cell, and that B1 is center justified, while C1 is right justified.

If we right mouse click and select **Merge Cell + /–**, the two cells will be merged, with the merged cell containing the contents and formatting of the anchor cell B1. If you wish C1 to be visible in the merged cell, you must alter the selection so that C1 is the anchor cell.

We see that the B1 and C1 cells are merged, as indicated by the large selection rectangle surrounding the merged cells.

Bear in mind that the C1 cell has not been cleared; its contents have merely been hidden behind B1 in the merged cell. Editing the value of the merged cells will replace the value in the cell B1, but has no effect on hidden cells, in this case C1.

If the merged cell is selected, toggling **Merge Cell +/-** will unmerge the cell so that cells are returned to their original form. The contents of C1 will once again be visible and may be modified using any of the table display formatting tools.

Printing Tables

To print a table object, click on the **Print** button on the table tool bar or select **View/Print...** from the main EViews menu to display the **Print** dialog.

The top section of the **Print** dialog may be used to select a printer and print options, or to redirect the print job to an RTF file, text file, table object, or spool object (see "Print Setup" on page 880).
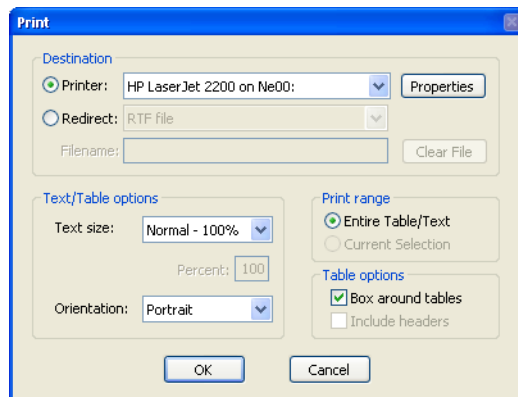
You may use the remainder of the dialog to scale all of the text in the table, to select the orientation of the table (**Portrait** or **Landscape**), to draw or not draw a box around the table, and to include or not include the table headers in the print job.

If you have selected a range of cells in the table, the **Print range** section of the dialog will offer you choice of printing the entire table or only printing the current selection.

Copying Tables to the Clipboard

You may copy-and-paste a table to the Windows clipboard, from which you may paste the table contents into your favorite spreadsheet or word processing software.

Simply select the cells that you wish to copy and then choose **Edit/Copy** from the EViews main menu, or **Copy** from the right mouse button menu. The **Copy Precision** dialog box will open, providing you with the option of copying the numbers as they appear in the table, or at their highest internal precision.
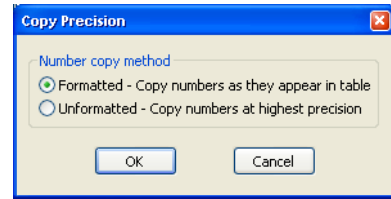
After you make a choice, EViews will place the table on the clipboard in Rich Text Format (RTF), allowing you to preserve the formatting information built into the table. Thus, if you copy-and-paste a table from EViews into Microsoft Word or another program which supports RTF, you will create a nicely formatted table containing your results.

To paste the clipboard contents into another application, switch to the destination application and select **Edit/Paste**. Note that some word processors provide the option of pasting the contents of the clipboard as unformatted files. If you wish to paste the table as unformatted text, you should select **Edit/Paste Special**.

Saving Tables to a File

EViews allows you to save your table objects in several file formats: Comma Separated Value (CSV), tab-delimited text (ASCII), Rich Text Format (RTF), Web page (HTML), Enhanced Metafile (EMF), Portable Document Format (PDF), or LaTex (TEX) file.

To save the table to disk, with the table window active or with table cells selected, right mouse click or select **Proc**, then select **Save table to disk...** to bring up the **Table File Save** dialog. The dialog displays default values from the global settings.

In the top portion of the dialog, you should provide the name of the file you wish to create. EViews will automatically append an extension of the proper type to the name (here, ".HTM" since we are saving a Web HTML file), and will prepend the default path if an explicit path is not provided.

Next, select the **File type**. You may select **Comma Separated Value**, **Tab Delimited Text-ASCII**, **Rich Text Format**, **Web page,** **Enhanced Metafile,** or **Portable Document Format**. The options section of the dialog allows you to save the entire table or only those cells that are currently selected, and, for all but CSV and ASCII files, to scale the table size.
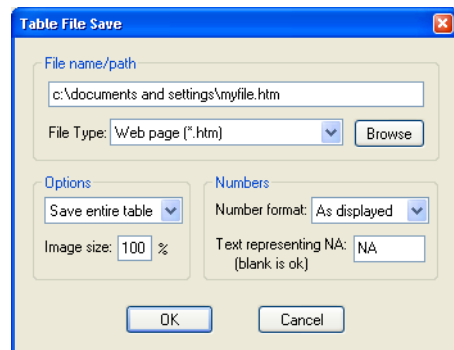
You may also specify options for how numbers are to be treated when written. You may specify a **Number format** so that numbers are written **As displayed** in the table, or using **Full precision**. In addition, you may change the text used to write missing values.

For LaTex output, you have the additional option to output full LaTeX document specification. Alternatively, you can disable the **Include document specification** option and the minimal amount of code will be generated for you to insert into your preexisting document.

Importing Tables

You may import data into a table object from a foreign data source, such as Excel or text files using the File/Import/Table from file... menu option. The import process closely resembles that for importing data into a series object, as outlined in.

Table Commands

EViews provides tools for performing extensive table customization from the command line or using programs. See "Table" (p. 782) in the *Object Reference* for additional details.

Text Objects

Some output views have no formatting and are simple displays of text information. Examples are representations of an equation and results from X-11 seasonal adjustment. If you freeze one of these views, you will create a text object.

You can also create a blank text object by selecting **Object/New Object.../Text** in the main EViews menu or by simply typing "text" in the command window. Text objects may be used whenever you wish to capture textual data that does not contain any formatting information.

Printing of text objects follows the same procedure and offers the same options as printing a table object; see "Printing Tables" on page 798.

Chapter 17. Spool Objects

EViews offers a *spool object* that is capable of holding multiple tables, graphs, text, and spools, allowing you to form collections of output objects. You may find spools to be useful for organizing results, for example, for creating a log of the results for a project or an EViews session, or perhaps for gathering output for a presentation.

In addition to being an output object container, spool objects provide easy-to-use tools for working with the objects in the spool. Among other things, you may manage (add, delete, extract, rearrange, hide) or customize (resize, space and indent, title and comment, and edit) the spool and the individual objects in a spool.

Creating a Spool

To create a spool object in a workfile, you may select **Object\New Object\Spool** from the workfile menu, optionally enter a name in the **Name for object** edit field, and click on **OK**. Alternately, you may simply enter the declaration command spool followed by an optional name in the command window, then press ENTER.

Spools may also be created by printing from an EViews object to a non-existent spool. You may simply redirect the print output of an object you wish to add to the spool (see "Adding Objects", below).
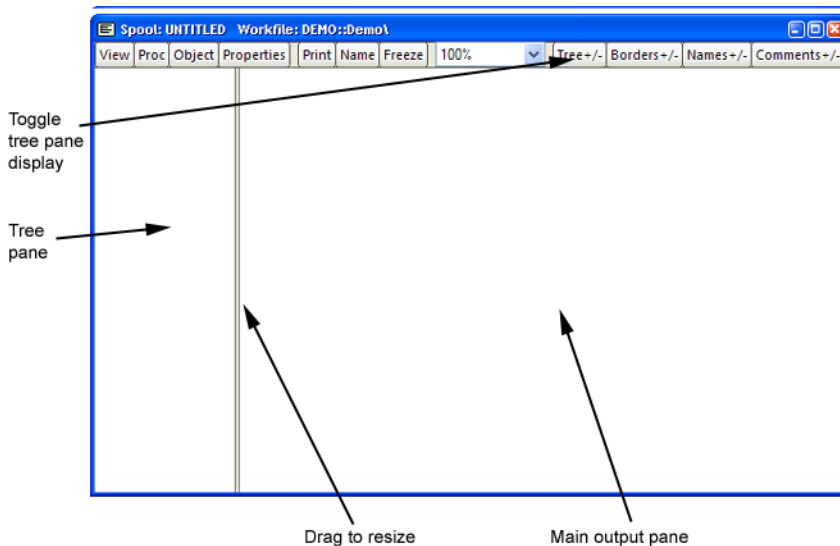
Here we see the spool view of an empty, unnamed spool object. As depicted here, the default is to display the contents of the spool using two panes. The left pane is the *tree pane*

which shows a tree-structure listing of the objects contained in the spool. You may use the tree pane to navigate quickly from object to object. The right pane is the *main output pane* which shows the actual output view objects.

You may select and drag the separator between the two panes to change their relative sizes, and you may use the **Tree +/–** on the toolbar to show and hide the tree pane. Note that hiding the tree pane provides you with a larger main window display area, but makes a number of spool management and customization tasks somewhat more difficult.

Managing the Spool

Most of your time using a spool will be spent managing the output objects contained in the spool. EViews provides easy-to-use tools for the all of the basics tasks like adding, deleting, extracting, and rearranging objects, as well as more advanced operations such as hiding objects and flattening portions of the tree.

Adding Objects

You may add an output object to a spool using any of the three primary methods: printing to the spool, copy-and-pasting objects, using the spool append procedure.

Printing to a Spool

Perhaps the easiest approach is to redirect the printing of the your object output into a spool object. You may, for example, display the estimation results for an equation, or perhaps the histogram view of a series, and "print" the output directly into a spool. Note that this method does not require you to freezing the original object view to create a separate output object.

To redirect your print job into the spool, simply click on **Print** from the object toolbar, or select **File/Print...** from the main EViews menu. In the **Print** dialog that opens, you should set the **Destination** to **Redirect**, specify **Spool object** as the redirect target, and then specify the name of the spool to which you wish to print. If you specify a spool object that does not exist, a new spool with the specified name will be created and a copy of your object will be added to it.
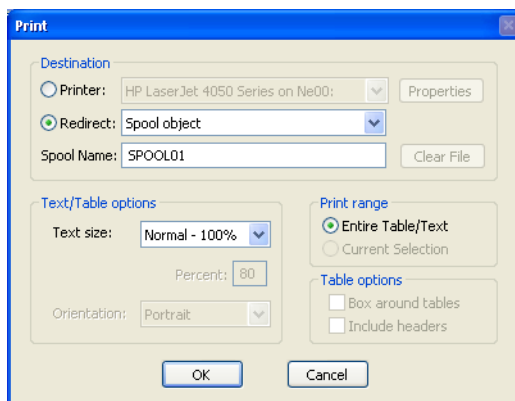
Click on **OK** to continue. EViews will add an output object containing the contents of the object window to the spool object.

If you wish to send all subsequent print jobs to the spool object, you may use the main **File/ Print Setup...** dialog or the output command to change the default print destination to your spool (see "Print Setup" on page 880 and output (p. 436) of the *Command and Programming Reference*). Then you may simply display your view, then click on **Print** and **OK** to send the output to the spool object.

Copy-and-Paste to a Spool

Alternately, objects may be copied from the workfile window and pasted into an existing spool object. Simply select the objects you wish to copy in the workfile window and press CTRL-C or select **Edit/Copy** from the main or the right-button menus. Next display the spool object window, and paste into the main object pane by pressing CTRL-V, or selecting **Edit/ Paste** from the main or right mouse-button menus.

EViews will append copies of output (graph, table, text, and spool) objects directly into the spool; for other types of objects, EViews will append copies of the frozen default views. For example, if you copy-and-paste a series, EViews will add a copy of the default spreadsheet view to the spool.

We emphasize that the objects contained in a spool are frozen copies of the original objects. The objects in the spool are independent of the source objects so that changes made to either the copy or the original will not have an effect on the other.

Appending to a Spool

Lastly, you may select **Proc/Append/Existing Object** from the menu of an existing spool object. You will be prompted by a dialog where you can enter the names of the objects to be added and commands for displaying the desired output windows in a space delimited list. Here, for example, we instruct EViews to display the default spreadsheet view for GDP, a line graph for GDP, and as a Wald test for "c(1) = 0" in equation EQ01, and to save the results for each of these three views into the spool.
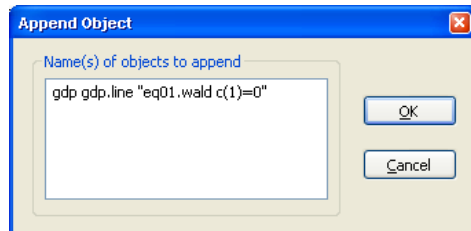
Note that when adding multiple outputs you should make certain to separate the names and commands by spaces. If necessary, you should enclose the text in double quotes, if, for example, the command statement includes spaces as in the Wald example above.

Click on **OK** to continue. The tree pane of the spool now shows the three object names and types, while the main output pane allows you to view the objects. You may scroll the window between output objects by clicking on the icon in the tree pane or by using the scrollbar on the right-hand side of the output pane.
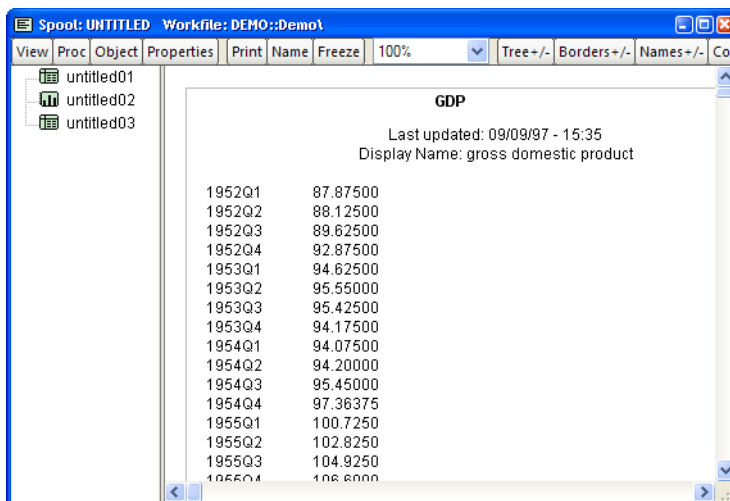
Note that the newly added objects are all given names that are variations on the name "UNTITLED". You may wish to provide more descriptive names (see).

Click on **Name** and enter MYSPOOL to name the spool object.

Embedded Spools

Spool objects may contain other spool objects. Nested spools allows you to categorize or group output objects within a spool.

There are two ways to add a spool object to an existing spool. First, if the spool you wish to add exists in your workfile then you may insert it into a spool as you would any other object using print, copy-and-paste, or the append proc as described in .

Second, you may add a new (empty) spool object to the spool by selecting **Proc/Append/New Spool**... and entering a space separated list of one or more names for the new spools. The names given to spools must be valid EViews names. EViews will create and append a new spool for each specified name. If there are duplicate names in the spool, the new spools will be named using the next available name.
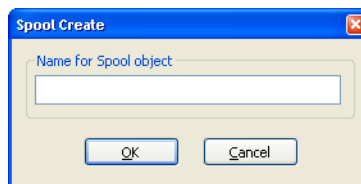
Suppose, for example, we have the named spool object MYSPOOL, which contains three output objects (A1, A2, and A3). We next create a new, UNITITLED spool object containing the output object B1. Lastly, we append MYSPOOL, and it's three output objects by copying-and-pasting MYSPOOL into the UNITITLED main object pane.

The contents of MYSPOOL are added to the end of the spool object. The copy of the spool object, shown with the spool icon, is assigned the name UNTI-TLED01.

The tree entry for UNITLED01 is shown here in open fash-

ion, with the three output objects in the embedded spool displayed in the branch of the tree. Clicking on B1 in the root of the tree scrolls the output window so that B1 is displayed in the main output pane; clicking on A1 in the UNTITLED01 branch of the tree scrolls to the A1 output object from MYSPOOL. As with all tree structures, you may click on the tree icon next to UNTITLED01 to open or collapse the display of the branch.
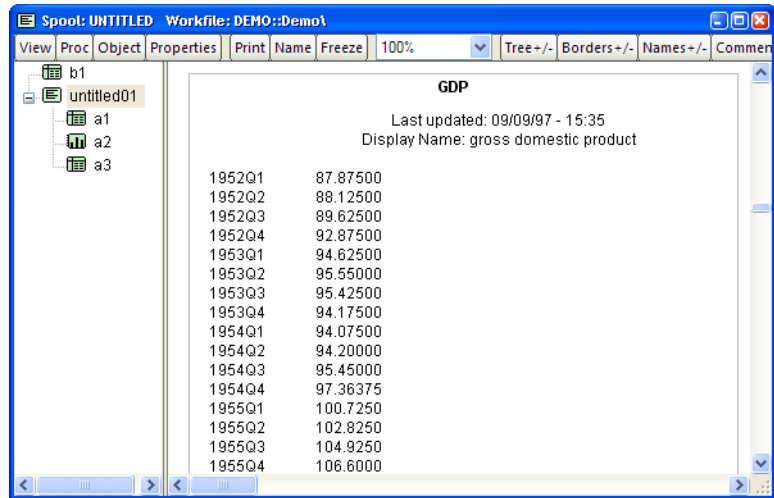
It is worth noting that the embedded spool object is a full-fledged output object inside the parent spool. Thus, selecting an embedded spool and performing operations such as naming, commenting, hiding, indenting, resize, *etc.* applies to the embedded spool object, and not to the objects inside the embedded spool. The most important implication of this fact is that if you wish to change the properties of the objects within an embedded spool, you must display the embedded spool in edit mode so that you can access its settings and its individual elements ("Editing Objects in a Spool" on page 816), or you must first move the object out of the spool, or you may flatten or move all of the objects out of the embedded spool ("Flattening a Spool" on page 811).

For example, the easiest way to turn on comments for all of the objects within the embedded spool in the example above is to double click on UNTITLED01 in the tree pane to activate edit mode, then to modify its display settings by clicking on the **Comments +/–** button. By placing the embedded spool in edit mode, we gain access to its display settings; if instead we pressed **Comments +/–** in standard mode, we would activate comments for the objects

in the parent spool only. (See "Editing Objects in a Spool," on page 816 for additional discussion).

Naming Objects

To rename an output object in the spool, select the object name in the tree pane or select the output object in the main output page, then right mouse-click and select **Name...**. EViews will display the **Object Title** dialog.

Note that the dialog may also be called up by double-clicking on the name of the object, if visible, in the main object window (see "Display Properties" on page 813 for a discussion of name display settings).

Simply enter a new, valid EViews name in the **Object Name** field, and if desired a display name in the optional **Display Name** field. The display name allows you to provide a more descriptive label that may be employed in place of the object name in the tree pane view when displaying object names.

Since all spool output objects will, by default, be named using some variant of "UNTITLED", we recommend that if you have spools with more than a few objects, you assign them more descriptive object or display names.

Adding Comments

Comments may be added to objects, allowing you to annotate individual objects in the spool. To add a comment to an object or to change the existing comment, select the object of interest, then choose the right button menu item **Comment…** to bring up the **Insert Object Comment** dialog.
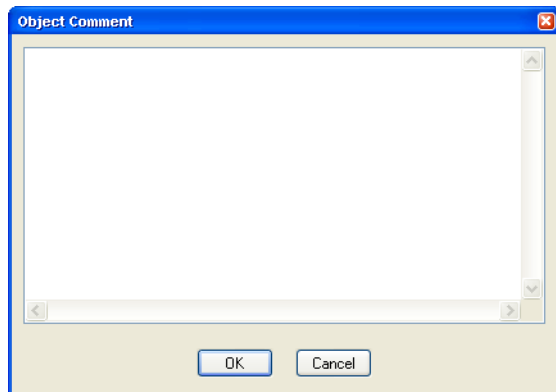
The dialog may also be called up by double-clicking on the comment for an object, if visible, in the main object window (see "Display Properties" on page 813 for a discussion of comment display settings).

Enter your comment or edit the existing comment and click on **OK** to add it to the object.

Hiding Objects
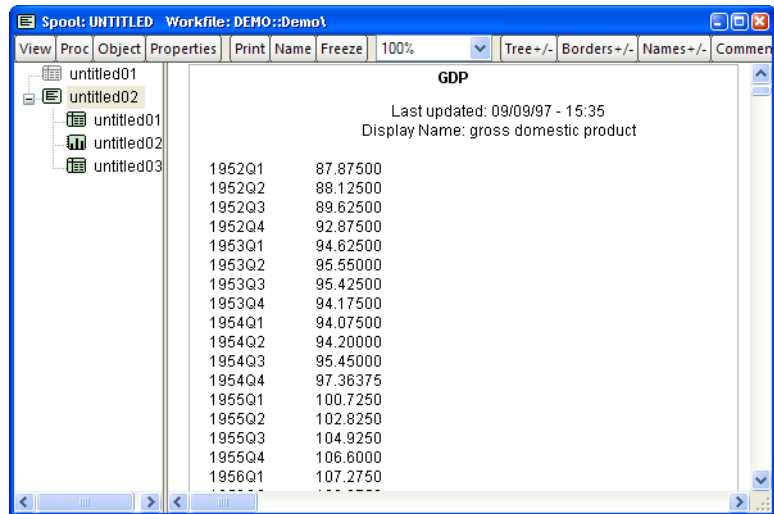
After creating a spool with objects of various types, you may wish to view a subset of the objects. You may hide objects either individually or by type (tables, graphs, or text).

Hiding Individual Objects

You may hide or show objects individually using a variety of methods. The easiest method is to select an object in the spool using the tree pane or the main window, and select **Show/Hide** from either the **Proc** or right mouse-button menus to toggle the item between hidden and visible.

Note that hiding an embedded spool hides all of the objects in the embedded spool.

Objects that are hidden do not appear in the main window; if, however, the spool settings are to display titles or comments ("Display Properties" on page 813), the object title or comments will still be visible.

The icon for hidden objects will be grayed out in the tree pane. Here, we see that the UNTITLED01 table object in the main branch of the spool is hidden.

An alternate method requires that object title display be turned on. Click the **Properties** button to bring up the **Spool Properties** dialog, and check the **Show Names** checkbox. When titles are displayed, the main window
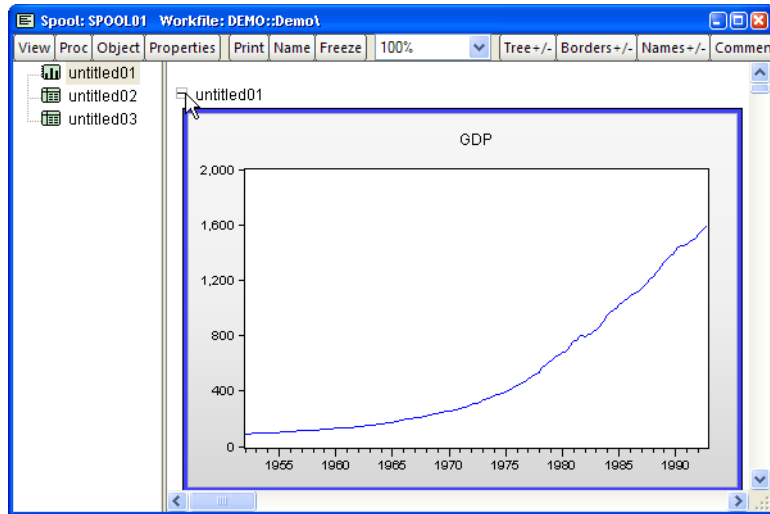
will contain a box containing a " + " or "–" alongside the name of the objects in the spool. Clicking on the box will toggle the object between being hidden and visible. In this example, clicking on the "–" symbol will hide UNTITLED01.

Hiding Types of Objects

In some cases it is desirable to hide objects of a specific type. Objects can be hidden by type by selecting **Proc/Hide** and choosing **All**, **Tables**, **Graphs**, or **Text**. Conversely, to show hidden objects of a specified type select **Proc/Show**.
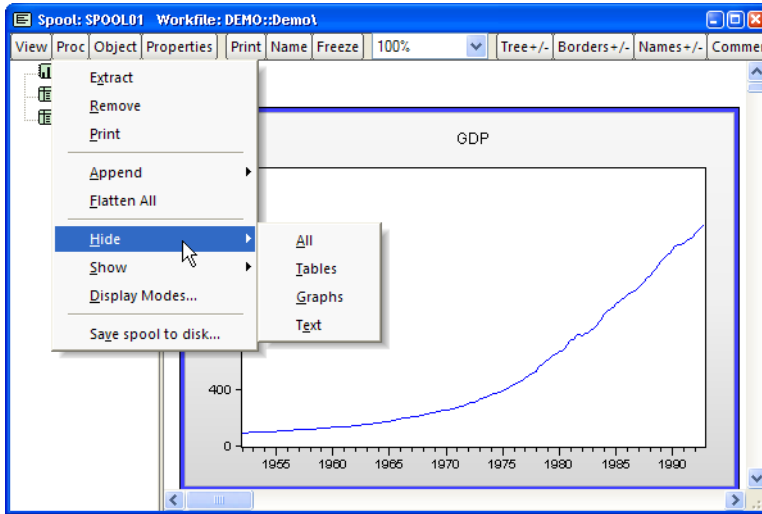
Resizing Objects

You may use the spool object to override the sizes of individual output objects.

The default is for graph objects to be displayed at a fixed size while tables and text objects are displayed at their native sizes. This implies that all graphs are displayed with the same width, whatever the size of the individual graph, while tables and text objects are displayed using the widths and font sizes specified in the individual object.
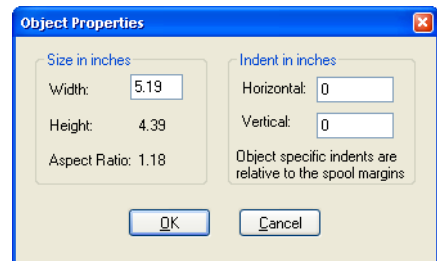
There are two different ways of using the spool to change the display size for an object. But before you are allowed to alter the display size of an object you must make certain that its display mode is set to **Variable Width** or **Variable Width w/ Limit** (see "Display Modes" on page 815 for details).

First, you may select the object in the tree pane or by clicking on the object in the main window. Press the right mouse-button and select **Properties...** to open the **Object Properties** dialog. Enter the desired **Width** in the edit field and click on **OK**. EViews will resize the object, maintaining the original aspect ratio. Font sizes in the object will be scaled accordingly.

Alternately, if the frame for an object is displayed in the output window, you may drag the border to resize the object (see "Display Properties" on page 813 for a discussion of border display settings).

Note that resizing the object does not alter the native size of the object. You may at any time select an object, right-click, and choose **Reset size** to return an individual object to its native size. Or, you may use the spool object to reset the display of all of your objects to their native sizes (see "Display Modes," on page 815).

Indenting Objects

To change the indentation of an object you should first select the object in the tree pane or by clicking on the object in the main window. Press the right mouse-button and select **Properties...** to open the **Object Properties** dialog. Enter the desired indentation in the **Horizontal** and **Vertical** edit fields, and click on **OK** to continue.

Note that the indentation values set in the **Indent in inches** section dialog will be added to any margins set for the spool.

Removing Objects

To remove an object from a spool select the object then press the **Delete** key or select **Remove** from the right mouse-button menu. EViews will prompt you for whether you wish to continue. Click on **OK** to proceed.

Note that if you remove a spool object, you will remove it and all of the objects contained in the spool. If you wish to retain some of the nested objects you must move them outside of the embedded spool ("Rearranging Objects" on page 810) prior to deleting the spool. Alternately, you may simply delete a subset of objects within the embedded spool.

Extracting Objects

You may copy, or *extract*, an object in a spool into the workfile. To extract an object, simply highlight the object, and then select **Extract** from the right mouse-button menu, or click on **Proc/Extract**. EViews will create an untitled output object in the workfile containing a copy of the object in the spool.

Rearranging Objects

Suppose, for example, that you have placed a number of output objects in the spool, but they are not in the desired order for your presentation. You may reorder objects in the spool using the tree pane to move objects (one at a time) to different positions in the spool.

To move an object, you should select it in the tree pane and drag it to the desired position. A horizontal black bar will indicate the object's new location when the mouse button is released. If the destination is an embedded spool, EViews will move the object into the embedded spool.
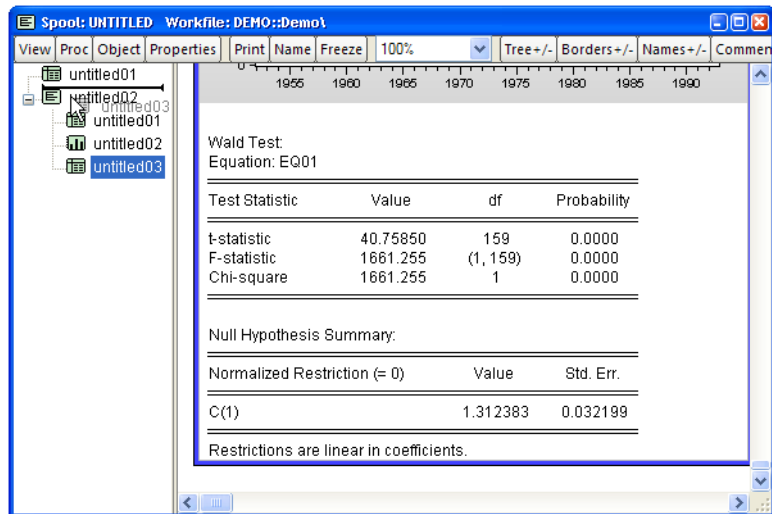
Note that objects may easily be moved into and out of embedded spool objects. In our example, we move the table object UNTITLED03 from the embedded spool UNTITLED02 into the top level of the spool, just below UNITITLED01.

In some cases, moving objects into or out of an embedded spool will cause an object name conflict. In these cases, EViews will display a dialog prompting you to provide a new unique name (a new name will be suggested), along with an optional display name which may be used in the tree view.

As noted in the dialog, display names need not be unique.
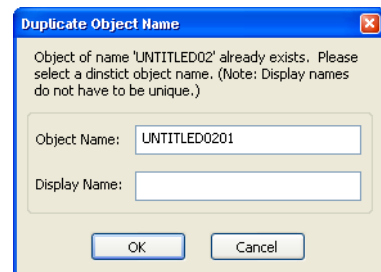
Flattening a Spool

You may move all of the objects out of an embedded spool by *flattening* the spool. Flattening moves out all of the output objects in an embedded spool (including objects in any spools embedded in the embedded spool), and then deletes the now empty spool.

Select the name of the embedded spool you wish to flatten in the tree pane, then click on **Proc/Flatten All** or press the right mouse-button and select **Flatten**. EViews will replace the spool with all of the output objects in the spool.
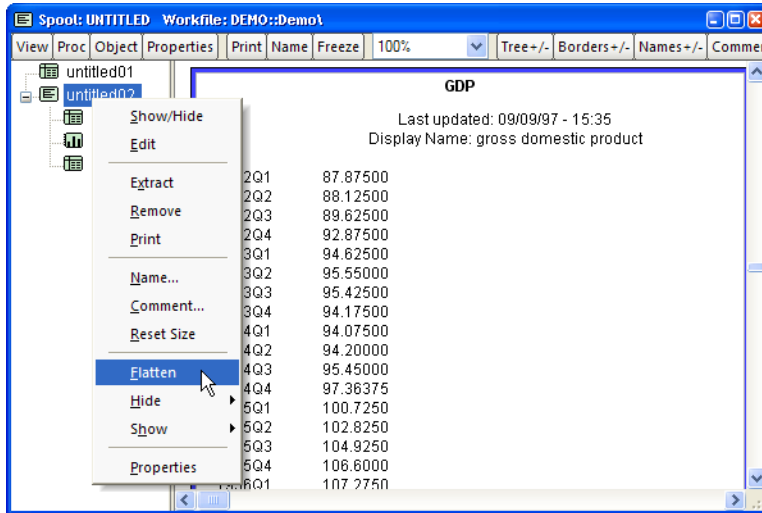
In our example, we select the spool object UNTITLED02, and use the right mouse-button to select **Flatten**. EViews will replace UNTITLED02 with all of its child output objects, renaming the objects as necessary to avoid name conflict.
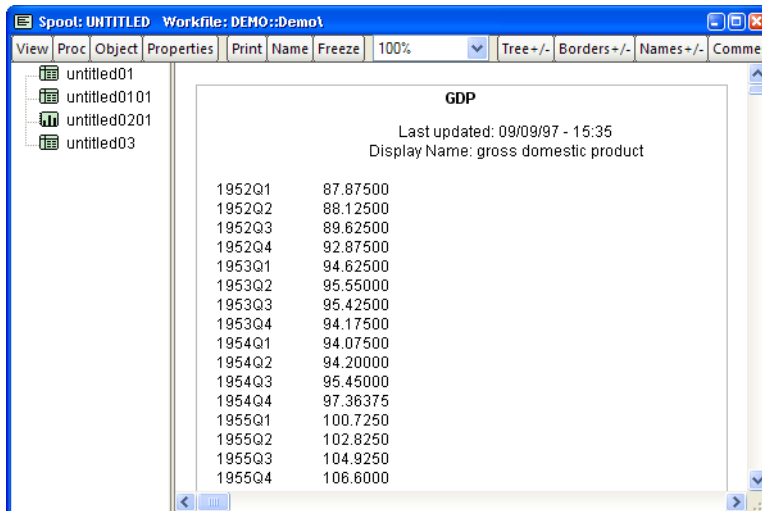
In this example, UNTITLED01 and UNTITLED02 have automatically been renamed in order to resolve the name conflict resulting from flattening the spool.

Customizing the Spool

The spool object provides several options for customizing the display of object elements in the spool.

Display Properties

Press the **Properties** button in the tool bar to bring up the **Spool Properties** dialog which allows you to set the basic display properties for the spool object:
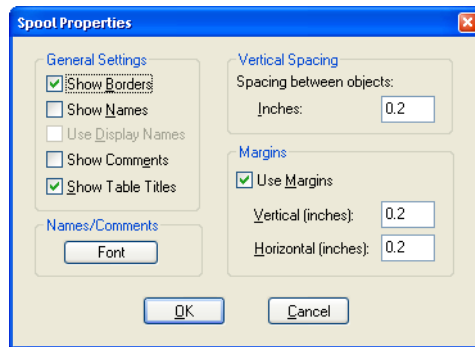
General Settings
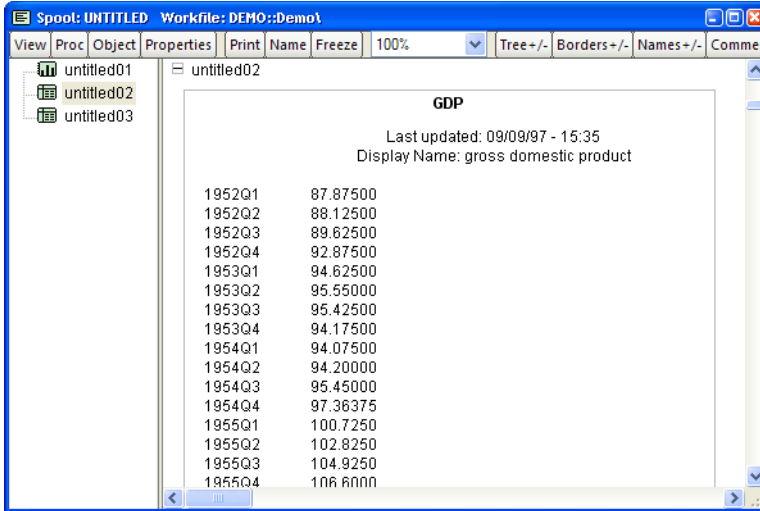
The **General Settings** section of the dialog allows you to show and hide various parts of the spool.

- The **Show Borders** option allows you to display a frame around each of the output objects in the main spool window. Dragging the border of an object will resize that object (see "Resizing Objects" on page 809).

- The **Show Names** and **Use Display Names** checkboxes control the display of names in the main output window. Object name labels for each object are displayed if **Show Names** is checked; the display names are used in place of the object name if **Use Display Names** is checked (see "Naming Objects" on page 806).

- The **Show Comments** checkbox of this dialog determines whether individual object comments are displayed (see "Adding Comments" on page 806).

- **Show Table Titles** controls whether to display the table objects using their own titles.

Note that the **Show Borders**, **Show Names** and **Show Comments** settings may be accessed directly from the spool object toolbar. For example, clicking on the **Names +/–** button is the same as checking or unchecking the corresponding box in the **Spool Properties** dialog.

The last option, **Show Table Titles**, requires some discussion. In EViews, table objects have titles which generally differ from the name given the table in the spool. When **Show Table**

Titles is checked (the default), the title of each table will be displayed. If we also turn on **Show Names**, we see that the object name in the spool is "untitled02" and the table title is "GDP":
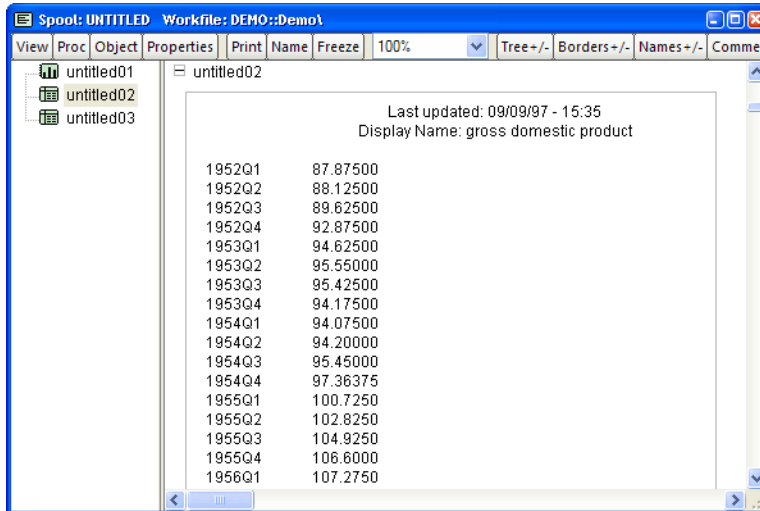
Next, we turn table titles off, which hides the "GDP" table title. Note here that the first line of the object display is now the update date:

Names/Comments Font

Click on the **Font** button to bring up a dialog for setting the font size, color, and family for text in displayed names and comments.

Vertical Spacing

The **Vertical Spacing** edit field controls the amount of vertical space placed between output objects in the spool.

Margins

The **Use Margins** checkbox controls whether the spool will display the contents of the output window with vertical and horizontal margin padding. If you wish to provide spool margins, you should select this option and enter values for the size of the vertical and/or horizontal margins.

Note that this setting controls margins for the spool object itself. Margins for individual objects may be adjusted by selecting an object, right mouse clicking, and selecting **Properties...** (see "Indenting Objects," on page 810). Individual object margins are specified relative to the margins for the spool object.

Display Modes

When you bring an output object into a spool, EViews copies the original output object at its original size; we term this the native object size. Subsequent editing of the object in the spool may change this native size (see "Editing Objects in a Spool" on page 816).

The spool object allows you to display objects at sizes that differ from the native sizes. By separating the display size of the object in the spool from the native object size, EViews permits customizing the display of objects without requiring manual editing of the native size in each output object.

By default, EViews displays graph objects in a spool using a uniform width while table and text objects are displayed in using their native widths and font sizes. This default setting allows for a relatively uniform appearance for the objects in the spool.

You may alter the display settings for the spool to allow graphs and tables to be displayed at different sizes. Click on **Proc/Display Modes...** to bring up the **Display Modes** dialog.

The top portion of the dialog offers **Graph Settings** that control the display size of graph objects in the spool. The **Mode** dropdown menu allows you to choose between various settings:
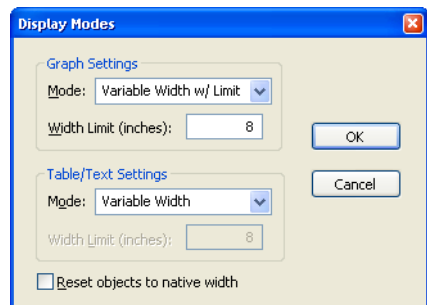
- The default **Fixed Width** setting fixes the spool display sizes for graphs at a uniform width, as specified in the **Width Limit (inches)** edit field.

- **Variable Width** displays the graphs at the native sizes, but permits individual customization as described in "Resizing Objects" on page 809.

- **Variable Width w/ Limit** displays graph objects in their native widths so long as they do not exceed the specified limit, otherwise the display size is adjusted accordingly. You may customize the display sizes of individual objects.

The **Table/Text Settings** portion of the dialog provides similar control over tables and text objects (though **Fixed Width** is not available for tables). For table and text objects, the default is to use **Variable Width**, but you may use the dropdown to change the settings to **Variable Width with limit**. If a table or text object is resized, the font will be scaled accordingly.

You may, after performing individual display sizing ("Resizing Objects" on page 809) decide that you wish to revert to the original native sizes; selecting **Reset objects to native width**, and clicking on **OK** will set all of the objects in the spool to use their native sizes.

Editing Objects in a Spool

In most cases, you may customize an object in a spool. Changes to an output object in edit mode are equivalent to editing the object in its own window.

To make changes to the object you must first place the spool in *edit mode*. Simply double click on the object in either the tree or the output panes to put the spool in edit mode. Alternately, you may select the object you wish to edit, right-click, and choose **Edit**.

When a spool is placed in edit mode, EViews will display the object being edited using the full width of the window. The toolbar will change to show the options for the edit object, instead of those for the spool object. In addition there be a new region, just below the toolbar, showing the object and display names for the edit object, as well as a **Close** button that saves any modifications, turns off edit mode, and returns you to the standard spool window.

For example, edit mode for a graph object will show the graph in the main display, along with options for customizing the graph. Here we see a spool in edit mode showing the toolbar for a graph object. Clicking on **Options** or double clicking on the graph brings up the standard **Graph Options** menu allowing you to perform extensive customization of the graph. Similarly, the **Proc** menu and toolbar buttons may be used to add text, line shading, or to apply template options (see "Graph Options," on page 768).
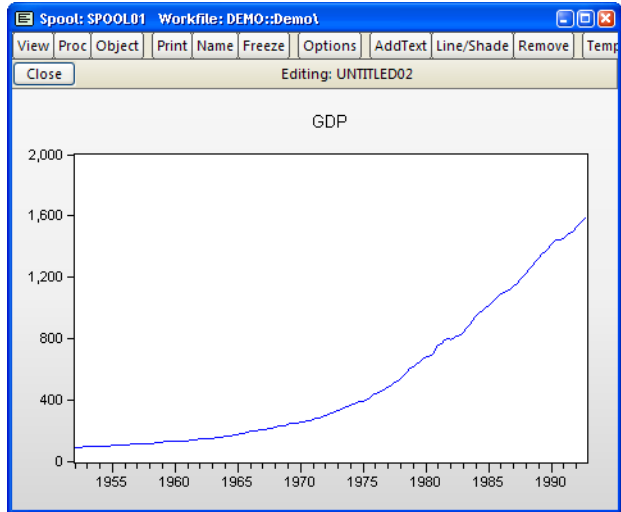
Similarly, editing a table in the spool allows you to perform extensive customization of the table. If a table is placed in edit mode, you will be able to perform all of the standard table editing operations, including row and column resizing, setting font sizes and styles, formatting cells, adding borders, titles, or comments, or simply changing the contents of a cell (see "Basic Customization," on page 793 and "Table Cell Customization," on page 794).

One important case of object editing involves editing an embedded spool object. When you double click on an embedded spool object or select it, right-click, and choose **Edit**, EViews changes the spool display to show the embedded spool in edit mode:
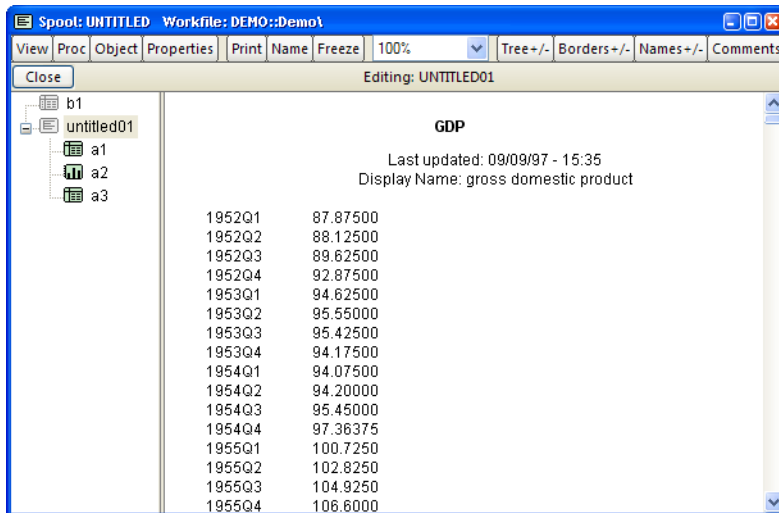
Edit mode for the embedded spool is visually similar to standard mode for an arbitrary spool. There are only two visual cues that we are in edit mode in a parent object. First, just below the toolbar is the standard edit mode line containing the **Close** button and the description of the object name. Second, the icons for the objects B1 and UNTITLED01 are now grayed-out in the tree window, indicating that they may not be selected.

Indeed, working with an embedded spool object in edit mode is identical to working with a spool object in standard mode; you may perform all of the operations in this chapter using the embedded spool and its elements. Clicking on the **Properties** button brings up the **Spool Properties** dialog which allows you to set display options for the selected spool. You may, for example, use the properties dialog to increase the vertical spacing or to display table titles for objects within the embedded spool. Since we are editing the embedded spool, changes in the spool in edit mode have no impact on the objects and settings in the parent spool. (See also "Embedded Spools," on page 804).

Once you are done editing your object, you should click on the **Close** button to save all of your object changes and to return to the standard display mode.

Printing a Spool

To print the spool object, simply click on the **Print** button on the toolbar or select **File/Print...** from the main EViews menu to display the **Print** dialog.

The top section of the **Print** dialog may be used to select a printer and print options, or to redirect the print job to an RTF file, text file, table object, text file, or spool object (see "Print Setup" on page 880).

The remainder of the dialog offers various printing options specific to spool objects.
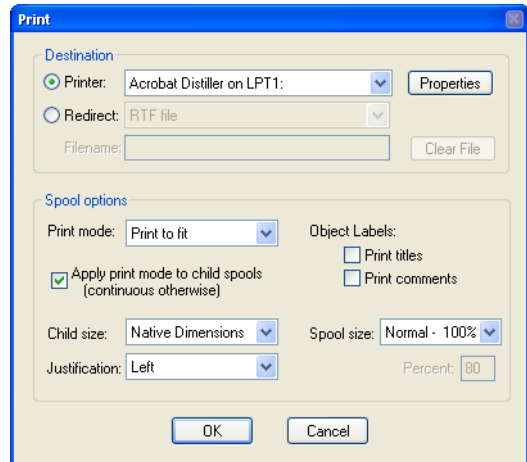
Print Mode

You may use the **Print mode** dropdown to specify where page breaks are to be used in your print job. There are three possible modes (**Print to fit**, **Continuous**, and **Individual**). If you select **Print to fit**, objects will not, unless larger than a page, be split across pages. In **Continuous** mode, objects in the spool will be printed successively and page breaks will occur when the bottom of a page has been reached. **Individual** mode prints the spool with a page break after every object. Note that whatever the print mode setting, graph objects are never split across pages.

If the spool contains embedded (child) spools, you will be provided with an additional option for whether to treat the embedded spool as a single output object or to treat it as a set of individual objects. In practical terms, this choice is equivalent to deciding between printing the contents of embedded spools in continuos mode (single object) or printing the objects in the spool using the selected print mode (multiple objects). To employ the latter method, you should check the box labeled **Apply print mode to child spools**.

To demonstrate the effects of selecting this option, assume we have a parent spool containing a table and an embedded spool which contains two graph objects. Suppose that we set the **Print mode** to **Individual**, and check the **Apply print mode to child spools** option. The resulting print job will generate three pages of output: a page for the table, and one for each of the individual graphs. Unchecking the option would (if the graphs both fit on a single page), generate two pages of output: one for the table, and a second for both graphs.

Size and Position

You may use the **Spool size** dropdown to specify a custom size for printing. For example, selecting **Custom** and entering 80 in the **Percent edit** field will scale the contents of the spool to 80% of the full size prior to printing.

You may use the **Child size** to determine the relative size of child objects. Selecting **Native Dimensions** prints the objects in the spool at their native sizes so that they will appear as they would if printed individually, outside of the spool. If instead you select Screen proportional, EViews will print the objects in the spool in the same proportions as the spool display window. All sizing customization in the spool will be used.

The positions of objects on the page is controlled using the **Justification** dropdown. Choosing **Left** or **Center** will force all objects to print on the left side or the center of the page. **Center graphs only** will left-justify table and text objects, and center graph objects.

Labeling

You may use the **Object Labels** section of the dialog to control whether to print titles and comments in the spool. Selecting the respective checkbox check will print the additional information along with the output object. These settings are especially useful if you turn on the titles and comments in the spool display, but do not wish to include them in your printed output, or vice versa.

Saving a Spool

You may save the contents of a spool object to disk. Click on **Proc/Save spool to disk...** to display the save dialog.

The **File type** drop-down menu may be used to change between saving a **Comma Separated Value (*.csv)**, **Text (*.txt)**, **Rich Text Format (*.rtf)**, **Portable Document Format (*.pdf)** or **LaTex (*.tex)** file. Note that if you specify a CSV or text file, any graphs in the spool will not be written to the file.

In the top portion of the dialog, you may provide the name of the file you wish to create. EViews will automatically append an extension of the proper type to the name and will prepend the default path if an explicit path is not provided.
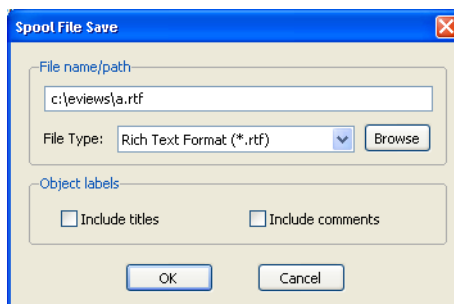
The **Object labels** section of dialog may be used to specify whether the titles and comments should be written to the output file. By default, neither titles nor comments are saved in the output file.

For LaTex output, you have the additional option to output full LaTeX document specification. Alternatively, you can disable the **Include document specification** option and the minimal amount of code will be generated for you to insert into your preexisting document.

Part IV. Extending EViews

EViews provides a number of ways to enhance its functionality outside of the built-in routines. Broadly speaking, you may:

- Create your own estimator using EViews tools.

- Add menu entries and create new object classes.

- Connect to external applications to share data, computation, and output.

These features are documented in a variety of places. We offer here a brief overview and pointers to more detailed descriptions.

Create Your Own Estimator

EViews provides two methods for creating your own estimators using built-in EViews functionality: the log-likelihood (logl) object, and the `optimize` command. (Note that you may also use the COM connection to R and MATLAB discussed below to extend EViews using any of the estimation techniques available in those packages).

The logl object provides you with a general tool for estimating a broad class of specifications by maximizing a likelihood function with respect to parameters. You may use EViews' wide range of series generation capabilities to describe the log likelihood contribution of each observation in your sample as a function of unknown parameters, and use the logl to optimize with respect to the parameters.

- For details, see Chapter 37. "The Log Likelihood (LogL) Object," beginning on page 565 of *User's Guide II*.

The `optimize` command provides tools that allow you to find the optimal parameters or control values of a general user-defined function. Notably, `optimize` supports quite general functions so that the computations and the user-defined objective need not be series-based

- For details, see Chapter 10. "User-Defined Optimization," beginning on page 243 of *Command and Programming Reference*.

Add Menu Items and Object Classes

EViews Add-ins and user objects allow complete customization of your EViews environment, allowing you to add menu items, and define new object classes.

Add-ins are simply EViews programs that are integrated into the EViews menus and command line, allowing you to execute the programs using the menus or by user-defined command. In this regard, any EViews program can be used as the basis of an Add-in.

If you do not wish to create and program your own Add-ins, you may take advantage of them by installing prepackaged Add-ins from the IHS EViews website or from third-parties. Once installed, Add-ins can provide you with user-defined features that are virtually indistinguishable from built-in EViews features.

- See Chapter 8. "Add-ins," on page 191 of *Command and Programming Reference* for details.

An EViews user object allows you to create your own object types inside of EViews. A user object may be as simple as a storage container for other EViews objects, or it may be a sophisticated new estimation object defined by multiple EViews programs, with views containing post-estimation tests and results, and procedures producing output from the estimation results.

As with Add-ins, if you do not wish to go to the trouble of creating your own objects, you may still take advantage of this powerful tool by using user objects downloaded from the IHS EViews website or obtained from third-parties.

- See Chapter 8. "Add-ins," beginning on page 191 of *Command and Programming Reference*.

Connect with External Applications

EViews provides a number of different ways to connect with external applications.

EViews OLE support, which allows you to copy and paste EViews output into OLE aware packages, such as Microsoft Word, Excel and PowerPoint, and have the output linked in such a way that any changes made in EViews are automatically pushed to the external document into which the output was pasted.

- See Chapter 18. "Object Linking and Embedding (OLE)," beginning on page 827 for details.

EViews' COM client support allows connection to both MATLAB and R. This support includes a set of EViews functions for exporting an EViews data object into the external application, running commands and programs in the application, and importing data back into EViews. These functions provide easy access to the powerful programming languages of MATLAB and R to create programs and routines that perform tasks not currently implemented in EViews. The interface also offers access to the large library of statistical routines already written in the MATLAB and R languages.

- See "EViews COM Automation Client Support (MATLAB and R)" on page 179 of *Command and Programming Reference* for discussion.

EViews may also be used as a COM Automation server so that an external program or script may launch and control EViews programmatically. The COM server classes provide access to

EViews functionality and data. Most notably, the Application class **Run** and a variety of **Get** and **Put** methods provide you with access to EViews commands and allow you to obtain read or write access to series, vectors, matrix, and scalar objects.

- See "EViews COM Automation Server" on page 179 of *Command and Programming Reference* for discussion.

For a complete description of COM methods, please refer to the *EViews COM Automation Server* whitepaper available from our website www.eviews.com/download/download.html.

The EViews Excel Add-in offers a simple interface for fetching and linking to data stored in EViews workfiles and databases from within Microsoft Excel (2000 and later) without using copy and paste.

- See "The Excel Add-in" on page 164 for further discussion.

Similarly, you may use the EViews OLEDB driver to allow OLEDB-aware clients or custom programs to read series, vector, and matrix objects directly from EViews workfiles and databases.

- See "The OLEDB Driver" on page 168 for further discussion.

Chapter 18. Object Linking and Embedding (OLE)

Users often include EViews output in external documents for presentation or publication. Like most programs, EViews offers the ability to copy-and-paste from EViews into other applications. Images and data transferred in this manner are static copies which are disconnected from the source workfile and application. If you wish to make changes in the appearance of the output in your document, you must first make the changes inside EViews and then redo the copy-and-paste. Furthermore, changes in the source within EViews do not produce corresponding changes in the external document.

Fortunately, EViews also supports the use of OLE (Object Linking and Embedding) technology. OLE allows for linking or embedding of EViews objects in third-party software application documents, including, among many others, Microsoft Word, Microsoft Excel, and Microsoft PowerPoint.

You may use OLE to paste links to EViews objects in your external document so that the underlying information is tied to the EViews workfile. Then, any time modifications are made in EViews, the changes may be pushed to the objects in your document. Alternately, you may use OLE to embed graph and table output in external documents so you may later modify the appearance of the output using EViews.

Suppose, for example, you have created a report in Microsoft Word based on EViews results using monthly data from January 2012 through June 2012. The report contains data, tables and graphs copied from EViews.

The following month, you receive data for July 2012, and update your EViews workfile sample, tables, and graphs to incorporate the new information. If you had included the results in Microsoft Word using simple copy-and-paste, you would need to redo the copy-and-paste of the EViews results. If instead you use OLE to *link* your results, your Microsoft Word document would be updated to reflect the changes with no recopying required. YOu don't even need to have your document open at the same time you are making changes in EViews. Your changes can be passed to the linked results the next time you open your Microsoft Word document.

Similarly, you might have a Microsoft Word report where periodic updating is not required, but where you make occasional changes the color of the graph lines and the text fonts. If you include the results using simple copy-and-paste, you will need to make the cosmetic changes in EViews and then redo the copy-and-paste. If instead you use OLE to *embed* the results, you could simply reopen the copy from within the external application, launching EViews to make the desired changes. The results would automatically be updated without requiring another copy-and-paste.

Embedding vs. Linking

Before continuing, it is important that we distinguish clearly between the two basic methods of using OLE to copy an EViews object to an external document:

- *Embedding* an EViews object is akin to taking a snapshot of your object and storing it in a document along with information connecting the output to the EViews application. The embedded snapshot contains the graph or table data, which can be edited to a limited degree by reopening it from within the external application. Embedded objects need not be connected to their EViews source.

- *Linking* an EViews object stores a *reference* to the information needed to rebuild the object in EViews. All data remains within EViews. Links should remain connected to their EViews source since the linked object will look to EViews and the linked data for updating when the underlying EViews data or attributes change.

The primary distinction between embedded and linked objects lies in which application owns the data. Linked objects are pointers to information that exists solely within EViews, while embedded objects are self-contained within the document. Making a choice between these methods really depends on who you consider to be the owner of the object and the types of updating that you anticipate requiring.

An object which will require periodic updating, say from new addition of data or periodic recalculation, should be pasted as a linked object. Pasting as a link fully utilizes the power of OLE by keeping the object connected to the original data.

An object whose data will not change, but might require customizing by EViews, is probably best suited to be an embedded object. The output behaves as a static picture until reopened for modification in EViews by the external application. Embedding objects means that the output is not connected to the original data, so that only a limited amount of modification is possible.

One advantage of linking objects is that your document size can be much smaller, as linked objects contain only references to the information needed to rebuild the object in EViews. The downside to linking is that a link should remain connected to its EViews source. You cannot move or delete the original workfile if you wish to have automatic updating of the object in the external document.

Using OLE

To use OLE, simply copy the contents of an EViews object window to the clipboard in the usual fashion, then paste the clipboard contents to the destination in another application using one of application supported OLE paste methods.

In the discussion to follow, we assume that you have already copied the EViews object information, and we describe how to paste into Microsoft Word, Excel, and PowerPoint. While the basics of the operations will be the same for most applications, there may be subtle differences for other applications (and versions of these applications).

The simplest approach is to select **Edit/Paste** or CTRL-V to perform paste the clipboard contents using the default program settings. In Microsoft Word or PowerPoint 2007, the default paste puts an embedded image in the destination document. Different applications will handle the default paste differently; some will simply paste a static picture of the content. (You can tell whether the paste uses OLE by double-clicking on the object in the target application. If the application launches EViews, the pasted image is not a static picture or static text.)

Since the default paste behavior varies by application, we recommend that you select **Edit/Paste Special...** from the main (or right-mouse button menu) to paste into the destination, as this will offer control over the output type. The destination application will determine which OLE paste options are available, but generally speaking, you will be offered a choice between three paste methods:

- The object may be pasted as a static picture or static text.

- The object can be pasted as an embedded EViews object.

- The object may be pasted as a linked text or image.

As discussed in "Embedding vs. Linking" on page 828, the preferred choice between these options depends on the types of modifications you anticipate performing in the future. If the object will not require additional modification, you can paste as a picture or text. If the object will only require cosmetic upgrading, you may paste as an embedded object. If the object requires full updating and access to the original workfile data, you should paste as a linked object.

In the following sections, we will describe in greater detail the behavior of selected **Paste Special** operations for copying graphs, tables, and spreadsheets into Microsoft Word, Excel, or PowerPoint. Our examples are constructed using Microsoft Office 2007.

Pasting Graphs

The methods for pasting EViews graphs into Word, Excel, and PowerPoint are very similar. We will illustrate the options and menus offered by Word for pasting a graph.

We begin by copying the graph in EViews. Simply display the EViews graph view, type CTRL-C, or right-click and select **Copy to clipboard...** to place the graph on the clipboard.

Copy to clipboard...	Ctrl+C
Options...	
Add text...	
Add lines & shading...	
Draw arrow	
Templates...	
Sort...	
Remove selected	
Save graph to disk...	
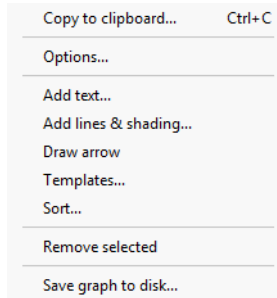
Now switch to Microsoft Word. To paste the graph, click the down arrow below the **Paste** icon in the top-left corner of the toolbar and select **Paste Special...** from the menu:
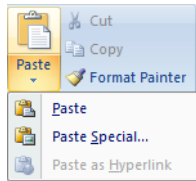
Microsoft Word will prompt you for additional options in a **Paste Special** dialog. Most importantly, you will be prompted to choose between the **Paste** (embedding or simple copy-and-paste) or **Paste link** (linking) options.

Here we see the dialog when you select **Paste,** with a listbox offering you the choice of pasting as an **EViews Object** (embedding) or as a **Picture (Enhanced Metafile)** (simple copy-and-paste):
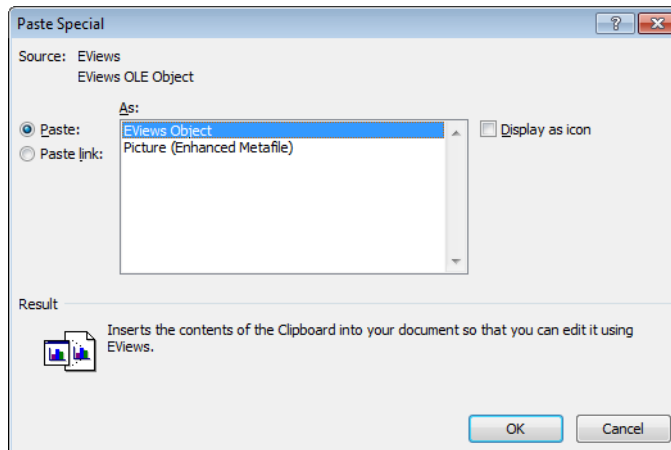
Note that since we have selected **Paste** as our method, choosing **EViews Object** means that we are embedding the object in the Word document.

Alternately, if you choose **Paste link**, the dialog offers only the choice of pasting (linking) the **EViews Object**:
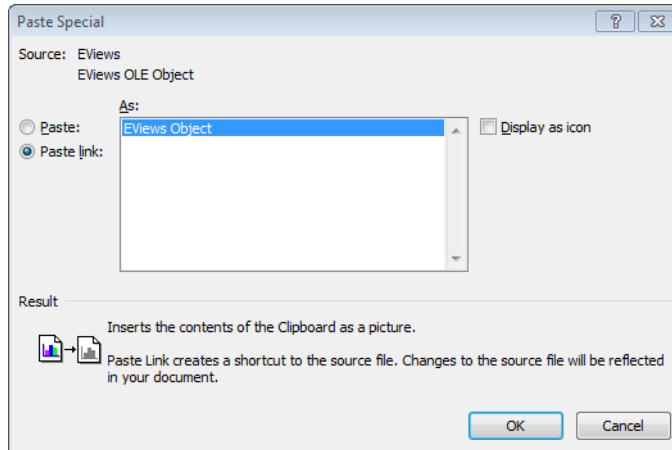
Note that since we have selected **Paste link** as our method, choosing **EViews Object** means that we are providing a link to the object in the Word document.

We discuss each of the three methods in turn.

Paste > As Picture (Enhanced Metafile)

To paste a graph as a static image, select **Paste Special...** in the main Microsoft Word menu, click on the **Paste** radio button, select **Picture (Enhanced Metafile)** from the list-box, and click on **OK**. A metafile will be inserted into your Word document. The pasted output is simply an image, which can be resized and edited only using the tools provided by Microsoft Word.
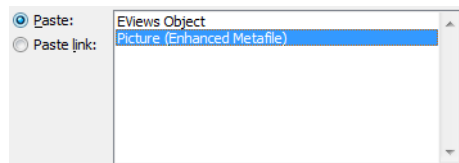
Paste > As EViews Object

To embed an EViews graph into your Word document, select **Paste Special...** in the main Microsoft Word menu, click on the **Paste** radio button, choose **EViews Object** from the listbox, and click on **OK**.
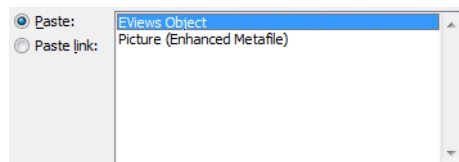
Note that since we have selected **Paste** as our method, choosing **EViews Object** means that we are embedding the object in the Word document. Your embedded graph will not change unless you double-click it and make modifications within EViews. Since it is disconnected from the original workfile and graph from which it was copied, it remains editable even if the source objects have been deleted.

The resulting pasted object initially appears very similar to the metafile. However, embedding a graph into Word produces an object similar to a frozen EViews graph. You may double-click on the graph inside the Word document to open the graph in EViews. If EViews is already open, the graph will open within the current instance. Otherwise, a new copy of EViews will be opened.

The embedded graph will open in a new graph window within EViews, with the title "EMBEDDED01". (As additional embedded graphs are opened at the same time, they will be titled "EMBEDDED02", "EMBEDDED03", etc.) Notice how the title implies that this is a temporary graph window within EViews. It is a copy of what now was embedded in Word, and is fully independent from the original graph.

From here, you can modify the graph attributes. You can set color and axis settings for the graph, turn on grid lines, change the legend text, and even sort the series.

When your edits are complete, you may
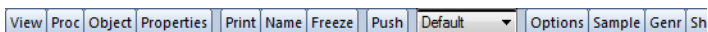
update the embedded graph in the Word document by clicking the **Push** button on the EViews graph window. Alternately, you may simply close the graph window in EViews, and you will be prompted to push the changes to the embedded graph in Word.

Note that if you push the changes, your edits are saved in the embedded graph in Word. If the original object in EViews from which the copy was made exists, it remains unchanged. This is a key difference between embedded and linked objects.

Paste Link > As EViews Object

The most powerful feature of OLE is its ability to link pasted objects to their source in EViews. To paste a graph object into Word that will remain linked to EViews, select **Paste Special...** in the main Microsoft Word menu, choose the **Paste link** radio button on

the left of the dialog, select **EViews Object** from the listbox on the right, and click **OK**. Note that your EViews object and workfile must both be named before the object may be pasted as a link. (You are simply pasting a reference back to the EViews object, and EViews must be able to identify the object from which to draw the information.)
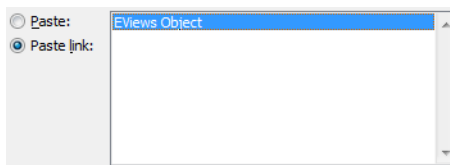
Again, the graph appears identical to one pasted using other methods. However, double-clicking on the graph brings up the actual series or graph within EViews. You may note that the title of the window is not "EMBEDDED01", as it was in the embedded case, but is the name of the EViews object in the workfile. This is not a copy of the graph; it is the actual series view or graph in EViews.

Any changes to the data, sample, or graph attributes that you make within EViews will be passed to the graph link in Word. Updates in OLE linked graphs in Word generally occur as the EViews graph is updated, but you may force an update by pressing the **Push** button at the top of the EViews graph window. Alternately, you may right-click on the graph in Word and select **Update Link**.

One fundamental difference between embedded and linked graphs is that to edit an embedded graph, you must double-click on the embedded graph to launch EViews. In contrast, a linked graph will update whenever the underlying information changes within EViews. If your Word document is closed and changes are made to the related data in EViews, the linked graph may be updated the next time the document is opened. (You will notice that when the linked Word document is closed, the **Push** button on your EViews graph is not shown since EViews cannot actively update your linked graph until it is reopened.)

It is important to note that linked objects can become "orphaned" if the original EViews workfile is deleted or moved, or the original object is deleted or renamed. If Word and EViews are both open and a linked object is renamed within EViews, Word will generally recognize this and the link will remain intact. However, if the Word document is closed at the time an object is renamed, Word will no longer know how to locate the object in EViews, and the linked object will become orphaned. You will be prompted the next time the document is opened that it cannot locate the source information. (You may, at any time, confirm the validity of your link target by double-clicking the linked graph to open the corresponding graph within EViews.)

Note that a single EViews graph can be pasted as a link in multiple documents, across various applications. This means you may include your results in both a Word document and a PowerPoint presentation, and that both linked objects will update as EViews changes.

Linked Graphs and the Graph Slider Bar

There is one subtle issue that you should keep in mind when linking graphs where you have modified the display using the sample bar.

Many EViews graph views allow you to adjust their local sample by dragging the slider bar at the bottom of the graph window. Crucially, the persistence of this view sample change depends on the type of graph you are modifying.

Recall that there are two types of graphs in EViews: a graph view and a graph object. A graph view is made by showing a graph view within a series or group object. A graph object is created by freezing a graph view of a series or group. Thus, a graph object is a frozen graph view.

When you change the sample of a graph view of a series or group object using the slider bar, the slider sample change is transitory; the sample for the view will be reset to the workfile sample when the series window is closed. In contrast, a slider modified view sample is preserved in a graph object, even after the object window is closed. This difference in persistence is relevant for link behavior since it may lead to your series or group object graph view and your linked graph to use different samples.

Let's see how the slider bar may affect linked graphs. Suppose you have created a graph object GRAPH01 by freezing a graph view of the series X. You re-adjust the sample of GRAPH01 using the slider bar, and then copy-and-paste the graph as a link into a Word document. If you return to EViews and adjust the sample of GRAPH01, the new sample will be stored within GRAPH01 and the linked graph's sample will follow.

Alternately, suppose you modify the graph view of series X by setting the slider bar to show a subset of observations. When you copy-and-paste the graph view as a link, EViews stores the current view sample as part of the link, however, since you are working with a series object and not a graph, EViews treat the slider modification as a temporary change which will be discarded when the series window is closed. When you reopen the series window,
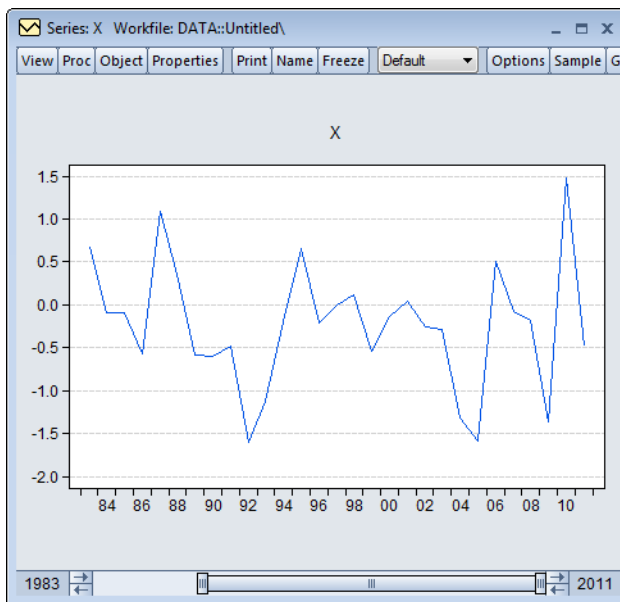
the graph view will use the workfile sample while the linked graph will use the slider sample at the time the graph was linked.

If you wish to reset the sample in the linked graph to match that of the EViews graph view, you may do so by dragging the slider bar in the series window. In fact, anytime the slider bar is used, the linked graph will adjust accordingly. Alternately, to edit or view the linked graph in EViews while preserving the sample, double-click the linked graph. This brings up the graph view of the series, with its sample set to match that of the linked graph.

Note however, that the series slider view sample will again be discarded when the window is closed so that the next time you open the series window, the samples will again be misaligned. Given the potential for confusion, we recommend that you freeze the view into a graph object and then link the graph object If you wish to use the slider sample.

Pasting Tables and Spreadsheets

The procedure for pasting EViews tables and spreadsheets into Microsoft Word, Excel, and PowerPoint is closely related to the method for graphs, described above in "Pasting Graphs," on page 829. There are, however, some unique options for pasting text and numerical values, which we focus on as we explore copy-and-paste from EViews spreadsheets and tables into other applications.

Generally speaking, EViews table and spreadsheet views used to hold two types of data: formatted information or output, and numerical data. For instance, output from an equation estimation is formatted and displayed in a table, while series spreadsheets are used to view the data contained in a series.

The type of information you are copying and what you wish to do with the copy will determine which paste method will best suit your needs:

- If the object contains formatted information and we would like to preserve this formatting, we should consider pasting as an EViews Object, Picture (Enhanced Metafile), or as Formatted Text (RTF).

- If you are pasting numerical data and precision is important (even though the full extent of the values may not be visible), you will want to paste the object as pure text. Additionally, if you are pasting into Excel and would like to include the data in Excel formulae, you should paste as text.

The nature of the application into which you are pasting also provides clues as to the best paste method. Generally, if you are creating a document in Microsoft Word or Microsoft PowerPoint, you may prefer a formatted image or formatted text, while if you are pasting into Microsoft Excel, you will likely want numerical values.

When pasting into external applications, you will typically be presented with a subset of the following **Paste Special...** options:

- **Picture (Enhanced Metafile)** (static)

- **EViews Object** (embedded or link)

- **Formatted Text (RTF)** or **Unformatted Text** (static or link)

- **Unicode Text** or **Text** (static or link)

As you can see, some of the choices have multiple forms, embedded or link, or static or link.

Pasting Formatted Information

Let's explore some of the available choices when pasting output that has been formatted by EViews. One of the more common operations involves pasting data from EViews into a Microsoft Word document in Rich Text Format (RTF). Pasting as RTF allows you to include formatted equation estimation output in your documents, such that the columns and sections created in your EViews table are recreated in the pasted object in Word.

(Our example uses Microsoft Word 2007. Your program and dialogs may vary.)

As with graphs, we must first copy the data from within EViews. When copying from a table or spreadsheet, you must choose your range of data and specify how you would like it copied. Highlight a range of text, or choose **Select all** from the right-click men to copy the entire spreadsheet. Then press CTRL-C or select **Copy** from the right-click menu.

A dialog will appear showing two methods for copying your data. If you select **Formatted**, the numbers will be copied using the display format of the table or spreadsheet. Alternately, if you select **Unformatted**, the numbers will be copied at the highest precision possible for your data. (Note that any NA's in your data will be coded as "NA" if you select **Formatted**. Otherwise, they will appear as empty values.) Click on **OK** to place the data on the clipboard.
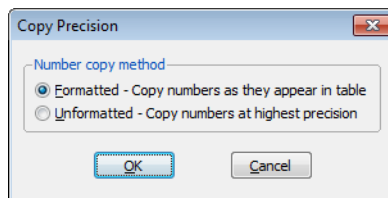
We emphasize that you do not have to copy an entire table or spreadsheet. You may select any range of cells, and copy just the selected cells. For instance, you may extract a single column from the estimation output of an equation, and paste it into Excel for presentation or numeric analysis.

You may perform a simple paste by switching to Microsoft Word and clicking the **Paste** button, selecting **Paste** from the right-click menu, or typing CTRL-V. The data will be pasted as static text using the format settings selected during copy.

Alternately, to take advantage of OLE capabilities in Microsoft Word, click the down arrow below the **Paste** icon in the top-left corner of the Word toolbar:
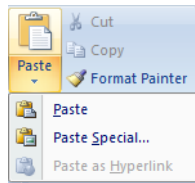
If you select **Paste Special...** from this menu, Word will display a dialog prompting you to specify a paste method. You should use the radio buttons on the left to choose between **Paste** (embedding or simple copy-and-paste) or **Paste link** (linking) the EViews object.

Here we see the dialog when you select the **Paste** radio button, with a listbox offering you the choice of pasting the **EViews Object** (embedded), **Formatted Text (RTF)** (static), **Unformatted Text** (static), or **Picture (Enhanced Metafile)** (static):
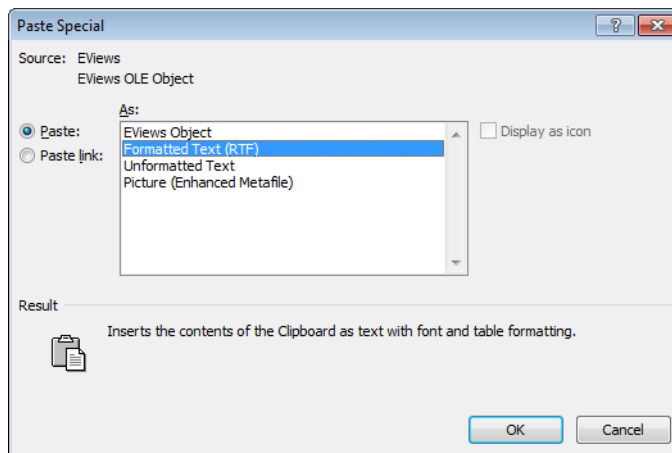
If instead you choose **Paste link**, the dialog offers you the choice of paste as a link of type **EViews Object**, **Formatted Text (RTF)**, or **Unformatted Text**:
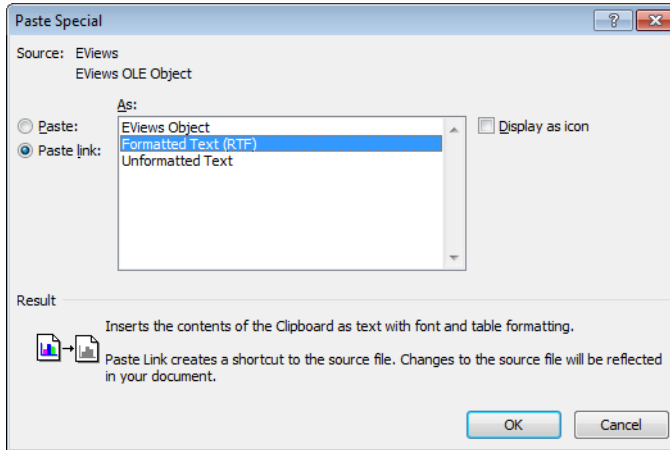
(Note that the **Unicode Text** and **Text** options are available when pasting into some applications, but not when pasting into Microsoft Word.)

Paste > As Picture (Enhanced Metafile)

To paste your text as a static image, select **Paste Special...** in the main Microsoft Word menu, click on the **Paste** radio button, select **Picture (Enhanced Metafile)** from the listbox, and click on **OK**.

A static metafile will be inserted into your Word document. Note that Even though you copied text from the table or spreadsheet, the pasted output is an image, which can be resized and edited only using the tools provided by Microsoft Word.
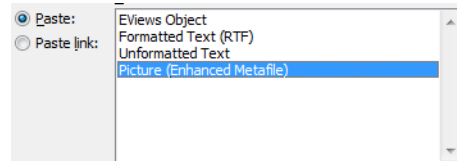
Paste > As EViews Object

To embed data from EViews into your Word document, select **Paste Special...** in the main Microsoft Word menu, click on the **Paste** radio button, choose **EViews Object** from the listbox, and click on **OK**.

Embedding the EViews object into Word allows you to double-click on the text inside the Word document to open the text for editing in a table object in EViews. If EViews is already open, the text will be opened within the current instance. Otherwise, a new copy of EViews will be launched.
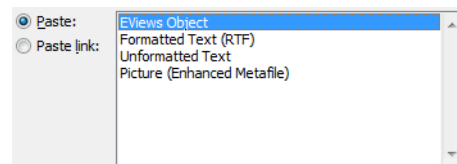
The embedded text will open in a new table window within EViews entitled "EMBED-DED01". (As additional embedded objects are opened at the same time, they will be titled "EMBEDDED02", "EMBEDDED03", etc.) This is a temporary table window within EViews used only for modifying the embedded object. It contains a copy of the embedded object view text in Word, but is fully independent from the original table or spreadsheet.

You may use EViews to modify the font and color of the text, the borders and outlines, or the data itself, as you would with any table object. You may also apply number formatting to the cells of your temporary table object.

| View | Proc | Object | Print | Push | Edit+/- | CellFmt | Grid+/- | Title | Comments+/- |
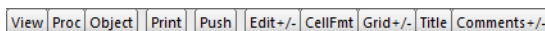
When you are done editing the table, you may update the embedded object in the Word document by clicking on the **Push** button on the EViews table window. Alternately, you may simply close the table window in EViews, and you will be prompted to send your changes to Word. If the original table or spreadsheet in EViews from which the copy was made exists, it will be unchanged.

Paste > As Formatted Text (RTF)

Formatted EViews output such as equation output is most commonly pasted into Word as **Formatted Text** using **Rich Text Format (RTF)**.

To paste a formatted table or spreadsheet from EViews into your Word document, select **Paste Special...** in the main Microsoft Word menu, click on the **Paste** radio button, choose **Formatted Text (RTF)** from the listbox, and click on **OK**.

(Note that the term "formatted" in the Word paste dialog refers to the overall settings of the object to be pasted, such as text size, fonts, column widths, and section specifications. This setting is unrelated of the use of the terms "formatted" or "unformatted" in the EViews copy dialog which refer to the format and precision of the *numbers* to be copied.)
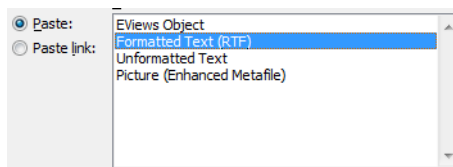
Here, we have pasted EViews estimation output from an equation. You can see that the table sections, column widths, and formatting have been preserved.

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 10/19/97   Time: 22:43
Sample(adjusted): 1952:2 1992:4
Included observations: 163 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.312383	0.032199	40.75850	0.0000
LOG(GDP)	0.772035	0.006537	118.1092	0.0000
RS	-0.020686	0.002516	-8.221196	0.0000
DLOG(PR)	-2.572204	0.942556	-2.728967	0.0071
R-squared	0.993274	Mean dependent var		5.692279
Adjusted R-squared	0.993147	S.D. dependent var		0.670253
S.E. of regression	0.055485	Akaike info criterion		-2.921176
Sum squared resid	0.489494	Schwarz criterion		-2.845256
Log likelihood	242.0759	Hannan-Quinn criter.		0.140967
F-statistic	0.000000			

Paste > As Unformatted Text

To paste unformatted table or spreadsheet data into your Word document, select **Paste Special...** in the main Microsoft Word menu, click on the **Paste** radio button, choose **Unformatted Text** from the listbox, and click on **OK**.

This paste method is pretty straightforward. The text is pasted without column or cell specifications, and without any other formatting EViews may have applied to the table. This can be useful if you have copied from a formatted table or spreadsheet in EViews, but you do not want to create a table in your Word document. Use this option to paste just the text itself.
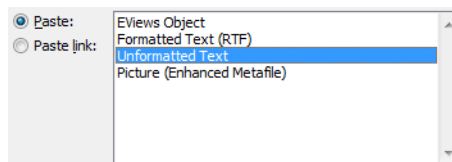
Again, note that the use of the term "unformatted" in the paste dialog refers to table formatting. The choice you made in the EViews dialog to copy "formatted" or "unformatted" text pertains to the precision of the numbers, and it will be preserved regardless of your choice of paste method.

Paste Link > As EViews Object

To paste data into Word that will remain linked to EViews, select **Paste Special...** in the main Microsoft Word menu, choose the **Paste link** radio button on the left of the dialog, select **EViews Object** from the listbox on the right, and click **OK**. Bear in mind that your EViews workfile and object must be named before the object can be pasted as a link.

Once you have pasted table or spreadsheet data into Word as a link, you can double-click on the link in your document to bring up the actual object within EViews. You may note that the title of the window is not "EMBEDDED01", as it was in the embedded case, but is the name of the EViews object in the workfile. This is not a copy of the table or spreadsheet; it is the actual object in EViews.
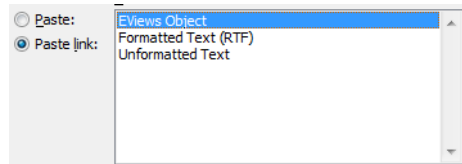
As with the embedded object, you may cosmetic edits to the table or spreadsheet, changing fonts, text size, *etc.* However, unlike embedding, linking allows you to make modifications to the actual object so that you may make substantive changes like changing numeric values in your tables or spreadsheets. Furthermore, since links are actively tied to the data in EViews, if you have copied the formatted version of your data, the precision of the numbers in a link will update if you modify the numerical formatting within EViews.

Generally, a linked object will update when the underlying data changes within EViews. You may force an update to a linked table or spreadsheet in an open Word document by pressing the **Push** button at the top of the EViews object window. Alternately, you may right-click on the linked object in Word and select **Update Link**. If your Word document is closed and the linked object can be updated the next time the document is opened.

As we mentioned for graphs, linked objects can become "orphaned" if the original EViews workfile is deleted or moved, or the original object is deleted or renamed. If Word and EViews are both open and a linked object is renamed within EViews, Word will generally recognize this and the link will remain intact. However, if the Word document is closed at the time an object is renamed, Word will no longer know how to locate the object in EViews, and the linked object will become orphaned. You will be prompted the next time the document is opened that it cannot locate the source information.

Note that a single EViews table or spreadsheet can be pasted as a link in multiple documents, across various applications. This flexibility means you may include your results in both a Word document and a PowerPoint presentation, and that both linked objects will update as the underlying EViews data changes.

Paste Link > As Formatted Text (RTF)

You may link formatted text by pasting as **Formatted Text**, or in **Rich Text Format (RTF)**. Pasting as a link offers us the additional advantage that as the data are updated in EViews, it may also be updated in your Word document. Furthermore, if you have copied formatted data, the display precision of the numbers in a linked object will follow the formatting specified in EViews.

To paste a link to a table or spreadsheet from EViews into your Word document, select **Paste Special...** in the main Microsoft Word menu, click on the **Paste link** radio button, choose **Formatted Text (RTF)** from the list-box, and click on **OK**. Again, remember that your EViews workfile and object must be named before the object can be pasted as a link.
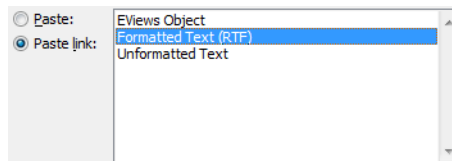
Here we have selected the output from an equation estimation in EViews, copied it in **Formatted** form, and pasted as linked **Formatted Text (RTF)** into a Word document.

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 10/19/97   Time: 22:48
Sample(adjusted): 1952:3 1992:4
Included observations: 162 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.071297	0.028248	2.523949	0.0126
LOG(GDP)	0.320338	0.118186	2.710453	0.0075
RS	-0.005222	0.001469	-3.554801	0.0005
DLOG(PR)	0.038615	0.341619	0.113036	0.9101
LOG(M1(-1))	0.926640	0.020319	45.60375	0.0000
LOG(GDP(-1))	-0.257364	0.123264	-2.087910	0.0385
RS(-1)	0.002604	0.001574	1.654429	0.1001
DLOG(PR(-1))	-0.071650	0.347403	-0.206246	0.8369

R-squared	0.999604	Mean dependent var		5.697490
Adjusted R-squared	0.999586	S.D. dependent var		0.669011
S.E. of regression	0.013611	Akaike info criterion		-5.707729
Sum squared resid	0.028531	Schwarz criterion		-5.555255
Log likelihood	470.3261	Hannan-Quinn criter.		2.393764
F-statistic	0.000000			

Just as with any linked information, formatted text will be updated as new data arrive in EViews. To update the linked estimation results, click within the text in Word, right-click and select **Update Link**. Alternately, press the **Push** button at the top of the EViews equation window. New results will be displayed in your Word document without having to re-copy and re-paste the data from EViews.

To open the link in EViews, double click on the object, or click on the object and select **Linked EViews Object** and **Open Link** from the right-click menu.

Paste Link > As Unformatted Text

To paste an unformatted table or spread-
sheet link into your Word document, select
Paste Special... in the main Microsoft Word
menu, click on the **Paste link** radio button,
choose **Unformatted Text** from the listbox,
and click on **OK**. We reiterate, your EViews
workfile and object must be named before pasting a link.
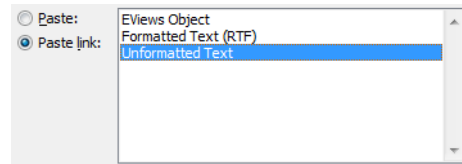
Just as for a non-link paste of unformatted text, the table or spreadsheet will be pasted with-
out column or cell specifications, and without any other formatting that EViews may have
applied to the object. You may want to use this method if you'd like to paste a link to the
pure text, without creating a table in your Word document.

Pasting Numerical Data

When pasting table or spreadsheet results from EViews into another application for numeri-
cal analysis, you are likely to be concerned with the underlying numeric values of your data
independent of their current display format.

In this section, we will illustrate how to paste spreadsheet data into Microsoft Excel, as this
is the most common target application for pasting numerical data. We employ Excel 2007 in
our examples; your dialogs may differ for alternate applications.

Just as with other paste methods, we must first copy the data from within EViews. When
copying a spreadsheet, you must choose the range of data and the format in which to copy
numeric data. To select a range of cells to copy, simply highlight them in the spreadsheet. To
copy the entire spreadsheet, choose **Select all** from the right-click menu. Then press CTRL-C
or select **Copy** from the right-click menu.

A dialog will appear showing two methods for
copying your data. If you select **Formatted**, the
numbers will be copied using the display format
currently showing in the spreadsheet or table
view. You may, for example, have specified fixed
format showing two decimal places. Formatted
copying will only capture these two decimal
places of precision. Alternately, if you select
Unformatted, the numbers will be copied at the highest precision for your data. Any NA's in
your data will be coded as "NA" if you select **Formatted**. Otherwise, they will appear as
empty values.
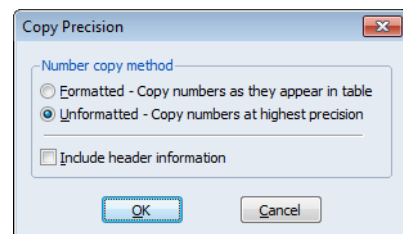
Additionally, if you are copying cells from a spreadsheet, you may choose to **Include header
information**. This will also copy the column and row labels for the selected range of cells.

Generally, these are the observation numbers and possibly the series names, if you are copying from a group. Click **OK** to place the spreadsheet data on the clipboard.

Remember that you do not have to copy the entire table or spreadsheet. You may select any range of cells, and paste them as an embedded or linked object. Note that when you copy a portion of the data from a series or group spreadsheet, the workfile sample becomes relevant. See "Pasting Links and the Workfile Sample" on page 850 for a discussion of how samples are managed in linked data.

Once copied, you can import the information into your document by switching to the application and clicking the **Paste** button, selecting **Paste** from the right-click menu, or by typing CTRL-V. This will paste the data using the default method using the format settings selected during copy.

Alternately, you may use wish to have better control of the paste method. Click the down arrow below the **Paste** icon in the top-left corner of the Microsoft Excel toolbar. You may select **Paste Special...** from the menu to bring up the main dialog, or you can choose one of the shortcut items shown directly on the **Paste** menu (if visible).

If you select **Paste Special...** from the **Paste** menu or from the right-click menu you will see a dialog very similar to the one we saw when pasting graphs and formatted text. You should use the radio buttons on the left to choose between **Paste** (embedding or simple copy-and-paste) or **Paste link** (linking) the EViews object.
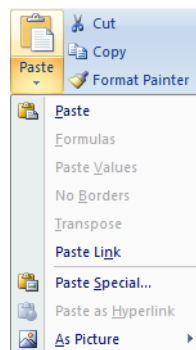
Here we see the dialog when you select the **Paste** radio button, with a listbox offering you the choice of pasting the **EViews Object** (embedded), **Picture (Enhanced Metafile)** (static), **Unicode Text** (static), or **Text** (static):
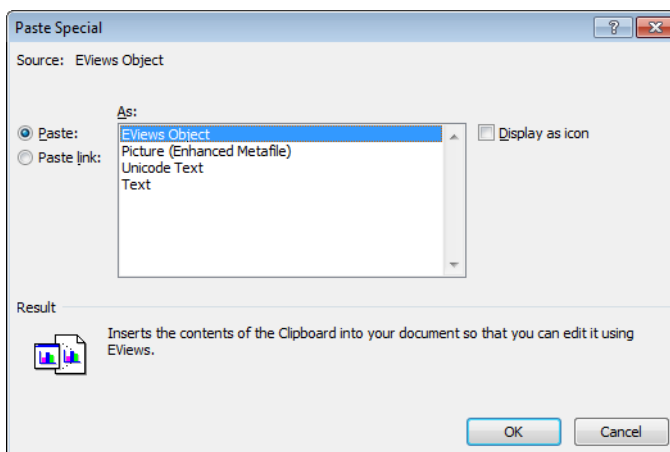
Alternately, if you choose **Paste link**, the dialog offers you the choice to paste as a link of the same types, **EViews Object**, **Picture (Enhanced Metafile)**, **Unicode Text**, or **Text**:
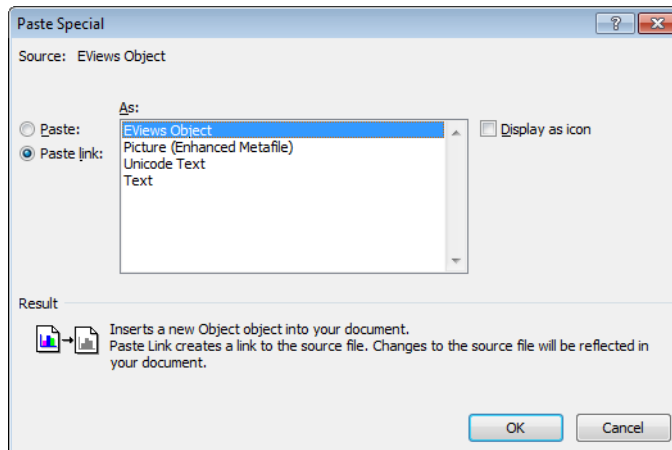
(Note that the **Formatted Text (RTF)** and **Unformatted Text** options are not presented when pasting into Microsoft Excel. In addition EViews does not allow links on some statistical views that require more information than can be described by a link formula. See "Modifying the Excel Array Formula" on page 851.)

Since we intend to treat our data as numbers in Excel, we would like to view each cell as a numeric value that can be manipulated using formulae. In this setting, the **Paste Special** options which import data an image or a single block of data do not meet our needs. While we describe all of the choices below, you will probably most often use the options **Paste As Text** or **Paste As Link Text** when linking to Excel.

The usual considerations will govern the appropriate choice when linking into PowerPoint or Word.

Paste > As Picture (Enhanced Metafile)

To paste data as a static image, select **Paste Special...** in the main Microsoft Excel menu, click on the **Paste** radio button, choose **Picture (Enhanced Metafile)** from the listbox, and click on **OK**.
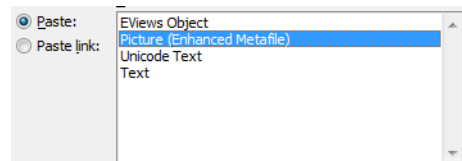
A metafile will be inserted into your Excel document. The static image can only be resized and edited using the tools within Excel. It is not connected to EViews in any way.

You may notice that the resulting image is not well formatted within Excel. The cells in the data do not correspond to Excel cells, and since we copied the numbers at full precision, the decimal points are not aligned. As a picture, the results can be stretched, resized, and rotated to any angle. Accordingly, we don't imagine you will use this paste option for Excel very often. However, if you are pasting into Microsoft PowerPoint or Word, this may be a good way to get a static image of your data into your document.

Paste > As EViews Object

To embed numeric data from an EViews spreadsheet into your Excel document, select **Paste Special...** in the main Microsoft Excel menu and the **Paste** radio button on the left-hand side of the dialog. Select **EViews Object** from the listbox and click **OK**.
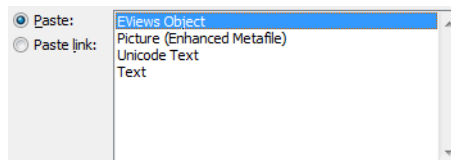
The embedded data looks very similar to the picture we pasted in the previous section. However, unlike the picture, this embedded object is similar to a frozen table and can be edited within EViews, notably to modify the formatting of the contents.

To open the table in EViews, double-click on the object within Excel. If EViews is already open, the table will open within the current instance. Otherwise, a new copy of EViews will be opened.

The embedded object opens in a new table window within EViews, with the title "EMBED-DED01". (As additional embedded tables are opened at the same time, they will be titled "EMBEDDED02", "EMBEDDED03", etc.) The title reflects that this is a temporary table window within EViews. It is simply an image of what now exists in Excel, and will not be affected if the source workfile or data are moved or deleted.

We may use standard EViews tools to modify the formatting of the cells in the table object. From the EMBEDDED01 window in EViews, you can edit the data and push the results back into Excel by clicking the **Push** button on the EViews table window. Alternately, you may simply close the table window in EViews, and you will be prompted to send your changes to the embedded table in Excel.

Notice that your changes are saved in the table within Excel, but the original data in EViews remains unchanged.

Paste > As Text and Paste > As Unicode Text

If you wish to treat your pasted data as numerical values that can be placed into individual Excel cells and optionally be used in formulae, you will want to paste as text.

To paste as text, select **Paste Special...** in the main Microsoft Excel menu and click on the **Paste** radio button on the left-hand side of the resulting dialog. Select **Unicode Text** or **Text** from the listbox. You will want to select **Unicode Text** if your data contains a special character set, or will be read on a foreign language machine which may use a special set of characters. Click on the **OK** button to paste your data as text into your Excel file.
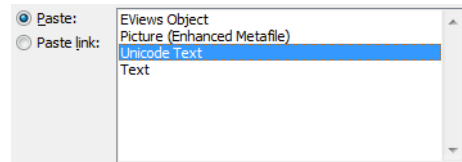
(Note that this static **Paste As Text** mimics the default Excel paste operation. You can perform this operation by selecting the **Paste** button or pressing CTRL-V. However, you should use **Paste Special...** to specify a paste of **Unicode Text**.)

Paste Link > As Picture (Enhanced Metafile)

If you wish to insert your table as a linked image, select **Paste Special...** in the main Microsoft Excel menu and choose the **Paste link** radio button on the left-hand side of the dialog. Select **Picture (Enhanced Metafile)**
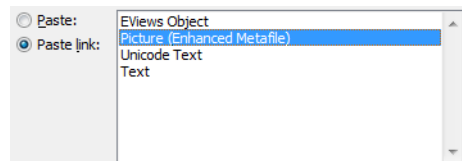
from the listbox and click **OK**. If the link option is grayed out in the dialog, remember that the EViews workfile and the object you copied from must be named before pasting as a link.

(If all you get is an empty object after using this paste method in Excel, see "Error when pasting links into Excel" on page 851.)

The resulting linked image may be resized and moved throughout the document, but its cells do not correspond to Excel cells. This is not the ideal paste method if you wish to treat your data as numbers which can be understood by Excel and optionally used in Excel formulae.

This approach does, however, still provide a link to the EViews data and the linked image will reflect changes in EViews upon update. If you are creating a Microsoft PowerPoint or Word document without cells or equations, this may be a good paste method.

As a side note, you may notice that **Select All** (CTRL-A) in Excel does not select these types of images. You need to select embedded images individually to perform operations such as move or delete.

Paste Link > As EViews Object

To paste your data as a linked EViews object, select **Paste Special...** in the main Microsoft Excel menu, click on the **Paste link** radio button, choose **EViews Object** from the listbox, and click on **OK**. If you do not see a link option in the dialog, remember that the EViews workfile and object you copied from must be named to paste as a link.
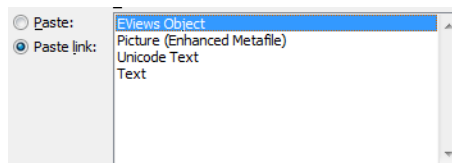
(If all you get is an empty object after using this paste method in Excel, see "Error when pasting links into Excel" on page 851.)

Pasting a table or spreadsheet link as an **EViews Object** behaves just like pasting a graph of this type. An image of the data is placed in your file, and you can double-click on the object to open the source in EViews. Edits will be passed back to the linked object upon update or by clicking the **Push** button in the EViews window. For detailed discussion, see the explanation for graphs, in "Paste Link > As EViews Object".

Paste Link > As Text and Paste Link > As Unicode Text

The real power of OLE is seen when placing numerical data into Excel lies in linking text. When you paste data values into your document, and retain the connection to their source in EViews by pasting as a link, you take advantage of the true ease in updating your data. Not only are the values available to be used by Excel, they are self-updating and can incorporate all changes that occur within EViews. Accordingly, you can build formulae using the standard Excel syntax whose output will change based on updates to data and results obtained in EViews.

To paste your data as linked text, select **Paste Special...** in the main Microsoft Excel menu, click on the **Paste link** radio button, and choose **Unicode Text** or **Text** from the listbox. You will want to select **Unicode Text** if your data contains a special character set, or will be read on a foreign language machine which may use a special set of characters. Click on the **OK** button to paste linked data into your Excel file.
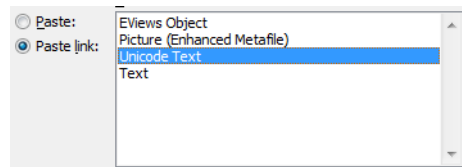
If the **Paste link** radio button is grayed out, remember that the EViews workfile and the object you copied from must be named before you can paste a link.

Just as with linked graphs, your linked data will update with changes to the values within EViews. You may force an update to the linked data by clicking the **Push** button on the EViews series or group window. Alternately, you may simply close the window in EViews to send your changes to the linked object in Excel.
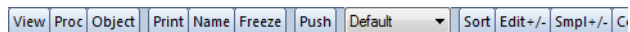
It is important to recognize that linked objects are unidirectional. That is, they receive updates from EViews but are simply display containers within Excel.

The fundamental difference between working with linked graphs and linked text is that you do not double-click your table data within Excel to view it in EViews. Instead, select any cell in your linked table, go to the **Data** menu tab in Excel, and click **Edit Links**. (We have described the operation in Excel 2007; other versions may vary significantly.) A dialog appears, showing information for all links in your document. To open a link in EViews, select the link from the list and click the **Open Source** button on the right-hand side of the dialog.

To open the EViews source of linked text from Word, highlight a portion of the text, right-click to get a menu, and select **Linked EViews Object**, and then **Open Link**.

To see the source of your links in PowerPoint, select the text, right-click, and select **Linked EViews Object** and then **Open** from the menu.

Again, we have described the process in Office 2007, and your version may offer different menu options.
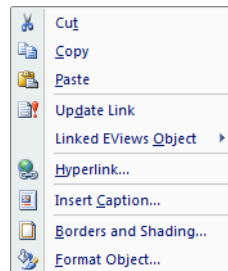
You may notice that when you have a cell selected within your linked text, Excel displays a string in the edit field at the top of the document. This is the array formula used to define the link, and is discussed in "Modifying the Excel Array Formula" on page 851.

Pasting Links and the Workfile Sample

The EViews workfile sample determines which series and group observations are included in a spreadsheet display. You may change the sample to look at different sections of your series and groups. When you copy-and-paste your data it as a link into another application, you would like the link to consistently point to the information that was copied, regardless of the sample settings inside EViews.

In nearly all cases, EViews smoothly handles the linking of spreadsheet data in the face of changing sample settings. We discuss the intuitive nature of how links and samples are managed, and we will point out one issue that occurs when you contract your workfile.

Recall that an EViews series or group spreadsheet has two approaches to displaying data with respect to the workfile sample. First, the spreadsheet may display values for every observation, whether the observation is included in the current workfile sample or not. Alternately, it can display only those observations which are included in the current sample. You may toggle between these two display modes by clicking the **Smpl +/-** button at the top of the EViews spreadsheet window.

When copying data, you should be mindful of the current workfile sample, as well as the sample of observations displayed in your spreadsheet. Crucially, when you copy cells in a spreadsheet, you are copying a range of cells in the current sample of observations in view. Both the range and the sample of observations in view at the time the data are copied are stored in the link. In this way, your link will remain connected to the proper values, regardless of sample changes within EViews.

Suppose, for example, that you have an annual workfile covering the years 1951 to 2000, and a series SER1. If you copy a spreadsheet display of all 50 observations in SER1 and paste as a link, the information stored with the link will indicate a range of 50 cells and a sample

from 1951 to 2000. (This link information is called the array formula, and is described in: .)

If you return to EViews and set the sample to include only the 25 observations from 1951 to 1975, your linked data are unchanged since the link still references the data for all 50 observations contained in the workfile range.

With the workfile sample change, you may now look at either the 25 observations in SER1 contained in the current sample, or all 50 observations in the workfile range, by toggling the **Smpl +/-** button. When you copy data from SER1, you should be mindful of which observations are being displayed. The link will behave properly when copied from either view, but the embedded range and sample are based on the current display, which will be based on either the workfile range or the workfile sample. (You can see the difference when looking at the array formula.)

EViews matches and updates the linked data seamlessly after sample changes, and this topic should not affect you unless you remove data within the cell range of the link sample. If you contract your workfile so that observations within the link sample are removed, the link cell range may lose the ability to reference the proper location of data within the workfile.

If, for example, you copied data for SER1 from 1960 to 1969, you were copying cells 10–19 from the spreadsheet display of the observations in the workfile sample or range. Then contracting the workfile by removing observations from 1951 to 1955 will lead to invalid linking. Alternately, removing observations from the 1991-2000 does not affect the validity of the link.

We urge you to recreate your links after workfile contraction, especially when data to be removed are not at the end of the workfile.

Error when pasting links into Excel

You may notice that sometimes when you paste into Excel, your object appears as an empty selection. Excel does not always handle the link paste operation correctly, particularly if you paste the same data as a link multiple times.

If this occurs, you should delete the link, recopy your data in EViews, and try the paste again. If you cannot clear the error in Excel, you should delete the link and restart Excel. Note that you must delete the item with the incorrect content; restarting Excel alone is not enough to clear the error.

Modifying the Excel Array Formula

When you click on a cell in Excel that is part of a linked table, you will see a formula in the edit field that provides Excel with the information it needs to link the object back to EViews. This expression is called the array formula. Embedded and linked objects may be visually

distinguished by observing whether an array formula is present. Linked objects contain only this information, while embedded objects contain the actual graph or text.

Because links must be completely represented by this text string, we do not allow links on some statistical views that require more information than can be described by the array formula alone.

You may notice that Excel complains when you click in the array formula, and then try to click elsewhere in the document. This is because Excel is attempting to edit the formula by including information about the cell you clicked on. Simply hit escape (ESC) to exit the formula edit.

The array formula has up to three parts, which are separated by exclamation marks ('!'). The expression begins with the full path name of the workfile. This is followed by the workfile page name (optional), the object name, and the object type (optional).

> workfile_name!(page_name\)object_name(\object_type)!view = viewnum;
> range = [A1:B2];smpl = [obs1 obs2];…;

The last part of the formula provides information about the state of the object at the time it was copied. This may include the sample, the current view, and labeling or header options that were set in the spreadsheet when it was copied. Each option in this section is followed by a semicolon (';').

Following is a list of options that EViews may code into the spreadsheet formula:

smpl = [*obs1 obs2*]	Sample specification for the data included in the link
view = *n*	EViews internal view number, indicating which view was active at the time of copy (spreadsheet view, graph view, etc.)
range = [*A1:B2*]	Range of spreadsheet cells included in the link
formatted	Specifies the copy precision: formatted or unformatted (set in the copy dialog)
mode = *arg*	Specifies the spreadsheet mode, where *arg* consists of a comma separates list of the following keys: no label ("nl"), compare ("c"), wide ("w"), transpose ("t"). Note that "w" is only for series spreadsheets.
headers	Indicates if the spreadsheet headers are visible in the link (set in the copy dialog)
sort = [*x1, x2(, x3 …)*]	Stores the sort order of the series, where descending series are represented by a negative value
transform	Indicates if there was a data transform on the series at the time of copy

Since the options above are coded into the link when the spreadsheet is copied, they will be preserved in the link object, even when the underlying EViews object is modified. For instance, if a spreadsheet is in compare mode when it is copied, the link will reflect this setting, even if the mode of the original spreadsheet is changed.

While the full array formula can be tedious to specify from scratch, you can modify an existing link by changing a few key elements. In this manner, you may be able to redo a copy-and-paste operation by editing the array formula.

After editing the array formula, you must always hit CTRL-SHIFT-ENTER to complete your edits.

If you world like to link to a different series in your workfile, you may edit the object name parameter, found in the second section of the formula. For example, for a simple series link to a series X on page MYPAGE of a workfile named "data.WF1", we have the array formula:

C:\data.wf1!mypage\X\series!view = 0;range = [B2:H3];smpl = [1974 2011];

Notice that the range element in the formula indicates that we had selected the cell range "B2:H3" before copying the series. If you'd like to link to the same range in another series in your workfile, say series Y, simply change the "X" to a "Y".

Another straightforward modification of an array formula would be to redefine the sample. In our example, the last item in the array formula specifies that the link uses a sample from 1974 to 2011. To change the sample in your linked series, simply modify these values.

Extending the Excel Cell Range

When you copy data from a spreadsheet view, the array formula will contain range information. If you would like to expand the range in your link you must extend the cell range within Excel along with making adjustments to the array formula.

For example, say you have data in your workfile for 2001 through 2010. You have selected the data for this range from one of your series and have pasted it as a link into Excel, into cells "A1:A10". If you receive new data for 2011, you might expand your workfile and would like to include the new data in your link. However, when you pasted a link of ten cells, Excel created an object with a cell size of ten. You must instruct Excel to expand the link size to eleven cells.

To do this, select the larger range, where the newest cell will still be empty. In our case, we select "A1:A11". Then modify the array formula such that the sample includes 2011 and the range reflects the additional cell in your EViews series. Hit CTRL-SHIFT-ENTER to complete your edits and to add a cell to the Excel link.

Note that you cannot shrink the range of cells in Excel. If your sample is shortened in EViews, the extra cells in Excel will be filled with zeros. To reset the range of cells, you must copy and paste the link into Excel again.

Copy Special for Spreadsheets

The preceding discussion offers a variety of choices for how you would like to select and paste your data. If you would simply like to copy an entire series or group and create a link in your spreadsheet, EViews provides the **Copy Special** shortcut. This method is useful when you are not concerned with selecting specific cells or the current object view.

Since **Copy Special** does not capture sample or view information, the operation should be done by copying the object directly from the workfile window. Right-click on a series or group and select **Copy Special...**.

A dialog will appear offering you a choice to copy the object name or the object data. **Copy object name** will only copy the text of the series or group *name*. This method is useful if you are creating a list of object names. The more commonly useful choice is to select **Copy object data**. This copies the data within the object, filling the clipboard with information you can then paste as a link into your spreadsheet. In this manner, you may copy object's data without having to open a spreadsheet view and select a range of cells.
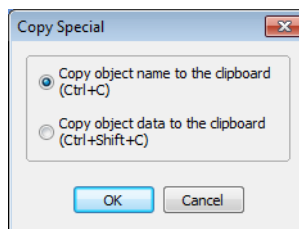
You can avoid the dialog by typing CTRL-C or CTRL-SHIFT-C, with the object selected in the workfile window.

(Note that **Copy Special** (CTRL-SHIFT-C) only works with a single series or group. However, there is a method by which you can create links to multiple series with a single **Copy Special** operation. This involves editing the array formula, as described in "Modifying the Excel Array Formula," beginning on page 851.)

Suppose you have a workfile with the three series X, Y, and Z, to which you'd like to create links. You could create a group containing the series and do a **Copy Special** on the group. This operation allows you to paste a link of the data with a single **Copy Special** and **Paste Special** operation.

However, you might prefer not to create a group or you might like each series to be its own link so the links can be placed throughout your spreadsheet instead of as one link object.

To accomplish this goal, first select any one of your desired series, do a **Copy Special**, then paste as a text link into your spreadsheet (**Paste Special, Paste link as: Text**). When you click within the linked data, you will see the link information in the array formula at the top of the Excel document. You will notice that the name of the series you just pasted can be found toward the end of this text string. In our case, you would see an 'X'.

To create another link, you will copy the text of the array formula and paste it into the appropriate cells. This is a simple operation, however it must be done precisely, as Excel is very specific about what is required to interpret array formulas.

First, click anywhere within the data of the first series to show the array formula in the edit field at the top of the document. Select and copy the formula text (CTRL-C), then hit escape (ESC).

Move to the desired location for your next series, and select the appropriate range of cells. Paste the array formula into the edit field at the top, edit it such that it contains the proper series name, and click CTRL-SHIFT-ENTER. It is very important that you do not simply exit the edit field, but that you click CTRL-SHIFT-ENTER so Excel receives your edits. In our case, we would change the 'X' to 'Y'. The second series is now fully linked, and data from EViews should appear in the cells you selected.

You may copy the array formula throughout your document in this manner, replicating your series links without needing to do multiple copy-pastes.

Note that if you use this method to create links for objects of different sizes (*i.e.*, use what was originally a series link to create a group link), you must be particularly careful to select the appropriate number and configuration of cells for the target object before pasting the array formula.

Opening Documents with Linked Objects

When you open a Word document containing linked objects, you will be prompted to update the file.
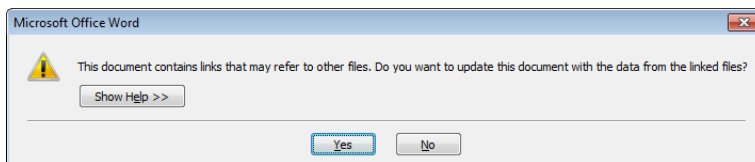
Note that different applications handle this operation differently; some applications will update automatically on open, without a prompt.

If you are prompted to update, it does not imply that the linked object has changed within EViews. It is simply a notification that the document relies on data from EViews. If you choose not to update the file, the linked object will be made inactive during that session, and it will not receive updates from EViews. This means that your EViews graph and your linked graph can potentially be in different states.

To reactivate the link, right-click the graph in Word and select **Update Link**. You may also choose to update the next time you open the Word document. Note that for some applica-

tions, pressing the **Push** button in the EViews window (if it appears) can also reactivate the linked graph. However, this should not be relied upon. In some cases the **Push** button will be visible even though the linked object is inactive. You should always choose to update links when you open the document, if you intend for the links to be active.

When opening a PowerPoint document with links, you will see an ominous looking dialog, warning of security concerns.
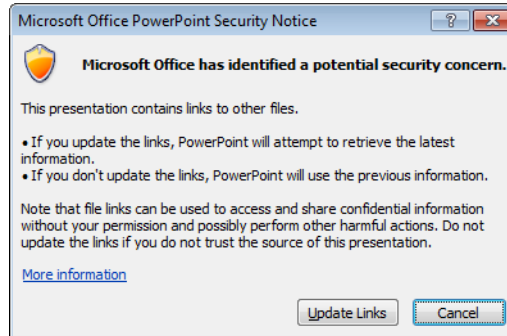
If you are confident the only links contained in the file are EViews graphs and tables, this should be harmless. Choose to open the document and update the links, or select Cancel to open without linking for this instance.

The easiest security warning to miss may be Excel's. Be aware that when you open an Excel document containing a linked table, you may not be seeing the updated data. You will instead see a security warning at the top of your document.
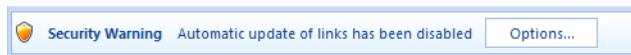
If this warning is visible, select **Options...** next to the warning text to bring up a dialog that could be similar to this one.
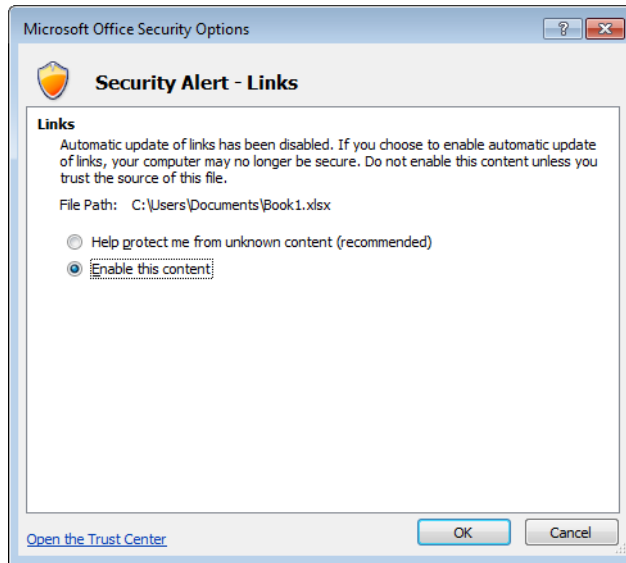
If the only links within your document are from EViews, select **Enable this content**, and click **OK**.

You can also tell when your Excel file is not in active link mode by looking at the associated series or group window. If the **Push** button is not visible, this can be an indication that the linked file is closed or is in inactive mode.

My External Application is Not Responding

If your application seems to be hung up, check EViews for a dialog awaiting your response. When you have embedded objects open in both EViews and another application, one may be waiting for a response before sending control back to the other. This state often occurs when you close a file with an embedded object that has not been updated and you attempt to save. EViews asks if you would like to update the object before saving. EViews will wait for your response, with this dialog open, before allowing the other application to close.

Appendix A. Global Options

EViews employs user-specified default settings in many operations. You may, for example, set defaults for everything from how to perform frequency conversion between workfile pages of different frequency, to which font to use in table output, to line color and thickness in graphs, to how to compute derivatives in nonlinear estimation routines.

These default options may, of course, be overridden when actually performing an operation. For example, you may have specified a default conversion of data from monthly to quarterly data by averaging observations, but may choose to use summing when performing a specific conversion. Similarly, you may instruct EViews to use the color red for the first line in newly created graphs, and then change the color in a specific graph.

The Options Menu

The **Options** menu in the main toolbar allows you to define the default behavior for many of the operations in EViews. Most options are found within the **General Options...** and **Graphics Defaults...** menu items.

General Options...
Graphics Defaults...
Database Registry...
Cloud Management...
EViews Auto-Update from Web ▸
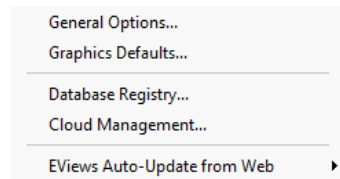
General Options

When you click the **General Options...** menu item, EViews brings up the **Global Options** dialog. This dialog is broken up into many sections which are accessible using the tree on the left-hand side of the dialog. We discuss briefly the options within each of these sections. In some cases, additional detail is provided in the corresponding sections of the manual.

Startup

The startup section controls the display of the startup page, and the handling of messages.
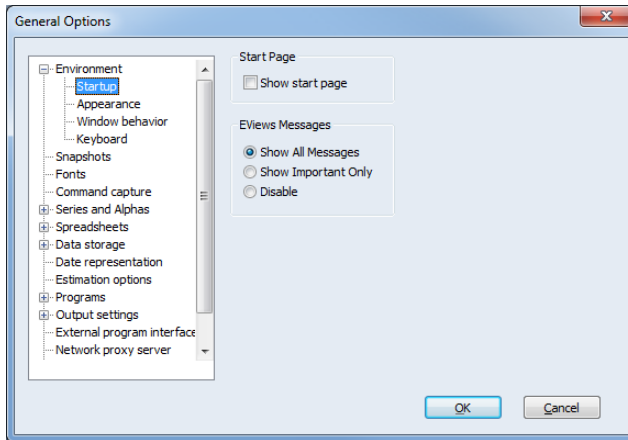
Environment Appearance

EViews provides a number of color themes to modify the appearance of your EViews environment.

In the **Window colors** groupbox, you may select from a variety of basic themes, including **Light Blue**, **Silver**, and **EViews Classic**. You can completely customize the color by selecting **Custom**.

You can let the background and status bar follow this color theme, or choose a different color from the **Background & status bar** dropdown.
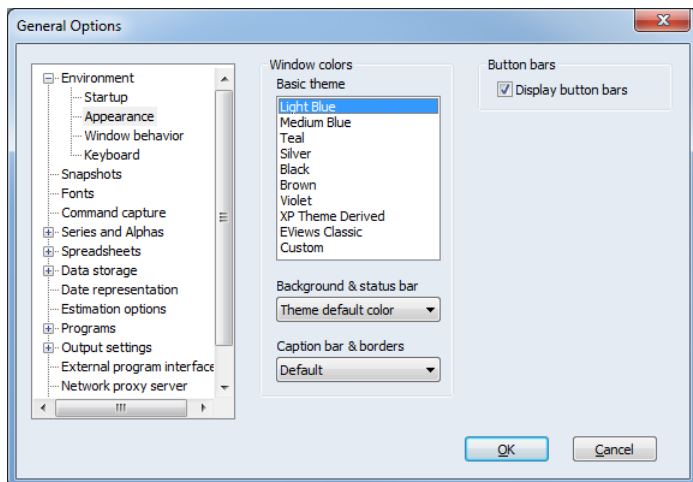
Not surprisingly, the **Caption bar & borders** dropdown may be used to choose the visual style of your caption bar and borders. The **Default** setting is for EViews to draw the caption bars and borders using colors in a style to match your operating system (either Windows XP or Windows Vista/Windows 7). You may instead choose **EViews rounded** or **Vista style** to have EViews choose a style independent of your operating system. If you have modified

your operating system settings for window borders and text, and wish to have the operating system draw the caption and borders, you should select **Operating system drawn.**

Most windows in EViews have a row of buttons along the top, called the button bar. To hide these buttons on all windows, uncheck **Display button bars** on the top right of the dialog.

Window Behavior

The **Window behavior** page controls the display characteristics of various types of EViews output.
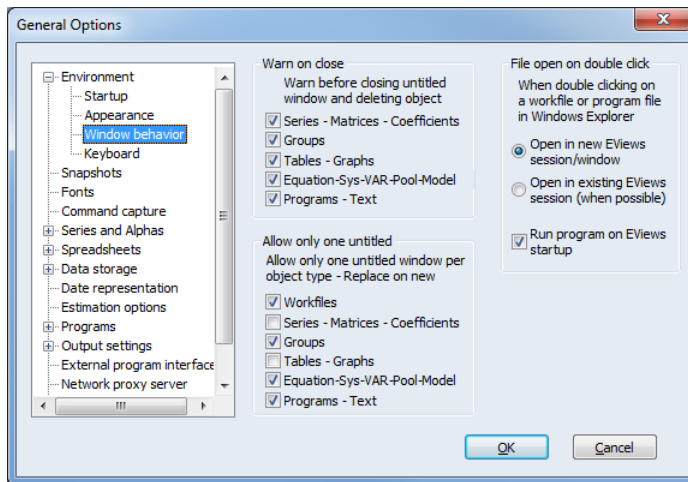
The settings are divided into broad groupings:

- The **Warn On Close** section in the upper left-hand corner of the dialog instructs EViews to provide a warning message when you close an untitled object. You may choose to set warnings for various object types. By default, EViews warns you that closing an unnamed object without naming it will cause the object to be deleted. Along with the warning, you will be given an opportunity to name the object. If you turn the warning off, closing an untitled window will automatically delete the object.

- **Allow Only One Untitled** specifies, for each object type, whether to allow multiple untitled objects to be opened at the same time. If only one is allowed, creating a new untitled object will cause any existing untitled object of that type to be deleted automatically. Setting EViews to allow only one untitled object reduces the number of windows that you will see on your desktop.

  If you elect to allow only one untitled object, we strongly recommend that you select **Warn on Close**. With this option set, you will be given the opportunity to name and save an existing untitled object, otherwise the object will simply be deleted.

- **File open on double click** controls the whether to use an existing EViews session or to launch an additional session when opening an EViews associated file. Selecting **Open in existing EViews session (when possible)** means that when using Windows to open an EViews file, say by double clicking, selecting an EViews file from the start menu, or opening as an email attachment, the file will open in an already running copy of EViews, if available.

- **Run program on EViews startup** determines the behavior of EViews when launching the program by double clicking on a program file in a Windows directory. By default EViews, will open and run the program when you double click. Unselect this checkbox to prevent the program from being run automatically. Note that if EViews is already running when the program file is double-clicked, EViews will only open the file, regardless of this setting. This setting only applies when a new instance of EViews is launched to handle the program file.

Keyboard

The **Keyboard** page offers options for setting focus and for smart editing.
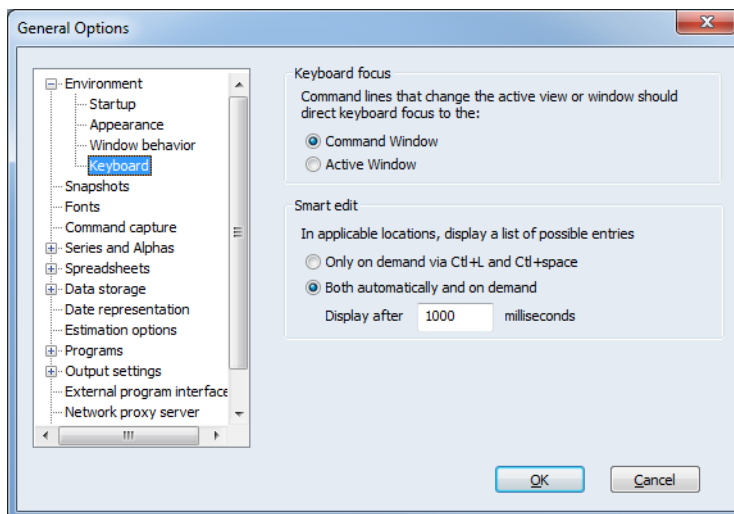
- **Keyboard Focus** controls where the keyboard cursor is placed when you change views or windows. As the label suggests, when you select **Command Window**, the keyboard focus will go to the command window following a change of view. This setting is most useful if you primarily use EViews via the command line. Choosing **Active Window** will cause the focus to go to the active window following a change of view. You will find this setting useful if you wish to use keystrokes to navigate between and to select items in various windows. Note that whatever the default setting, you may always change the keyboard focus by clicking in the command window, or by clicking in the active window.

- **Smart edit** allows you to turn off the automatic pop-up that displays possible entries in edit fields. If **Both automatically and on demand** is selected, the list will be presented, after a pause in typing, from within most edit fields except the command window. If **Only on demand via Ctl + L** is selected, you can use the keystroke, as always, to bring up the list of entries.

- You may change the delay used when automatic pop-ups are enabled, by entering the desired number of milliseconds in the appropriate edit field.

Snapshots

By default, EViews 10 will automatically check for changes every five minutes and then save a snapshot only if changes are detected. These automatic snapshots will be deleted when the workfile is closed.

You can change the default snapshot settings in the **General Options/Snapshots** dialog.
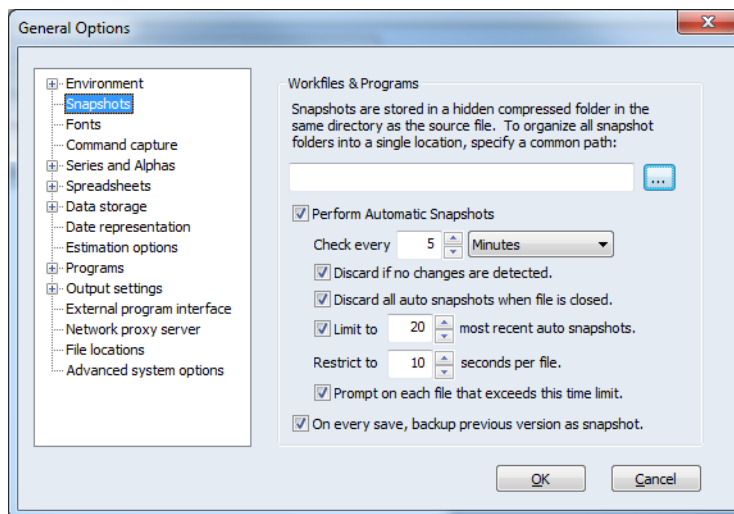
Snapshots are usually saved into a hidden folder in the same directory as the main workfile. However, you can group all the snapshot folders into the same location by specifying a common snapshots folder.

For automatic snapshots, you can change how often EViews performs the check, and whether snapshots should be kept if no changes are detected. You can also tell EViews not to discard automatic snapshots when the workfile is closed and to limit how many auto snapshots should be kept. Setting this value to zero will allow unlimited auto snapshots.

Finally, you can also set a time limit (in seconds) for an automatic snapshot to operate for each workfile. Because workfiles can sometimes take a long time to load or save, auto snap-

shots will be skipped for any workfile that takes longer than 10 seconds to load or save. This time restriction prevents auto snapshots from locking up EViews for too long while running in the background. Any workfile that takes too long to open or save will then optionally prompt you to exclude it from any further automatic snapshots. You can change this time limit to zero to remove all time restrictions.

To prevent EViews from prompting you with each workfile that takes too long, you can also uncheck the **Prompt on each file...** checkbox.

Fonts

The **Fonts** section allows you to change the default font styles and sizes for various sets of windows and objects. Select the type of object for which you want to change the font from the dropdown at the top of the dialog and modify the font settings below. For example, to set the default font face and size to be used in table objects and table views of objects, select **Spreadsheets & Table defaults** and modify the settings below the dropdown.
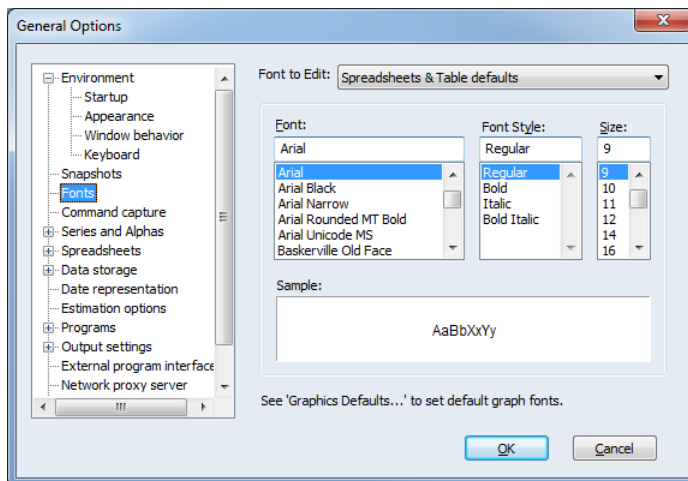
Command Capture

The **Command Capture** setting determines whether to echo the contents of the command capture to the command window. Choosing **Send command capture to Command Capture Window and to Command Window** allows the contents of the command capture to be displayed in both windows. Choosing **Send command capture to Command Capture Window Only** puts the contents of the command capture in the command capture window.

Frequency Conversion

The **Frequency conversion** dialog within the **Series and Alphas** section allows you to set the default frequency conversion methods for both up and down conversion, and the default method of displaying dates.

The default frequency conversion method tells EViews how to convert data when you move data to lower or higher frequency workfiles. The frequency conversion methods and use of default settings are discussed in detail in "Reading EViews Data using Other Applications," beginning on page 164.
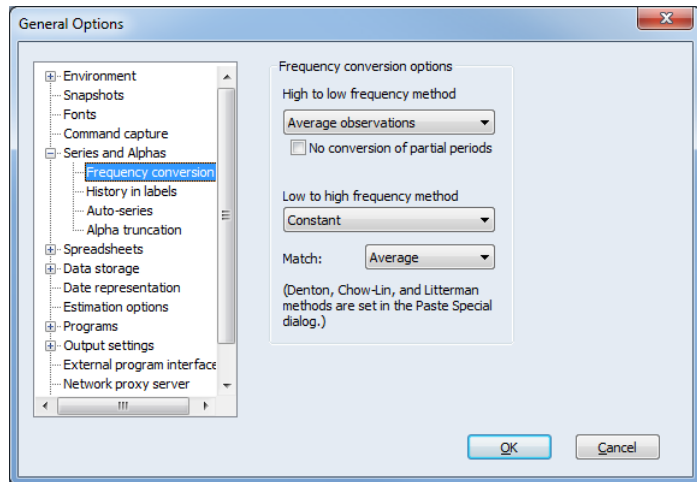
Series Auto Labels

You may elect to have EViews keep a history of the commands that created or modified a series as part of the series label. You can choose to turn this option on or off in the **History in labels** dialog within the **Series and Alphas** section.

Note that the series label may be viewed by selecting **View/Label** in a series window, or at the top of a series spreadsheet if the spreadsheet defaults are set to display labels ("Spreadsheet Layout" on page 867).

Auto-series

In the **Auto-series** page under **Series and Alphas**, the **Alpha-series in stats** option lets you control whether EViews simply ignores evaluation errors in auto-series when performing statistical analysis, or reports an error. Auto-series are functions or expressions of existing workfile series that may be used in a statistical procedure.

For example, the auto-series LOG(X) may be used as a regressor in an equation object, or as one of the series used in a cointegration test. When the option is set to **Report evaluation errors in statistical views and procs**, if the auto-series cannot be evaluated for any observation (for example, any observation where X is negative when using LOG(X)), EViews will error and, possibly, stop computation of the statistical procedure. If the option is set to **Treat evaluation errors as NAs**, any observation that cannot be evaluated will simply be converted into an NA and the statistical computation will continue, usually ignoring that observation.
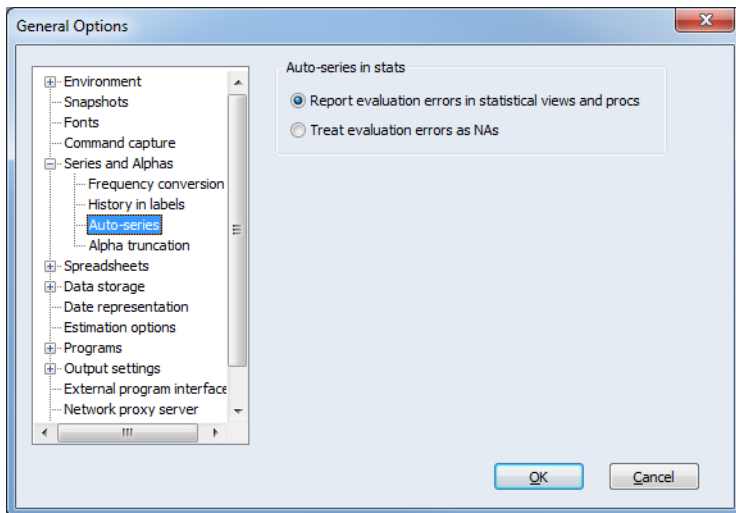
History in Labels

You may set the default for whether or not to save the command history in the series labels.

Auto-series

You may choose whether or not to treat evaluation errors in auto-series as missing (NA) values, or whether to report an error when an evaluation error is encountered.

Alpha Truncation

Note that EViews alpha series automatically resize as needed, up to the truncation length.

To modify the alpha series truncation length, select **Alpha Truncation** from the **Series and Alphas** section, and enter the desired length. Subsequent alpha series creation and assignment will use the new truncation length.

You should bear in mind that the strings in EViews alpha series are of fixed length, so that the size of each observation is equal to the length of the longest string. If you have a series with all short strings, with the exception of one very long string, the memory taken up by the series will be the number of observations times the longest string size.
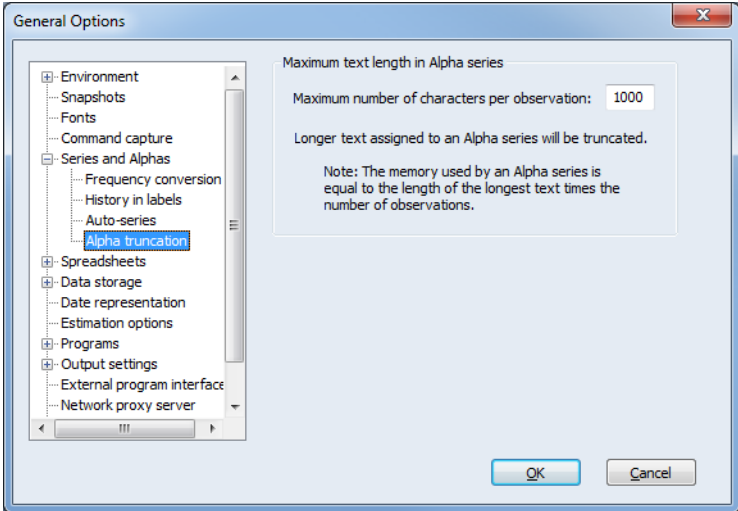
The maximum string length is 1,000 characters.

Spreadsheet Layout

The options in the **Layout** dialog within the **Spreadsheets** section control the default spreadsheet view settings of series, group, and table objects. The defaults are set using these option settings when the object is first created.

In the **Series spreadsheets** portion of the dialog, you may set the display format and choose to display the spreadsheet in one or multiple columns, with or without object labels and header information. In addition, you may choose to have edit mode turned on or off by default.
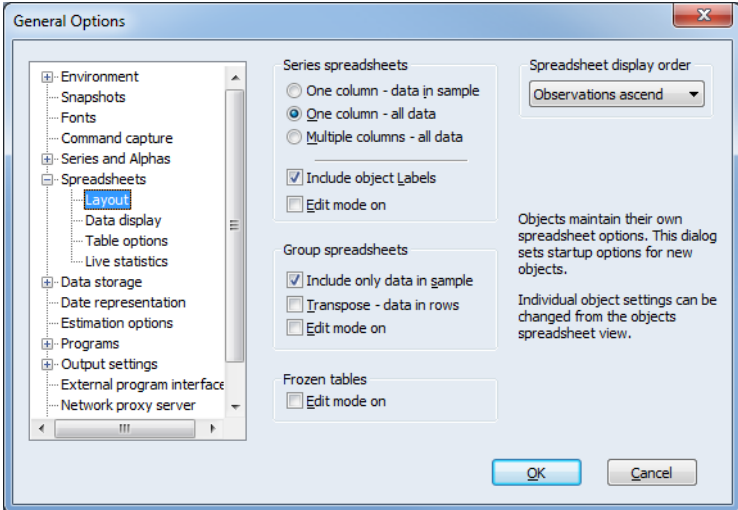
Below this section are the **Group spreadsheet** options for sample filtering, transposed display, and edit mode.

Frozen tables allows you to modify whether edit mode is on or off by default.

You may set the **Spreadsheet display order** to **ascending** or **descending** from the dropdown in the upper-right corner of the dialog.

Spreadsheet Data Display

The **Data display** dialog within the **Spreadsheets** section provides other default settings that apply to new numeric and text spreadsheets.

The **Data displayed as** dropdown menu, in the **Numeric display** portion of the dialog, determines what type of data the rest of the section refers to. You can specify a display format for each of the transform types specified in the dropdown menu (**Level & Difference**, **% Change & Log Dif**, or **Logarithm**). For instance, if you select **Level & Difference**, you can alter the numeric formatting of cells, set the number of digits to be displayed, as well as specify the characters used as separators and indicators for negative numbers, for data displayed using this transform method. To set the default display for other transform types, you may change the **Data displayed as** dropdown menu.

You may also choose the default **Text justification** and **Indentation** for alphanumeric display.

We emphasize the fact that these display options only apply to newly created series or group objects. If you subsequently alter the defaults, existing objects will continue to use their own settings.

Table Options

The table options section contains options for copying tables.
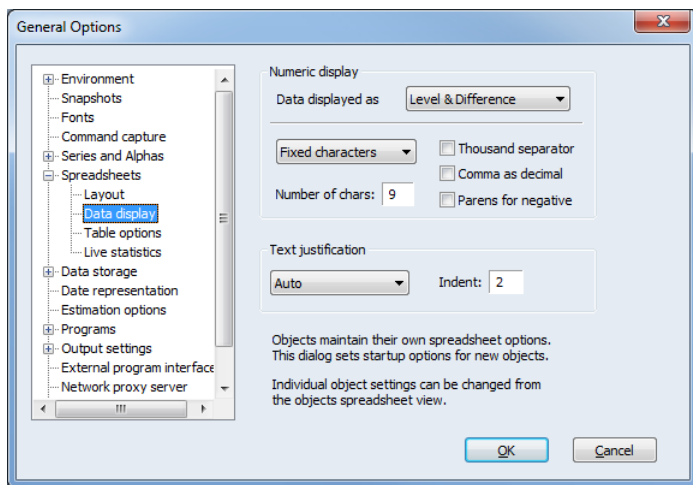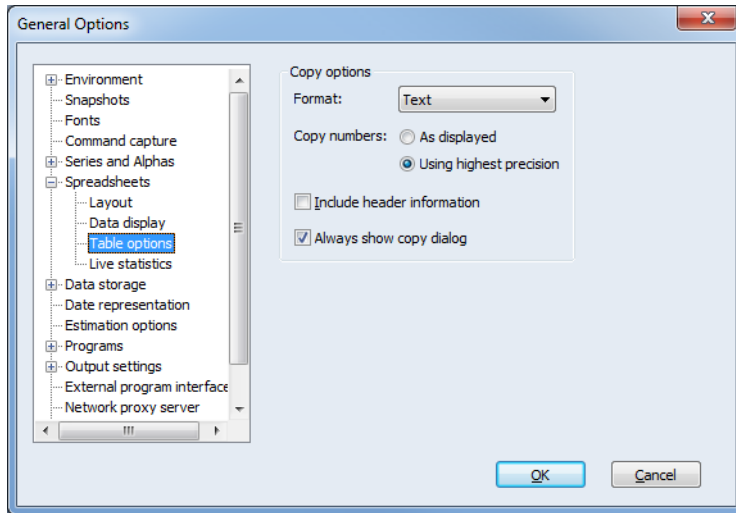
You may, by default, copy in text, LaTex or HTML formats. Number may be copied as depicted, or at highest resolution. Table headers may be copied or not, and you may show or suppress the copy options dialog each time you perform a copy.

Workfile Storage Defaults

The **Workfile Save** dialog in the **Data storage** section allows you to specify storage precision and compression options for your saved workfiles, and to specify whether to create backup workfiles when overwriting existing files on disk.

The **Series storage** portion of the dialog controls whether series data are stored in saved workfiles in single or double precision. Note that workfiles saved in single precision will be converted to double precision when they are loaded into memory.

In addition, you may elect to save
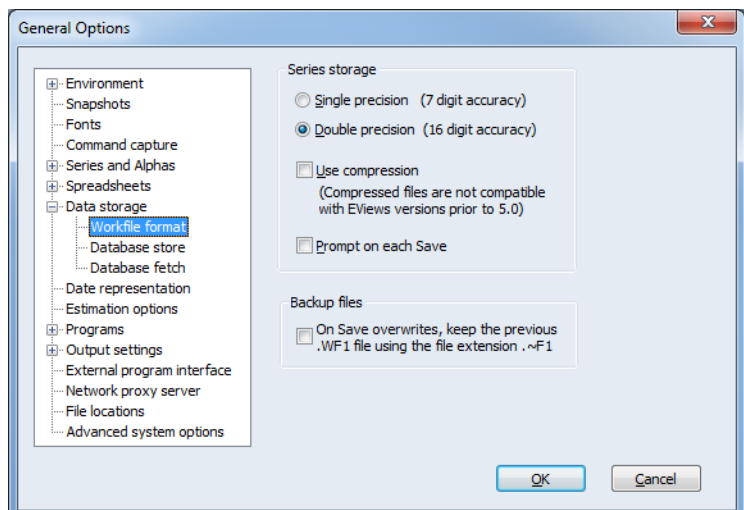
your workfiles in compressed format on disk by checking the **Use compression** setting. Note that compressed files are not readable by versions of EViews prior to 5.0, and that they do not save memory when using the workfile, as it is uncompressed when loaded into memory. Compressed workfiles do, however, save disk space.

You should uncheck the **Prompt on each Save** checkbox to suppress the workfile save dialog on each workfile save.

In addition, you may specify whether EViews should keep backup copies of workfiles when saving over an existing file. If selected, the automatic backup copy will have the same name as the file, but with the first character in the extension changed to " ~ ".

Database Storage Defaults

You may also control the default behavior of EViews when moving data into and out of databases using the **Database Store** and **Database Fetch** dialogs within the **Data storage** section. These dialogs control the behavior of group store and fetch (see "Storing a Group Object"

and "Fetching a Group Object" on page 329), and whether data are stored to databases in single or double precision.
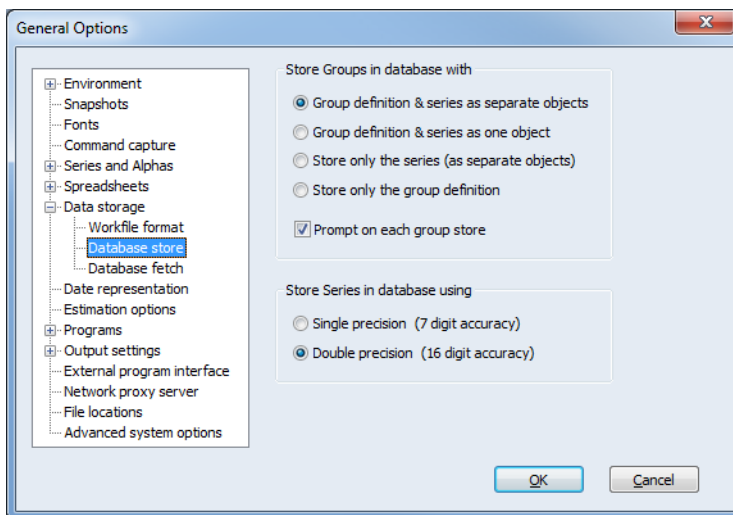
Date Representation

The **Date representation** dialog controls the format for dates in sample processing, and in workfile and various object views. For daily and weekly data, you can set the default to American (**Month/Day/Year**) or you may switch to European notation (**Day/Month/Year**), where the day precedes the month. You may also specify your quarterly or monthly date display to use the **Colon delimiter** or a **Frequency delimiter** character based on the workfile frequency ("q" or "m"). The latter has the advantage of displaying dates using an informative delimiter ("1990q1" vs. "1990:1").
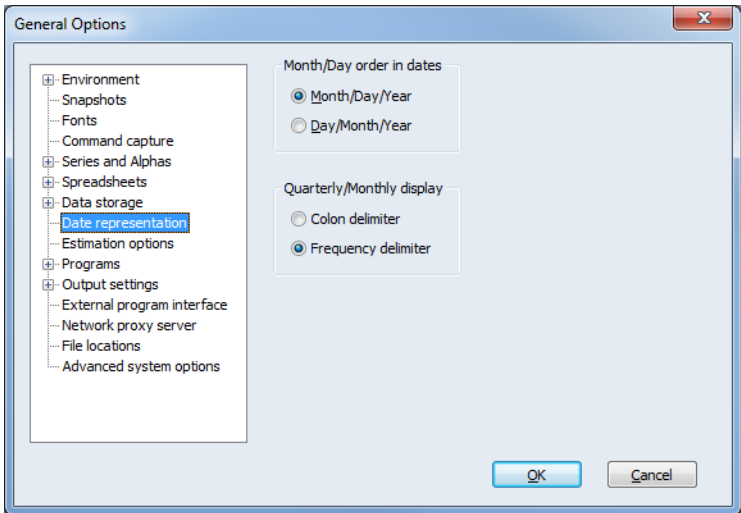
See also "Free-format Conversion Details" on page 114 in the *Command and Programming Reference* for related discussion.

Estimation Defaults

You can set the global defaults for the maximum number of iterations, convergence criterion, and methods for computing derivatives in the **Estimation options** dialog.

These settings will be used as the default settings in iterative estimation methods for
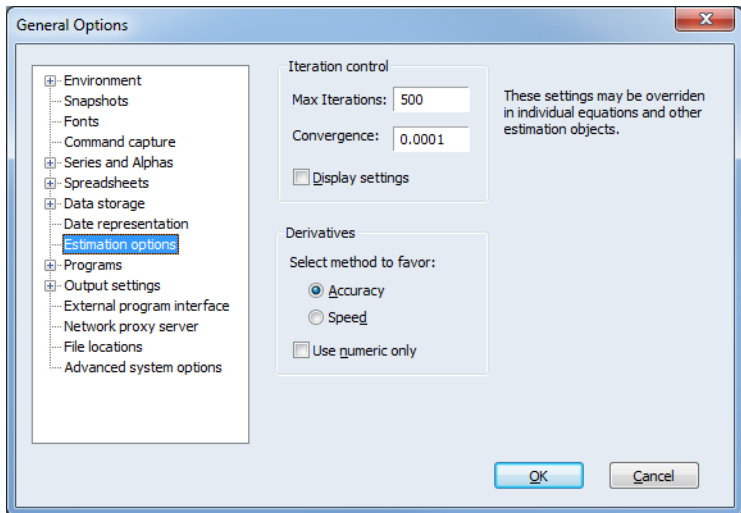
equation, log likelihood, state space, and system objects. Note that previously estimated EViews objects that were estimated with the previous default settings will be unchanged, unless reestimated. See "Setting Estimation Options" on page 1089 for additional details.

When the **Display settings** option is checked, EViews will produce additional output in the estimation output describing the settings under which estimation was performed.

Programs

The first page of the **Programs** section provides some general options for running EViews programs. You can stop a program from running after exceeding a specified number of errors by entering the error count in **Maximum allowable errors before halting**.
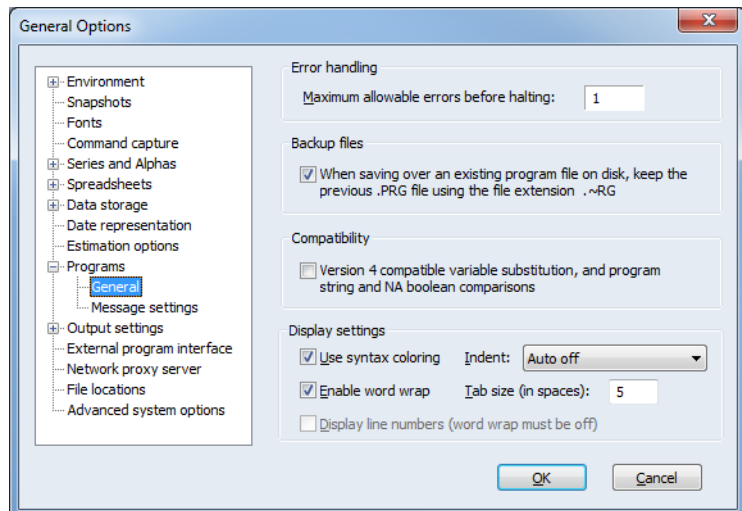
You may also use this dialog to specify whether EViews should keep backup copies of program files when saving over an existing file. The backup copy will have the same name as the file, but with the first character in the extension changed to " ~ ".

If the checkbox labeled **Version 4 compatible variable substitution** is selected, EViews will use the variable substitution behavior found in EViews 4 and earlier versions. EViews 5 changed the way that substitution variables are evaluated in expressions. You may use this checkbox to use Version 4 substitution rules. See "Version 4 Compatibility Notes" on page 172 for additional discussion.

In addition, you may modify the **Display Settings** to change the display of text and editing behavior in program files.

If **Use syntax coloring** is selected, basic programming keywords, strings, and comments will be displayed in color. You may also **Enable word wrap** and **Display line numbers** in programs.

The two controls on the right-hand side determine auto-indenting and tab size settings. The **Tab size (in spaces)** edit box lets you set how many characters a tab will use in the program editor. The **Indent** dropdown lets you specify the type of automatic indenting that the pro-

gram editor will use. **Auto on** means that EViews will attempt to automatically apply a single tab for lines of code that are inside a program statement ("For/Next", "If/Endif", "While/Wend"), or inside a subroutine. Note that the single tab will be applied on top of any tabbing that was already in place. **Auto on w/completion** means that EViews will apply automatic indenting, and will also automatically close any program statement. For example, EViews will automatically add "Next" to the start of the next line when you finish a line starting with "For". **Auto off** means no automatic indenting will be employed.

The **Runtime settings** dialog in the **Programs** section specifies whether, by default, EViews runs programs in **Verbose** mode, listing commands in the status line as they are executed, or whether it uses **Quiet** mode, which suppresses this information. EViews will run faster in quiet mode since the status line display does not need to be updated.
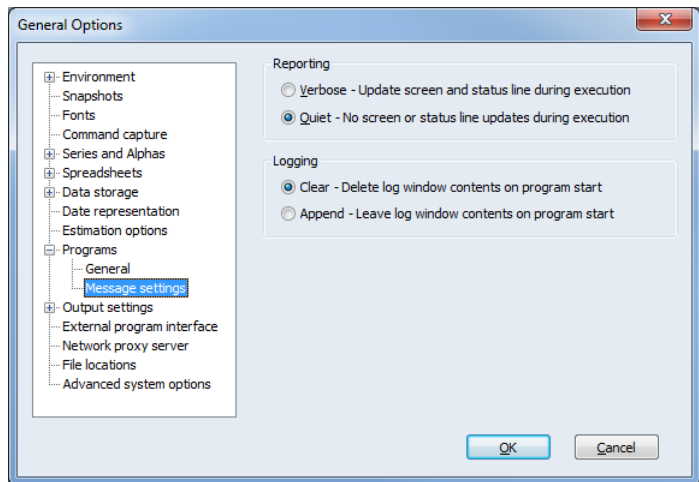
This default may always be overridden from the **Run Program** dialog, or by using the option "q" in the run statement, as in:

```
run(q) myprogram
```

For details, see "Executing a Program," beginning on page 121 of the *Command and Programming Reference*.

PDF Page Settings

The **PDF page settings** dialog determines default settings when you export an object as a PDF (Portable Document Format). You may set the **Paper** to a preset type and size, or determine your own. You may also set the default **Orientation** and **Margins** for your PDF output.
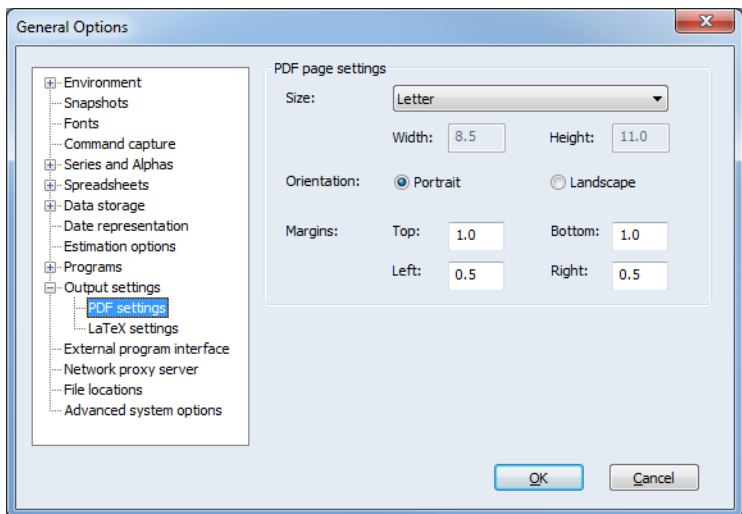
LaTex Settings

The LaTex settings section controls the default output of tables in LaTex as either tables, or images embedded in LaTeX commands. In addition, you can specify a directory in which the images should be stored, and instruct EViews to put the LaTeX input command on the clipboard so it may easily be used in your documents. You may also set the default for whether or not to include a full LaTeX document specification in your output.
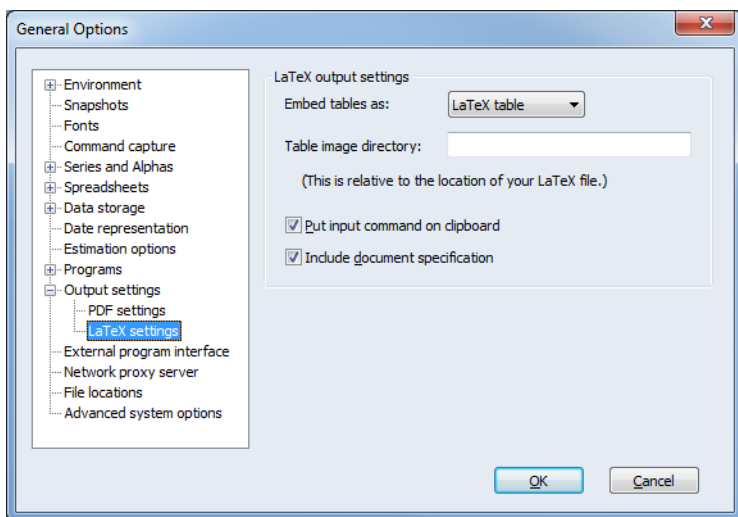
External Program Interface

The **External pro-
gram interface**
entry allows you to
specify the default
case to use when
creating destina-
tion objects in
MATLAB or R
using the `xput`
command. The
defaults may be
overridden by
specifying the
"case = lower" or
"case = upper"
option in the

`xopen` command or by specifying an explicit destination name using the "name = " option in
the `xput` command.

You may also use the **COM ProgID** section of the dialog to specify the default Programmatic
IDs (ProgIDs) used when instantiating a MATLAB or R COM server. The ProgID is used to
indicate the version of MATLAB or statconnDCOM to which EViews connects when you
issue an `xopen` command. Each dropdown will contain values found in the Windows regis-
try on your system that were entered when you installed MATLAB and statconnDCOM. You
may choose from entries in the dropdowns, or type in a specific ProgID for EViews to use
when opening a MATLAB or R session. The default settings may be overridden by specifying
"progid = " in the `xopen` command.

For details on these settings see `xopen` (p. 555) and `xput` (p. 557) in the *Command and Pro-
gramming Reference*.
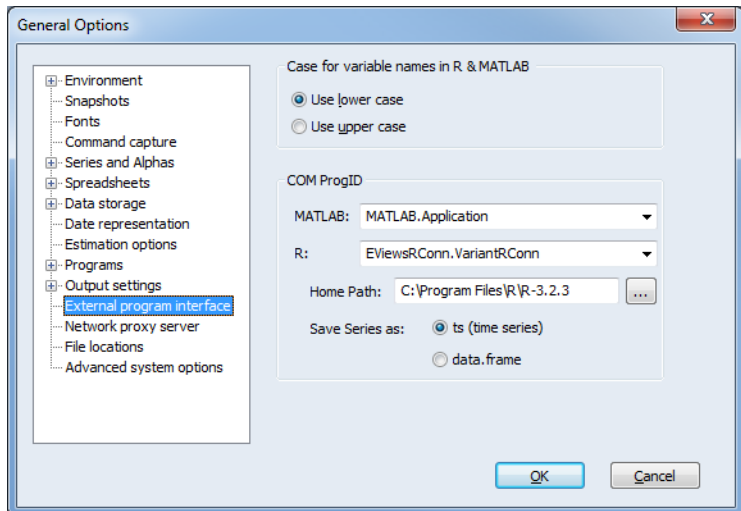
Network Proxy Server

Some EViews features (such as automatic update, registration) require internet access. If internet access is only available on your network via a proxy server, you may provide EViews will your proxy server settings.

Simply click on **Network proxy server** to display the relevant dialog and enter your settings. You may click on the **Auto detect** button to have EViews try to determine your settings for you. Note that valid proxy server setting values generally take the form:

```
servername:8080
```

where `servername` represents the name (or IP address) of the proxy server and `8080` is replaced by the actual port number be employed.

If Microsoft Internet Explorer has previously setup properly with a proxy server on the current machine, you can click the **Auto detect from IE** button to import the current settings into EViews.

EViews is only concerned with HTTP and HTTPS access. For a single proxy server that supports both protocols, enter the server name and port number into the HTTP edit field.

If a different server is used for HTTPS, uncheck the **Use the same proxy server** checkbox and enter the server name and port number into the HTTPS edit field.

For a proxy server that requires username/password authentication, enter the information into the **Proxy Authentication** fields.

Click the **Test Internet Access** button to test these settings on your local network.
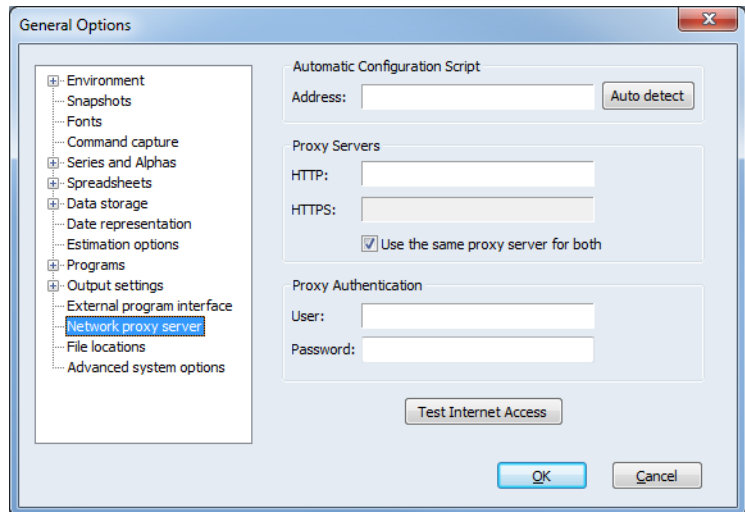
File Locations

This selection allows you to set the default working directory, and the locations of the .INI file, database registry and alias map, and the temp directory.

The dialog also reports, but does not allow you to alter, the location of your EViews executable.
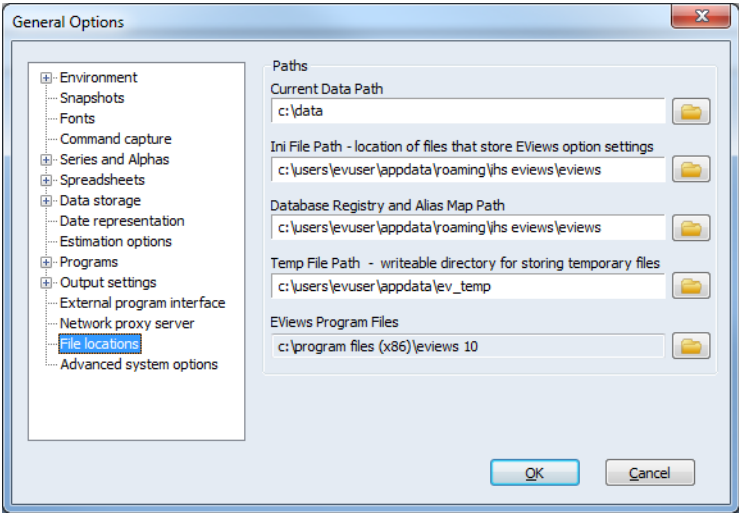
Note that the default working directory may also be changed via the **File Open** and **File Save** or **File Save As** dialogs, or by using the `cd` command.

Advanced System Options

You may use the **Advanced system options** dialog to set memory allocation and expression evaluation options. *Unless there is a specific reason for changing settings, we strongly recommend that you use the default EViews settings.*

Memory Allocation

The top portion of the dialog allows you to adjust the EViews memory allocation settings.
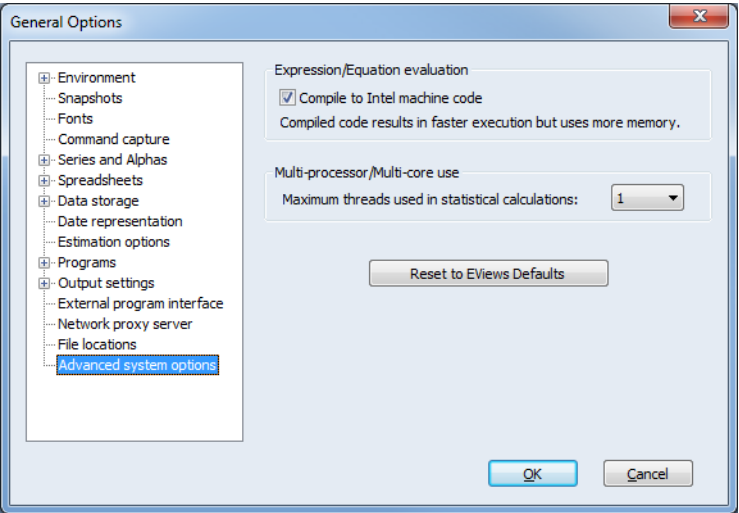
By default, for 32-bit machines, EViews allows a maximum of 4 million observations per series. You may use the dropdown menu to increase or decrease this limit. If you using the

64-bit version of EViews, this control will not appear, and you will be limited to 120 million observations per series.

The total memory available for all EViews objects is limited by the maximum address space per application. Standard Windows XP systems allow for 2GB of address space, but may be configured to use 3GB (using the /3GB switch). Windows XP x64 systems allow for 4GB of address space per application. The amount of available address space in 64-bit versions of Windows will depend on the specific version. Note that if the amount of physical memory available is less than the address space in use, available disk space will be used as virtual memory, which will significantly degrade performance.

EViews reserves a portion of this address space for auxiliary purposes. This memory is used by the operating system, DLLs, external database drivers, and by EViews when creating objects. Reducing the size of the reserved space will increase the amount of address space available for holding observations, but will decrease the total number of objects allowed in EViews, and may lead to instability caused by out-of-memory conditions.

By default, EViews increases the speed of evaluation by first compiling expressions to machine code. The pre-compiled code executes more rapidly, but the compilation procedure and evaluation requires additional memory. You may uncheck this setting to use the slower, memory efficient method of evaluation.

Multi-Processor Use

The **Multi-processor/Multi-core use** section may be used to define the maximum number of threads you would like to allow EViews to use when processing statistical calculations. EViews treats processors and cores symmetrically, and does not include hyper-threaded processors. In general, we recommend leaving this setting at **Auto**.

When set to **Auto**, EViews detects the number of cores in your system and attempts to use all of them. You may override this setting for up to eight threads. This allows you to increase the number of threads EViews uses to beyond the physical number of cores. We have not found this to be effective, but, in principle, it could improve performance with hyper-threading. If you require more processing power for other applications, you may wish to limit the number of threads EViews uses by setting this number below the physical number of cores. This is guaranteed to work with Intel processors, and, to our knowledge, should work with other brands of processors as well.

You may click on the **Reset to EViews Defaults** button to return to the default settings.

Graphics Defaults

These options control how a graph appears when it is first created. The great majority of these options are explained in considerable detail in "Graph Options," beginning on page 768.
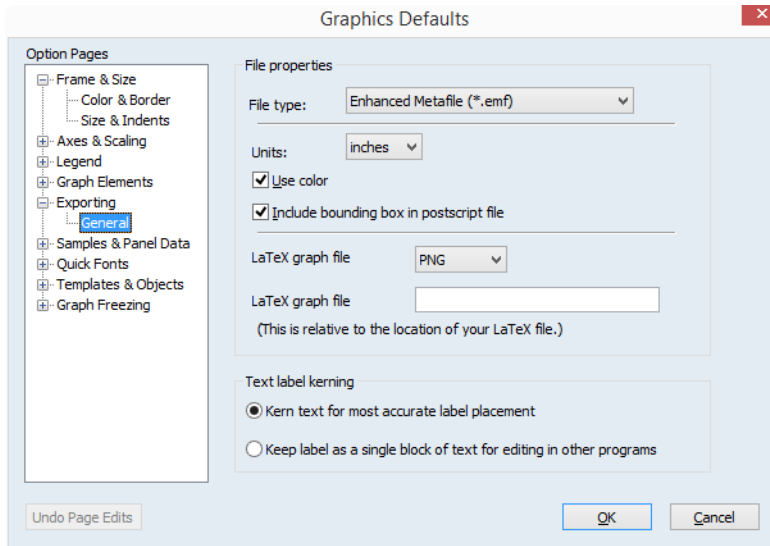
Additional dialog pages are provided for specifying the default settings for use when saving graphs to file (**Exporting**) and for handling sample breaks, missing values, and panel data structures (**Samples & Panel Data**).

Default settings for color and background transparency may be set in the **Frame & Size/ Color & Border** entry of the **Graph Options** dialog.

The **Quick fonts** section lets you set multiple graph font settings all at once. You can change the font face, style and size for all axes, legends, text objects and observation labels in a single step.

See "Basic Customization," beginning on page 648 and Chapter 15. "Graph Objects," on page 751.

Database Registry

The **Database Registry** settings are described in detail in "The Database Registry" on page 331.

Cloud Management

The **Cloud Management** settings are described in "Cloud Drive Support" on page 96.

EViews Auto-Update from Web

IHS Global Inc. occasionally releases patches to the EViews program, and we strongly recommend that you keep your copy of EViews as up-to-date as possible.

EViews includes an automatic updating feature that can check for new updates automatically, and install any update that may be available. The automatic update feature can be disabled from the **Options/EViews Auto-Update from Web** item in your main EViews menu. You can also manually check for updates from within EViews at any time by selecting **Check now...** under the **EViews Auto-Update from Web** menu item, or by selecting **EViews Update** from the **Help** menu.

You may also check for updates to the EViews program or for updates to supplementary components such as the EViews documentation or sample data and programs, by visiting the Download section of the EViews website (www.eviews.com).

Print Setup

The **Print Setup** options (**File/Print Setup...** on the main menu) determine the default print behavior when you print an object view.
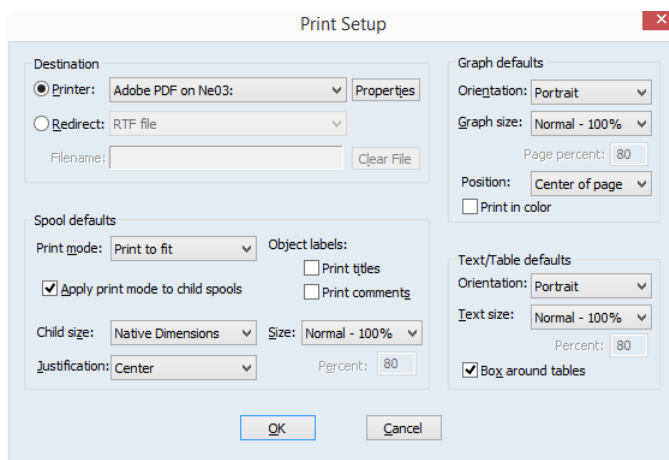
The top of the **Print Setup** dialog provides you with choices for the destination of printed output.

You may elect to send your output directly to the printer by selecting the **Printer** radio button. The drop down menu allows you to choose between the printer instances available on

your computer, and the **Properties** button allows you to customize settings for the selected printer.

Alternatively, you may select **Redirect** so that EViews will redirect your output. There are three possible settings in the drop down menu: **RTF file**, **Text file (graphs print)**, **Frozen objects**, and **Spool object**:

- If you select **RTF file**, all of your sub-sequent print commands will be appended to the RTF file specified in **Filename** (the file will be created in the default path with an ".RTF" extension). You may use the **Clear File** button to empty the contents of the existing file.

- If you select **Text file (graphs print)**, all printed text output will be appended to the text file specified in **Filename** (the file will be created in the default path with a ".TXT" extension). Since graph output cannot be saved in text format, all printing involving graphs will still be sent to the printer. Spools will send all text and table output to the file, and will ignore all graph output.

- **Frozen objects** redirects print commands into newly created graph or table objects using the specified **Base Object** name. Each subsequent print command will create a new table or graph object in the current workfile, naming it using the base name and an identifying number. For example, if you supply the base name of "OUT", the first print command will generate a table or graph named OUT01, the second print command will generate OUT02, and so on.

- Selecting **Spool object** sends subsequent print commands into a spool object in the workfile. Spool objects allow you to create collections of frozen object output (see Chapter 17. "Spool Objects," on page 801 for details).

There are also settings for text printing, CSV printing and LaTeX file printing.

The other portions of the **Print Setup** dialog set various default settings for spool, graph, and table printing.

The **Graph defaults** section has settings for printing in portrait or landscape mode, scaling the size of the graph, positioning the graph on the page, and choosing black and white or color printing. For example, the **Orientation** dropdown allows you print graphs in **Portrait** or **Landscape** orientation. Similarly, you may use the **Graph size** dropdown to scale all graphs by a fixed percentage when printing. Note that some graph settings are not relevant when redirecting output, so that portions of the dialog may be grayed out.
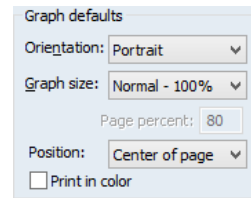
See "Printing Graphs" on page 786 for details.

The **Text/Table defaults** allow you adjust settings for printing tables and text objects. You may use the **Orientation** dropdown allows you print graphs in **Portrait** or **Landscape** orientation, Similarly, you may use the **Graph size** dropdown to scale all text by a fixed percentage when printing. Some of these options are not available when redirecting output.
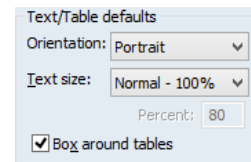
See "Printing Tables" on page 798 for details.

The **Spool defaults** portion of the dialog includes settings specific to printing spool. You may use the **Print mode** dropdown and **Apply print mode to child spools** checkbox to specify where page breaks are to be used, the **Object labels** section to print or suppress titles and comments, or the Child size, Justification, and Size dropdowns to size and position the output.
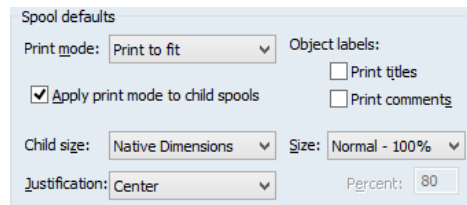
See "Printing a Spool" on page 819 for details.

We remind the user that this dialog only specifies the default settings for printing. You may override any of these settings when printing a particular job using the **Print** dialog or the `print` command.

Global Options Export and Import

Note that you may save your options settings to disk for backup and update your settings from an existing settings file. One use for these commands is to establish a set of global options settings which are then saved and then distributed to other users.

The `optsave` command (optsave (p. 433) of the *Command and Programming Reference*) saves a copy of the current EViews global options settings ".INI" files into a directory. Any existing option settings in the directory will be overwritten.

You may use the `optset` command (`optset` (p. 434) of the *Command and Programming Reference*) to replace the current EViews global options settings ".INI" files with ones from a different directory.
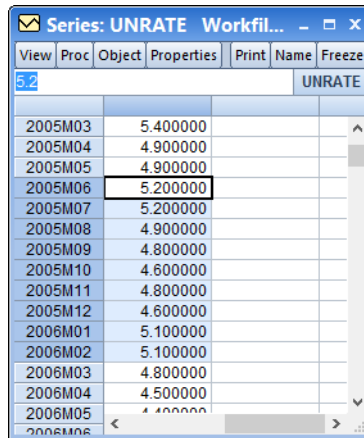
Appendix B. Enhanced Spreadsheet Editing

Array expressions offer you a powerful new language for editing cells in a spreadsheet. You may use these expressions to assign single values or ranges, to modify the existing cells of a spreadsheet using the original or adjacent cells, or to interpolate values.

Array Expressions

When editing a spreadsheet view of a series or group you may edit multiple cells of the spreadsheet at a time by using *array expressions*. To use an array expression, simply select the range of cells you wish to edit, then type the array expression you wish to use in the edit field at the top of the window, and hit return.

The simplest *array expression* is the " = " operator. This assigns the same value to each cell currently selected. In the example below, we select nine cells of the series UNRATE covering the period from June 2005 to February 2006.

Type " = 3.5" and hit return. All of the selected cells' values change to be equal to 3.5:

Array expressions take one of two forms: operator/value adjustment and interpolation. Operator/value adjustment lets you specify an operation to apply to the existing data and specify values to use for that operation. Interpolation simply interpolates any missing values in the selected cells.

The full syntax for an array expression is either made up of the operator and values component:

        [operator][values]

or a single interpolation component:

        [interpolation]

Operators

The operator component defines how the new values will be assigned to the cells. The following table outlines the available operators.

Operator	Description
=	Overwrites the existing value with the new value.
+=	Adds the new value to the existing value.
-=	Subtracts the new value from the existing value.
*=	Multiplies the existing value by the new value.
/=	Divides the existing value by the new value.
=_	Overwrites the existing value with the previous cell's value.

+_	Add the new value to the previous cell's value.
-_	Subtract the new value from the previous cell's value.
*_	Multiply the previous cell's value by the new value.
/_	Divide the previous cell's value by the new value.
\	Reverse the order of the selected cells. Note this operator does not require a values component.

We have already seen an example of the simplest operator, the " = " which simply changes each cell to contain a new value. The following examples, using the same data as above, shows the impact of typing " + = 3" as the array expression, and " + _3" as the expression:

The first pane shows the original data. The middle pane shows the impact of " + = 3", where each value in the selected region is equal to the original data plus 3.

The final pane shows the impact of entering " + _3" on the original data. Each value in the selected region is now equal to the previous cell's value plus 3. The calculation is done recursively; only the first cell's calculation uses the existing data. In the example shown in the final pane, the value of the first selected cell (7.9 in 2005M06) is equal to the previous value (4.9 in 2005M05) plus 3. The value in the second cell is equal to the new value in the first cell, 7.9, plus 3.

You should note that the two examples above were not done recursively. We performed the second example on the original data, not the data that had been already changed by the first example. EViews does not remember the "original" state of the data in the spreadsheet; once you use an array expression to change the data, that change is permanent and all subsequent modifications will be made using the new data.

Values

The simplest form of a value is a single number, such as the "3.5" or the "3" used in the examples above. When a single number is used, each cell in the selected range uses the same number. In the " + =3" example above, each cell had 3 added to its existing value.

If you do not wish to use a single value for all cells, you may also specify a start point and an end point, separated by a ".." and let EViews fill in the values in between. For example, if you want the first cell in your selection to be the value "3", and the last cell to be the value "6", you would enter "3..6" as your values:

In this example we used the array expression " =3..6". Since we used the "=" operator, we are overwriting the existing values with the new values. The first cell in the selection received a value of 3, and the last cell received 6. The cells in between were interpolated using simple linear interpolation.

When specifying a start and end point, you do not need to use the "=" operator; any of the operators can be used. In this example, we use the " + _3..6" expression:

The first pane shows the original data, and the second pane shows the effect of array expression. The value for the first cell is equal to the previous cell's value, 4.9, plus 3. The value in the second cell is equal to that of the first cell, 7.9, plus the interpolated value of 3.375. The value of the third cell is equal to the value in the second cell, 11.275, plus the interpolated value of 3.75. This continues until the value in the last cell is equal to the value in the penultimate cell plus 6.

If you wish to interpolate between two existing data values, you do not need to type them in as the start point or end point. If you use a ".." without a value for the start point, EViews will simply use the value in the spreadsheet cell *prior to the selection*. If you do not provide an end point, EViews will use the value in the cell *after the selection*.

For example, using an array expression of " = ..6" will interpolate between the value in 2005M05 (4.9) and 6, and enter the first interpolated value in the first selected cell, 2005M06:

Since we entered an end point, the final cell in the selection, 2006M02, receives that end point value. If we use an array expression of " = .." without specifying a start or end point, EViews will interpolate between the value in 2005M05 and 2006M03 (the points before and after the selection), and the entire selection will be replaced with the interpolated values:

By default, any ".." values will be interpolated using simple linear interpolation. You may change this to perform multiplicative (log-linear) interpolation by appending an "*" to the end of your expression. This allows you to perform a geometric trend, rather than a linear trend to fill in the values. The following example shows the use of the expression " = 3..11" and " = 3..11*":
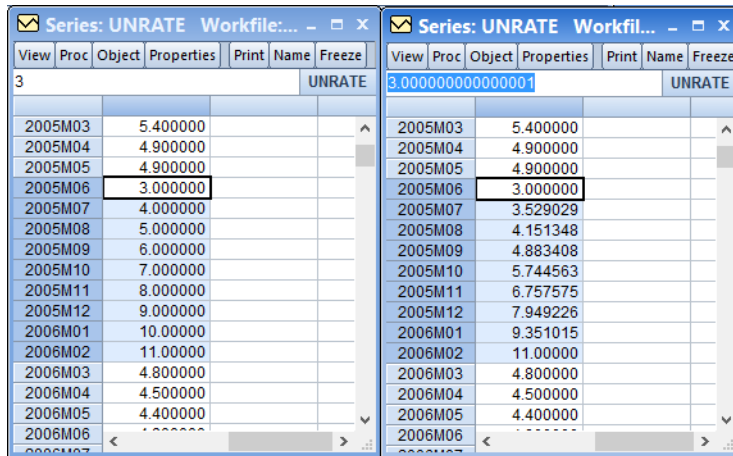
Since we have 9 cells selected, and we specified the values 3 and 11, linear interpolation (the left pane) is straightforward to calculate: the increase in each cell is constant, and equal to (11-3)/(9-1) = 1. The right hand pane shows multiplicative interpolation. Here, the increase in the log of each cell value is constant, and is equal to (log(11)-log(3))/(9-1).

Interpolation

An interpolation array expression will interpolate any missing (NA) values in the selected range, and should not be used with either an operator or values component. There are a number of different interpolation techniques that can be used, each with its own symbol:

Method symbol	Description
_	Repeats previous non-missing value.
^	Linear interpolation.
~	Cubic spline interpolation.
&	Catmull-Rom spline interpolation.
^*	Log-linear (multiplicative) interpolation (linear in the log of the data).
~*	Multiplicative cubic spline interpolation (a cubic spline on the log of the data).
&*	Multiplicative Catmull-Rom spline interpolation (a Catmull-Rom spline on the log of the data).

Take the following selection as an example:

This selection contains a number of missing values. We first use the simple array expression "_" (we simply type "_" and then hit return) to interpolate the missing values with the previous available value:

The missing values in 2005M07 and 2005M08 are replaced by the previous available value, that for 2005M06, 5.20. Similarly, 2005M12 receives 2005M11's value, and 2006M4, 2006M5 and 2006M6 all receive the value from 2006M3.

If we use the linear interpolation expression instead ("^"), we obtain different results:

2005M03	5.400000
2005M04	4.900000
2005M05	4.900000
2005M06	5.200000
2005M07	5.066667
2005M08	4.933333
2005M09	4.800000
2005M10	4.600000
2005M11	4.800000
2005M12	4.950000
2006M01	5.100000
2006M02	5.100000
2006M03	4.800000
2006M04	4.850000
2006M05	4.900000
2006M06	4.950000
2006M07	5.000000
2006M08	4.600000
2006M09	4.400000
2006M10	4.100000
2006M11	

Here, the value for 2005M7 is one third of the distance between the value for 2005M6 and 2005M9 (i.e. $(1/3)*(4.8\text{-}5.2) + 5.2$), and the value for 2005M8 is two thirds of the distance between those two values. Similarly, the value for 2005M12 is half way between the value for 2005M11 and 2006M1.

Using one of the spline interpolation methods allows for more curvature in the interpolated values. Here we use a multiplicative Catmull-Rom spline ("&*"):

Index

D

X

Y

Z