Basic FAME Instructions

Structures

Comments

Any line proceeded by a double hyphen "--" is a comment. It can also follow either:

Command command_options command_arguments

Command arguments are often separated by commas if multiple arguments are allowed.

Command command_options arguments

These do the main tasks. The format is always:

Commands

Any line proceeded by a double hyphen "--" is a comment. It can also follow a

Options

Options generally provide details for commands, and frequently associated with

Option option_arguments

Options are universally for all commands following it:

Option option_arguments

Options do not apply only to a specific command, but to all commands thereafter.

Example:

For example, if you want to report a variable to two decimal places you can type either:

report <decimal 2> e_eu, e_rc

Or:

report e_eu, e_rc
decimal 2

For example:
For the purposes of graphs and reports there is a standard set of options that we normally use:

- abort on
- ignore on
- pause off
- length full
- show vertical object precision decimal 2

What These Options Do:

- **abort on**: the program will crash if there is a mistake in the FAME code.
- **ignore on**: any mathematical operation on the data will proceed, even if there are missing data points.
- **length full**: uses the full page for graphs and reports.
- **show vertical object precision**:
  - **decimal 2**: when displaying the data, all data is shown up to two decimal places (even if those places are each zero).

Functions

Of course, there are also a whole array of mathematical functions that are good for a variety of purposes:

- **display series name en (en)**: displays the name of the series, e.g., "e\textsuperscript{en}"
- **display string (en)**: displays the data, e.g., "e\textsuperscript{en}", to type the average value we would use:

**functions**

0.6223 is the exchange rate in 2008

would display

\texttt{display string\"en\"[2002\}]+\texttt{\"is the exchange rate in 2008\"

These are also a number of useful string functions that are good for a variety of purposes:

- **type (en)**: so if we have a time series e\textsuperscript{en}, to type its average value we would use:

Functions

Of course, there are also a whole array of mathematical functions that are of the form:

- **decimal**: when displaying the data, all data is shown up to two decimal places (even if those places are each zero).
- **object precision**: in calculations, all data uses as many decimal places as possible.
- **command to start a new page**: commands that start a new page.
- **save file**: when prompting to the screen, the output will not pause with a "hit any key" prompt.
- **show vertical**: data is shown in columns (better for long time series).
- **length full**: uses the full page for graphs and reports.
- **show points**: shows points.

Files

FAME commands can be typed in by hand, but it is more useful to use files to run a bunch of commands. These are input files (such as graphs.inp in GIMF, which runs some reports) which are just groups of commands.

When These Options Do:

- **decimal**: when displaying the data, all data is shown up to two decimal places (even if those places are each zero).
- **object precision**: in calculations, all data uses as many decimal places as possible.
- **show vertical**: data is shown in columns (better for long time series).
- **length full**: uses the full page for graphs and reports.
- **save file**: when prompting to the screen, the output will not pause with a "hit any key" prompt.
- **show points**: shows points.

For the purposes of graphs and reports there is a standard set of options that we normally use:
There are also procedures, which are files which are called, usually with extra arguments supplied. For example, to run graphs and tables in GIMP we use the procedure file graphs.pro.

First we compile it in Fame using the compile command:

```
compile graphs.pro
```

And then we load the resulting Fame-readable file:

```
load graphs.pc
```

Graphs.pro contains many procedures which we can call by name. For example, we create steady state reports in 2001 using the control database with the procedure call:

```
report en
display en
type en
```

You can report a series from the open databases by either typing, displaying or reporting.

**Reporting data directly**

FAME always has a database available and ready to write to. Its database alias is work.

Simple databases more than once.

Databases simultaneously. Provided you are only reading the database, you can open a database alias, you can open up to fifty databases simultaneously. The word “con” is what is known as the database alias. You can open up to fifty databases simultaneously.

You could also create it simply by typing:

```
create ss.db
```

The word “con” is what is known as the database alias. You can open more than once.

You can open a database, if you create a database, it assumes the database does not yet exist.

Use the open command, with the access option. You can either [r]ead, [u]date or [c]reate a database.

```
open <acc r> "ss.db" as con
open <acc up> "ss.db" as con
open <acc create> "ss.db" as con
```

You could also create it simply by typing:

```
create ss.db
```

**Accessing data directly**

```
report en
display en
```

To define the date range, use the option date. Once you have defined the frequency of the data, you can define the dates with the option frequency. Dates can be expressed as [a]nnual, [q]uarterly, [m]onthly, [d]aily, or [b]usiness.
This will define the data as annual data, and you will report it from 2000 until 2003. Although GIMF can be either an annual or a quarterly model, we use the annual frequency for data, because it is more convenient when building complex tables and graphs.

Here is the output of the three commands:

```
deci 4
type  e_eu
0.6223
display e_eu
E_EU
2000 0.6223
2001 0.6223
2002 0.6223
2003 0.6223
report eu
E_EU
----
2000      0.6223
2001      0.6223
2002      0.6223
2003      0.6223
```

Notice that the `type` command only shows the first observation of the series, without mentioning the period.

The `report` command can also be used to build more complicated labels. For example, let's report e_eu, first as a horizontal table, and then as a vertical table. To do so, we open a report with the `report` command, then define the columns with the `select` command and the rows with the `print` command.

So here is the output of the three commands:

```
0.6223
deci 4
```

**Grateful:**
The Structured Graph allows us to enter extra commands. In this case, we are adding a title to the graph, plus drawing a line at 0.7 on the graph.

The structured graph allows us to enter extra commands. In this case, we are adding a title to the graph, plus drawing a line at 0.7 on the graph.

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

There is the simple graph:

date 2000 to 2020

date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:

We can also graph the data using the graph command. Assume we have specified the date range:

date 2000 to 2020
date range:
Nominal Exchange Rate

E_{EU} = \text{dzero} + 0.68756071761638