Chapter 4

Building A Data Bank

The subject of this chapter is the creation and maintenance of a Data Bank. Following from the discussion of the Memory File in Chapter 4, the obvious first question is: Why a Data Bank?

There are several advantages to storing your data permanently in a Data Bank, rather than a Memory File. One of these is faster access to the data, and generally a higher level of documentation, plus its capacity to hold a much larger number of series, a maximum of 10,000, rather than 500. However, in addition, a Data Bank has the characteristic that it can be interviewed even if it is not open: you can print the index of any bank that is located on any accessible disk; as a result, you can often find out without accessing it whether or not a given bank contains a particular data series. In contrast, a Memory File must first be loaded before its index can be displayed.

A further benefit is that you will be able to update or extend the series easily, while keeping them distinct from any temporary or working data in the current Memory File. It is a good idea to put series into a bank if you want to use them repeatedly or to make them available to other people. It is also possible to access simultaneously as many as 15 Data Banks, whereas only one Memory File can be loaded at a time, although it is possible to merge two or more of these. In addition, banks are designed to be used both in a single user environment and on a network, whereas a Memory File is meant to be used exclusively by one person.

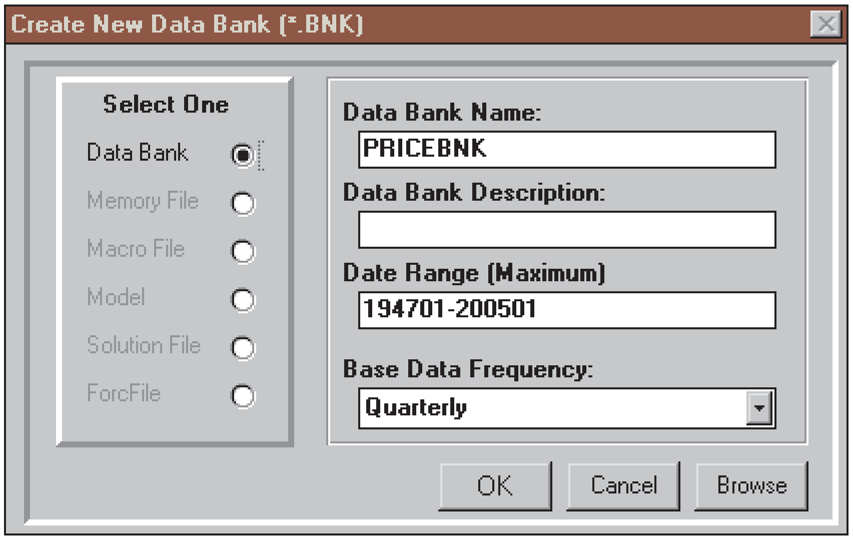
Of course, the defect of the Memory File—fundamentally its role as a temporary work file—is also its advantage: Simplicity. The creation and maintenance of Data Banks requires more thought and organizational care, in large part because of their archival characteristics. The presumption is that bank series are valuable and likely to be used either by more than one person or, if by only one person, over a sufficiently long period of time that it is worth going to the trouble of documenting the work.

Creating A Data Bank

Suppose you are going to build a Data Bank of price indexes derived from series in DEMOBANK. Start by typing:

```
SET FREQ=Q
CREATE PRICEBNK
```

to tell DataView to initialize the Data Bank; this bank will appear on the disk directory as PRICEBNK.BNK. Alternatively, you can click on **File** and then **New**. In either case, you should see next a display similar to that in Figure 4-1.


Create New Data Bank (*.BNK)

Select One: [✓ Data Bank] Memory File Macro File Model Solution File ForcFile

Data Bank Name: PRICEBNK

Data Bank Description:

Date Range (Maximum): 194701-200501

Base Data Frequency: Quarterly

Figure 4-1. Data Bank Creation Form

This display shows filled in the entries that must be provided. The description, although strongly recommended as it is prominently displayed in various contexts, is not strictly necessary and can be provided later.

Once you fill in the required entries and click OK, DataView immediately creates the bank. From this point on, although empty of series, the Data Bank exists, and for the time being (until you Close it or leave DataView) it is open for updating. “Open for Updating” means that new documentation and data will go into the bank, not the Memory File. While a bank is open for updating, it can be accessed on more or less the same terms as a Memory File ordinarily: during this time it is designated as the primary storage file.

Important: Before you can create a bank, you must set the observation frequency. The setting does not limit your ability to put series of different frequencies in the same bank (provided they have different names), but it will establish the maximum number of observations on any series in the bank. For example, the period from 1950 to 2049 defines a hundred observations of annual data, 400 observations if the frequency is quarterly, or 1200 observations if monthly. Thus if the base frequency is annual, for example, you will only be able to include just over 8 years of data on any *monthly* series you might choose to include (100/12=8.33).

Data Banks are designed to store a maximum of 1600 observations on each series, but a bank that hits this limit will tend to be large, particularly if it contains more than a few series. Today's gigabyte hard disks make size less of an issue than it once was, but you may wish to attach a bank to an email, or copy it in some other way, so that size is not yet irrelvant. Ideally, you should use only the space you actually need; a new bank can be created easily whenever you need to store more observations.

Data Entry Options

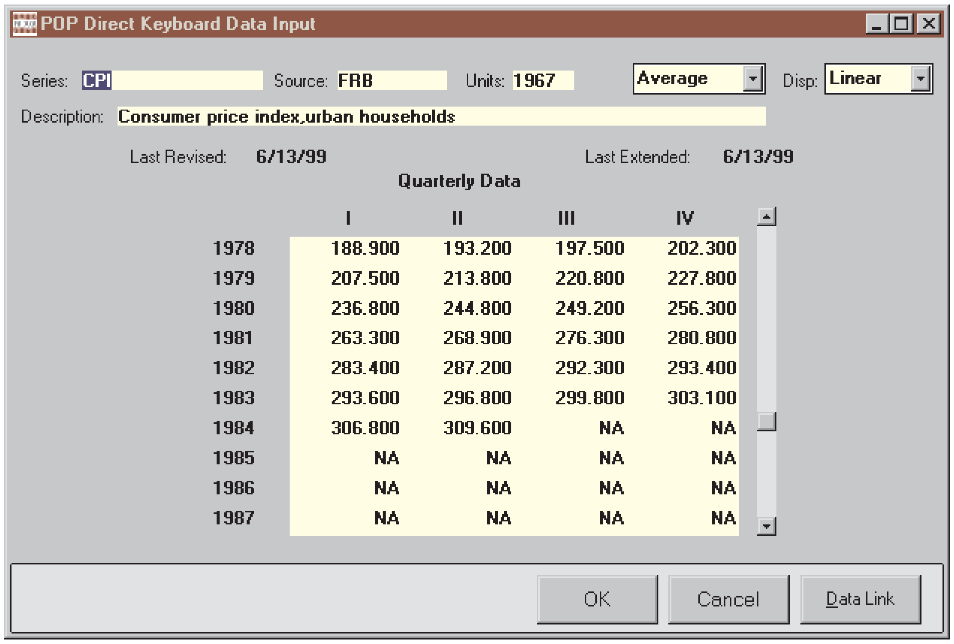
The next few subsections describe methods of data entry for series. However, these descriptions should be regarded as introductory, rather than exhaustive. There are other options available. These are described in the relevant sections of the online help facility, which also contain more detailed information about the methods described below.

The POP Facility

The easiest way to load single series into a bank from the keyboard is to use the POP on-screen data entry facility, introduced in Chapter 3. To create a new series, from the Central Control screen either click **Data** | **Key In Series** or simply type:

POP seriesname

in the process providing a mnemonic name for the series. Data-



Series: CPI Source: FRB Units: 1967 Average Disp: Linear

Description: Consumer price index,urban households

Last Revised: 6/13/99 Last Extended: 6/13/99

Quarterly Data

	I	II	III	IV
1978	188.900	193.200	197.500	202.300
1979	207.500	213.800	220.800	227.800
1980	236.800	244.800	249.200	256.300
1981	263.300	268.900	276.300	280.800
1982	283.400	287.200	292.300	293.400
1983	293.600	296.800	299.800	303.100
1984	306.800	309.600	NA	NA
1985	NA	NA	NA	NA
1986	NA	NA	NA	NA
1987	NA	NA	NA	NA

Figure 4-2. Series Update Form

View will then display the series shown in Figure 4-2, providing space for both documentation and observations.
This example uses the name CPI. If the series already exists, the available documentation and observations will be displayed, ready for you to make onscreen changes as appropriate. Otherwise the documentation fields will be blank.

The Macro Entry Facility

Alternatively, you can use the Macro Entry facility, also introduced in Chapter 3, so called because it allows you to organize the data entry as a macro containing observations on all the series you wish to include. For each series, the format is:

```
seriesname,date:obs1,obs2,....,obsn
```

where the seriesname is the mnemonic, the date is the date of the first observation, and the observations are typed one after another in the order in which they should be included. DataView will determine the dates associated with observations after the first. Note that a colon separates the date from the first observation. The observations can be separated by blanks or commas, but there should be no blanks or other non-numeric characters within a series name, date, or observation.
.

For each series, you must start on a new command line. Each command line must be 80 characters or less and if you wish to continue the observations from one command line to another, the last observation on the line to be continued must be immediately followed by a comma. A command line can begin in the first column or after.

An example may be helpful:

```
OPEC,7001: 1.4    1.4    1.4    1.4   1.6    1.8    1.8    1.8,
           1.8    1.8    1.9    2.0   2.1    2.4    2.8    3.5,
           9.2    9.6    9.8   10.4  10.5   10.5   10.5   11.5,
          11.5   11.5   11.5   11.5  12.1   12.1   12.7   12.7,
          12.7   12.7   12.7   12.7  13.9   17.0   20.2   23.5
OPEC1,7001: 1.2   1.3    1.4    1.5   1.6    1.8    1.9    1.8,
            1.8   1.8    1.9    2.0   2.1    2.4    2.7    3.2
```

This method of data entry also allows you to include series documentation, which can be done in either of two ways. If you wish to document the series *before* you add the observations, the process is known as *defining a series* and takes the following form:

DEFINE SERIES: seriesname
DESC: series description
SOURCE=code,**UNITS=**code,**FTC=**code,**DIS=**code,**FREQ=**code

The series name should appear on the first line, after DEFINE SERIES. The description (DESC:) can be up to 72 characters, but codes for SOURCE and UNITS must be kept short (up to 8 characters for the source and 4 for the units). The FTC is the Frequency Transformation Criterion used for automatic frequency conversion and DIS is the conversion criterion for lower to higher frequency. FREQ is the series frequency, and can be established by default—as the same as that currently set. For example, consider:

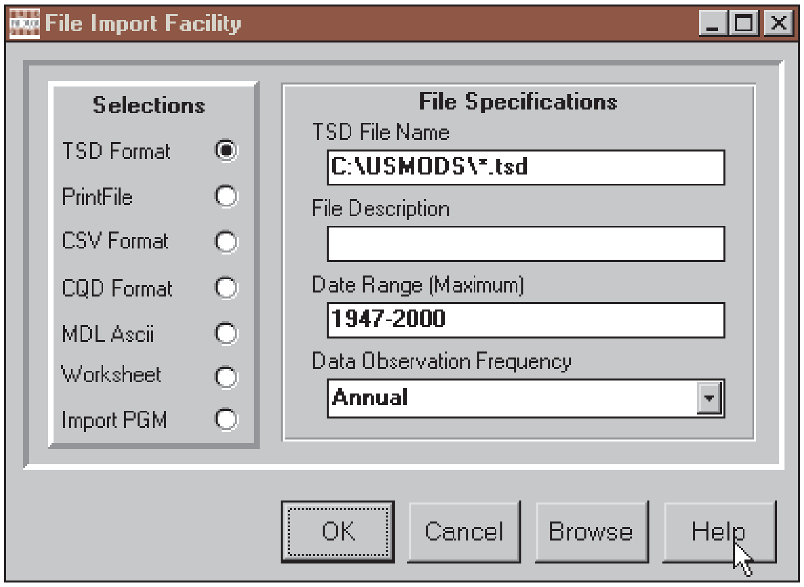
DEFINE SERIES: CPI
DESC: CPI, urban households, 1967=100
SOURCE=FRB,
UNITS=indx,**FTC=**avg,**DIS=**line,**FREQ=**quar

If, alternatively, you wish to include the observations before you document the series, then the procedure is known as *Documenting a Series* and takes the following form:

DOCUMENT: seriesname
DESC: series description
SOURCE=code,
UNITS=code,**FTC**=code,**DIS**=code,**FREQ**=code

Procedurally, the only difference is the use of the word DOCUMENT instead of DEFINE SERIES. The DEFINE SERIES command must not be used two or more times for a given series; once defined a series cannot be redefined. In contrast, there is no limit on the number of times a series can be documented—or redocumented.

The DEFINE SERIES and DOCUMENT commands can be included in the same macro files as the data entry commands, provided that any DEFINE SERIES commands come first. Alternatively, they can be put into one or more separate macros; there is no need to mix them.


File Import Facility

Selections
TSD Format [✓]
PrintFile
CSV Format
CQD Format
MDL Ascii
Worksheet
Import PGM

File Specifications
TSD File Name: C:\USMODS\*.tsd
File Description:
Date Range (Maximum): 1947-2000
Data Observation Frequency: Annual

Figure 4-3. File Format Conversion

The IMPORT facility

One of DataView's most useful data entry facilities takes any of several forms. One method is to click on **Data** and then **Import Data File** from the Central Control screen menu, which causes the form shown in Figure 4-3 to be displayed:
Context sensitive help is available by clicking on the **Help** button, which then provides details on the particular conversion process for each type of import file format.

Alternatively, from the Central Control screen command space you can enter a command of the form:

IMPORT programname

or

IMPORT programname filename

depending upon the data source. The alternative forms of the command cause the execution of a variety of data conversion programs, including programs to import data from LOTUS worksheet files, TSD format files, LOTUS Printfiles, mainframe downloading routines and many others. Additional programs can be written relatively easily, so that the possibilities are open-ended. Prototype conversion routines are available at the MODLER.COM website.

The specific characteristics of the data import procedure varies from data source to data source. However, all have the characteristic that once the file format is accounted for the IMPORT command will cause data series observations to be imported and stored in a Data Bank or a MEMORY FILE, as appropriate. In some cases, series documentation will be imported with the data;

in others it will not and you will need to document the series separately. For specific details, consult your data base vendor.

Copying Series from an Existing Data Bank or Memory file

Another way to include data series in a new Data Bank is to copy from an existing Data Bank or Memory File. The basic COPY command takes the form:

COPY SERIES=*x-y* **FROM** *file1* TO *databank*

Here x-y denotes a range of series; for example A-ZZZ, where this choice indicates that series with names beginning with the letter A through ZZZ should be copied. The range x-y can be defined in terms of either series names or series numbers (you may know there are 233 series to be copied: x-y might be 1-233). The designation file1 denotes either the name of the Data Bank from which the series should be copied or the word MEMORYFILE, if this is the source. The designation data bank should be the name of the new Data Bank or, more generally, the name of the destination data storage file. For example:

COPY SERIES=A-ZZZ **FROM** DEMOBANK **TO** PRICEBNK

The first time a series is copied, all the existing documentation on that series will be copied also. Subsequently, only the observations will be copied. In either case, which observations are copied depends on the last SET FREQUENCY and SET DATES commands executed; be sure to set FREQUENCY and DATES just before you copy series to make sure the right observations are copied.

Copying Groups of Series

You can also copy groups, using the command

```
COPY GROUP=groupname  FROM DEMOBANK TO PRICEBNK
```

Groups can contain series that have some affinity apart from the alphabetic ordering of their names. This command presumes the existence of a groupfile of the given name, with an extent .GRP, as previously described.

Updating the Data Bank

The description of the **COPY SERIES** command just above specifically refers to a new Data Bank. However, copying series is a common procedure when using existing Data Banks as well. The general rules are easily stated: If you want to fetch data from a Data Bank (read-only), use **ACCESS** to open the bank. If you want to move data into a Data Bank (write-only), open the bank with the **STORAGE** command.

For example, to copy all series beginning with WP from DEMO-BANK into the PRICEBNK bank you could type

```
ACC DEMOBANK
STORAGE PRICEBNK
COPY SER=WP* FROM DEMOBANK TO PRICEBNK
```

Note the use of the wildcard * operator in WP*.

It is also possible to copy only certain series. For example, to copy individual series from the Memory File, type:

```
**CLOSE ALL BANKS**
**STORAGE** PRICEBNK
**STORE:** CPI
**CLOSE** PRICEBNK
```

Of these commands, the **STORE:** command both causes CPI to be retrieved from the Memory File and then to be stored in PRICEBNK, the STORAGE bank. In this case, you can be sure that the data values come from the Memory File since the STORAGE and STORE commands are preceded by a CLOSE ALL BANKS command, thus insuring that no ACCESS bank will be open. Otherwise, DataView will copy series from wherever it first finds them, be it another Data Bank or the Memory File, storing them in whatever bank is designated as the STORAGE bank.

To open a Data Bank for retrieval and storage simultaneously (read-write), you use the UPDATE command. You will need this command if you want to transform data already contained in the Data Bank and if you do not want to use the STORE command; which after all involves two steps. For example, to rebase a price index already in the PRICEBNK, you could use the commands:

```
CLEAR MEMORYFILE
UPDATE **PRICEBNK**
CPI=CPI/100
CLOSE **PRICEBNK**
```

The essence of the concept of UPDATING a Data Bank is that data must be retrieved, revised or transformed, and then re-stored. Only the **update** command allows selected observations on a particular series to be changed without changing others within the update time interval.

Some Comments on Series Descriptions

The manner in which series are described is important both in terms of the use of the FIND command and for printouts of Data Bank indices. You should experiment with the FIND command as you build the bank to make sure that series are easy to find by concept. Similarly, it may be useful from time to time to produce a hard copy of the fully documented Data Bank index, just to see how the descriptions scan.