A Software Development Summary for the 2008 Inforum World Conference

Ronald Horst[1]

Abstract

This article summarizes work at Inforum over the past year on the G7 *software. We document recent development of* G7*, the database and econometric software package developed at Inforum with the assistance of its partners. We also announce the addition of a new section of the Inforum web site. This new section features* G7 *scripts to perform useful tasks and to demonstrate* G7 *capabilities and economic methods.*

1. Introduction

Since the last annual report on Inforum software, distributed in 2007 at the 15th Inforum World Conference in Trujillo, work has continued on *G7*. Efforts have yielded a more reliable program with new capabilities. Inforum's flagship software program, in order to make the program more reliable and to introduce new capabilities. The Inforum web site[2] also has been updated and extended. In particular, a new section has been introduced to feature *G7* scripts that perform useful tasks and that demonstrate *G7* capabilities and economic methods.

We begin with a review of recent developments of *G7*. In addition to the details presented in this paper, these developments are reported in the latest *G7* Help files and Reference Manual that are available on the Inforum web site.[3]

We next review a new section of the Inforum web site. This new section provides demonstrations of *G7* features and tools. It also provides guidance in the development of input-output data and models, and it guides users in the methods of applied economics. The web site content depends on contributions of Inforum partners and other users of *G7*. We gladly accept contributions of new scripts from *G7* users, and we welcome comments regarding the files posted.

We conclude with a brief review of other software development. In particular, we note updates to the *Compare* and *Banker* software programs.

In summary, work in the past year has corrected bugs and introduced several new capabilities to aid model builders and other researchers. The *G7* demonstrations now hosted on the Inforum web site should make G7 more accessible to new users and should help experienced users to learn about new features, including some of the new features presented here.

2. G7

Inforum offers the *G7* software for use in data construction and analysis, econometrics, and for the formation of large-scale structural econometric models. We summarize here most of the significant

[1] Ronald Horst, Inforum, University of Maryland, College Park, USA. Ronald.Horst@gmail.com.
[2] www.inforum.umd.edu
[3] www.inforum.umd.edu/software/g7.html

new features and extensions that were developed recently. We omit documentation of most bug fixes, though these bug fixes are quite important and have produced a program that substantially is better than before.

In addition to this document, the *G7* Help files and Reference Manual have been extended and updated. This new documentation better reports the capabilities of *G7* and guides the model builder through difficult steps of data development.[4]

If you do not have *G7* installed on your machine, please download the *G7exe.exe* program from the Inforum web site and run it. While this package does not have the latest version of each program, it will provide a fairly comprehensive set of programs. You then can upgrade particular programs, when they are available, by downloading the individual files linked at the bottom of the *G7* web page. These files should be stored in the C:\PDG directory. They will replace the files that were installed earlier. Please check the *G7* web page periodically. As software updates become available, they will be posted to this page.

Improvements to *G7* are presented in three following sections. First, we present new tools and features introduced to the *G7* scripting language. This section also lists a few of the problems that have been fixed. Second, we describe refinements of the G7 graphical interface. Third, we conclude by reporting extensions and improvements to the documentation.

2.1 Extensions to the Language

Though *G7* features a useful graphical interface, it remains primarily a script processor. The extensive *G* language offers commands and routines that allow users to create and manage databases, estimate regression parameters, and to design structural economic models.

Guides to the software often introduce the language by reporting about a dozen key routines. Indeed, learning these features will serve the model builder well. Some recent developments improve and extend these fundamental tools. Other commands and tools are brand new, and still others extend some of the many lesser, but still important, parts of the language.

Note that some of these routines are preliminary and need additional testing. See the *G7* Help files and the Reference Manual for additional details. Demonstration routines soon will be featured on the Inforum web site; see the Section 3 in this paper for more details.

We begin by reviewing extensions to the *vmake* tool. This handy feature allows data to be copied from a set of vectors and matrices into another vector or matrix.

<vmake|vplus|vminus> [startyr [endyr]] <target> <source> [source ...]

The **vmake**, **vplus** and **vminus** commands make a vector out of another vector or part of a matrix, or it makes a matrix out of one or more vectors and/or other matrices. *vmake* replaces selected values in the target with the values from the source; *vplus* adds the values in the source to the values is the target; and *vminus* subtracts the values in the source from the values is the target. Up to 200 sources may be taken from one or more assigned *vam* files.

[4] For the G7 software and for additional documentation on the material introduced here, please visit the Inforum web site at www.Inforum.umd.edu. A copy of this document and related materials may be found on the "Inforum World Conference XVI" page. The current *G7* version number is 7.3756.

The command syntax is presented above. <*startyr*> and <*endyr*> are optional parameters. If not given, then the process will work over the range of dates common to the target *vam* file and all the source *vam* files. *target* is a matrix or a vector, and sources are matrices and/or vectors. Each matrix, whether a target or a source, is represented by

```
MatrixName(<r|c> [lines])[(<c|r> elements)]
```

<*r*|*c*> indicates whether the data will be selected first by row or by column, and *lines* specifies the selected rows or columns, and *elements* specifies elements in the remaining dimension. The total number of lines selected from the sources must equal the number of lines selected from the target, and the number of elements selected from *each* source must equal the number of elements selected in the target. If lines or elements are omitted, the default is all rows/columns or all elements. Each vector, whether a target or a source, is represented by

```
VectorName [(elements)]
```

where *elements* are the items selected within the vector. If elements are omitted, the default is all elements. The number of elements selected from *each* source must equal the number of elements selected in the target. See the *G7* Help Files or Reference Manual for examples.

if([comparisons]){ [...] }
[else if([comparisons]){ [...] }]
[else{ [...] }]

G7 flow control capabilities were introduced at the 2006 Inforum World Conference. The most common flow control tool is the *if-else* statement. Since its introduction, the capabilities of the *G7* feature have been polished and extended.

The *if-else* routines evaluate single arguments or compare two arguments, and if the evaluation is true, then *G7* executes the following script. Useful comparisons allow checks for equality and inequality of numbers, checks for the existence of database entries, and string comparisons. Several conditions may be checked simultaneously through use of C++-style Boolean operators that link together various tests.

Several new capabilities have been introduced for the construction and evaluation of arguments. First, simple arithmetic calculations now can be made. Addition, subtraction, multiplication, and division are supported. For example, the following line demonstrates an equality test of a sum.

```
if( 1+2 == 3){...}
```

Modulo arithmetic also is supported.

```
if( mod(7,2) == 1){...}
```

The *isnum()* test allows the type of argument provided to be evaluated. In the following line, a value for the variable *%1* is supplied by an *add* command, *do* loop, or similar routine. The test result is *true* if the argument is a number and otherwise is *false*.

```
if( isnum(%1) ){...}
```

These routines also can perform tests on strings. Most of the tests follow the syntax of the C standard libraries. Details were provided at earlier conferences and were described in previous documentation.

An important recent change is that the return values for these tests were changed to follow the C standards. For example, the following string comparison test will return "0," indicating that the strings are identical; the return values in earlier versions of *G7* was 1 (*true*).

```
if( strcmp("string", "string") ){...}
```

Because the strings match, and the test returns *0*, the block of code following this command would not be executed. To execute a script following such a test with matching strings, the "!" logical negation operator should precede the string comparison routine and change the resulting *false* code to *true* (that is, change the *0* return value to *1*).

```
if( !strcmp("string", "string") ){...}
```

Another string tool has been added to allow calculation of string length.

```
if( strlen(%1) > 3 ){...}
```

Finally, the presence of a series in a data bank can be tested with the *exists()* function. To test whether variable *x* exists in bank *a*, use the following command.

```
if( exists( a.x ) ){...}
```

The command returns *true* if the series is found.

xl read < column > < row > < direction > < series > [< start > [end]]

The *G7-Excel* interface again has been refined and extended. Several critical bugs were detected and fixed.

The time series reading and writing routines now are somewhat more flexible. They now can be read or written even when the data are in decreasing vertical order or right-to-left order. *direction* now may be *up* or *left*, extending the original support for *right* and *down*. *column* and *row* specify the starting cell in the spreadsheet, *start* specifies the starting date, and *end* provides the optional ending date. *series* is the name of the time series. Similar changes were made for the vector routines.

Other Data Entry and Printing Updates

The *G7 matty* command is used to print several time series as columns in a table. The formatting of these tables has been revised to make them significantly more legible. The *vp* command also is available to print vector data. While previous documentation indicated *vp* options for printing data as rows or columns, *G7* did not support those capabilities. This discrepancy has been corrected. The output of the *vp* command also has been modified to improve legibility. Finally, *G7*'s ability to recognize "?" as a missing value code has been extended. The *vf* command, a tool for calculating arithmetic expressions and storing them as vector elements, now will store a missing value if "?" is provided on the right-hand side of the equation. This capability already existed in the corresponding time series tool, the *f* command.

fadd <CommandFile> <ArgumentFile>

The *fadd* command executes the script file named *CommandFile*. As *CommandFile* is executed, arguments within the file are set with values from the first line of named *ArgumentFile*, then the *CommandFile* is executed again with the arguments from the second line of the *ArgumentFile*, and so on until the lines of *ArgumentFile* are exhausted.

Several improvements have been made. First, the file names now may be provided within quotation marks. This allows full and partial paths to be specified. For example, *G7* will find the following files only if they are provided with quotation marks.

```
fadd "..\directory\commandfile.fad"  "123ArgumentFile.fin"
```

The number of arguments per line that are permitted in the *fadd* argument file has been increased to match the capabilities of the *add* command. This number is 99 arguments.

Several improvements have been made to the parsing of *fadd* command and argument files. Comments now are permitted in argument files, and negative numbers now may be provided as valid arguments. Finally, the *break* command now may be issued effectively within the command file to halt execution of the script.

f <variable> = @function( )
vf <vector element> = @function( )

A number of new functions have been introduced to simplify and automate particular calculations. These tools are used in conjunction with the *f* command, for calculation of time series data, and with the *vf* command, to calculate vector elements. The following routines typically can be used either with the *f* or *vf* commands. The first set of functions reported here perform aggregation to convert data series from high frequencies to lower frequencies. The second set is a collection of assorted routines to perform specialized tasks.

The following functions convert the frequency of a variable by aggregation. While *G7* previously possessed some of these capabilities, others required multiple lines of code or simply were impossible to implement easily.

@<high>to<low>(x)

converts the high-frequency series *x* to a lower-frequency series by forming the arithmetic mean,

@<high>to<low>e(x)

converts the high-frequency series *x* to a lower-frequency series by applying the end-of-period values,

@<high>to<low>max(x)

converts the high-frequency series *x* to a lower-frequency series by taking the maximum,

@<high>to<low>min(x)

converts the high-frequency series *x* to a lower-frequency series by taking the minimum,

@<high>to<low>s(x)

converts the high-frequency series *x* to a lower-frequency series by taking the sum,

where <*high*> and <*low*> must be replaced with *m* (monthly), *q* (quarterly), *s* (semiannual), or *a* (annual). For example,

```
f y = @mtoa(x)
```

converts the monthly series *x* to annual frequency by forming the average of monthly values. Note that sums and averages are formed over non-missing values, so that missing values are ignored.

The next set of functions adds assorted capabilities to *G7*.

@hpfilter(x [, lambda])

The function *@hpfilter()* implements the Hodrick-Prescott filter for the series *x*. Default values for lambda are 14400 for monthly data, 1600 for quarterly data, and 100 for annual data. These values commonly are used in economics literature.

@bmk(x,y,[type])

The benchmark function adjusts series *y* so that *y* equals to the non-missing values of *x*. *type* either may be *d* (default) or *g* (growth rate). The benchmarking procedure first scales *y* so that it equals the first non-missing value of *x* in the corresponding period. The default process then calculates the difference between the next non-missing value of *x* and the adjusted value of *y*. This difference is allotted to *y* in each intermediate period in increasing proportions, so that the linear adjustment to *y* leaves it equal to the second non-missing value of *x* in the corresponding period. This process continues with each subsequent non-missing value of *x*. Values of *y* after the last period of non-missing *x* are scaled but otherwise the pattern is not changed from the original series.

An alternative benchmarking process now is available. The procedure is identical to the default method before the first non-missing value of *x* and after the final non-missing value of *x*. The difference is in the adjustment to *y* for periods between non-missing values of *x*. First, the geometric average growth rate is calculated for *x* and for *y* between the first and second non-missing values. Next, the difference in the average growth rates is calculated, and this result is used to adjust the growth of *y* such that its resulting geometric average growth rate is equal to that of *x*. This procedure ensures that *x* and *y* will coincide for the following non-missing value of *x*. The process is repeated for each remaining non-missing value of *x*. In some cases, this procedure produces results that are more reasonable than those created by the default procedure.

VAM File Routines

A number of *vam* routines for handling databases of vectors and matrices have been improved. Several such improvements are listed here.

The *minv* routine for inverting matrices has been improved in two ways. First, the internal calculations now are done with double precision, though the initial matrix and the result are stored in single precision. Second, the original inversion algorithm has been replaced by the method published in Numerical Recipes. The results now should be more accurate and the procedure more stable.

A serious bug fix was made in the division operator of the *vc* command. The bug would appear only in rare cases. It required that the / operator be applied to a matrix and a vector, where the intended result is the multiplication of the matrix transpose by the vector. The bug also required very small negative values to be present either in the matrix or the vector, and that these values be interpreted as missing values. In these cases, results would be flawed both for the corresponding element and for other elements in the left-hand side vector. When missing values appear now, they simply are skipped. Note

that the result is *not* set to *MISSING*, but rather the missing right-hand-side values are assumed equal to zero. Note also that internal calculations with this routine now are performed in double precision.

Finally, a search routine has been added for packed matrix files. The routine is similar to the one for vector and matrix title files. When a packed matrix file is needed, *G7* first looks for it in the traditional position relative to the current working directory. The new capability is implemented if the file is not found with the initial attempt. In such cases, and when the position of the file is provided with a relative path or without a path, then *G7* again looks for the file. This time, it essentially moves the search position from the working directory to the location of the *vam* file, and then it attempts to follow the relative path provided. If the desired file is found, it is loaded and employed. Otherwise, *G7* gives up and reports failure. The same capability was documented earlier for title files. These search features make it easier for the user to store a single set of title files and packed matrix files with each *vam* file, but to have the freedom to load the *vam* file and to link its corresponding files from other locations.

2.2 Interface Improvements

Several improvements have been made to the *G7* graphical interface.

First, *G7* graph titles now have empty space removed at the end of the string. Also, pesky buffer overflow problems with various graph titles and data bank titles have been fixed.

Second, an unpleasant bug was fixed that previously appeared during a particular sequence of operations with the show command. This problem caused *G7* to crash. While this specific problem has been fixed, a similar bug has been reported but remains at large.

Third, we have witnessed some objectionable behavior of certain dialog boxes. Some windows appear but are not assigned the focus, so that the user must navigate to the new box in order to proceed. A much worse case sometimes occurred with a new window appearing behind another window, and the first window could not be moved to obtain access to the second window. These problems affected the editor's “Error” box that appears after a fruitless text search and with the graph control bar. Similar problems appeared elsewhere. At least in most cases, these problems have been fixed.

2.3 Documentation Descriptions

The G7 Help files and Reference Manual have been updated. They are available on the Inforum web site on the G7 page. The Reference Manual and other resources may be found at the top of the page, and the Help files and latest software may be found at the bottom of the page. Please download the files listed there and save them in your C:\PDG directory. When they become available, future updates to the documentation and the software will be posted on the web site.

3. G7 Demos on www.inforum.umd.edu

G7 demonstration files can help *G7* users to implement complex tasks and can demonstrate the capabilities of the software. Currently, there are three demos available on the Inforum website. The collection will be extended as *G7* users develop and contribute materials. The current collection can be

found by following a link on the G7 software page.[5] We provide here a brief summary for each.

Basic G Commands: G7 is a powerful tool with which a user can carry out sophisticated tasks when constructing economic data sets. A user new to the *G7* software package may be overwhelmed by the extensive set of tools available. This demo script provides the user with a solid foundation on which to build his expertise. After running through the demo script, a user will be able to load and assign a bank; assign *fdates*, *tdates*, and *gdates*; print data to the screen using the *type* command; graph data; form unique variables; add data to the workspace; and view a stub file.[6]

Use G to Read and Write in Excel Format: A recent addition to *G7* is the ability to read and write in Excel format. Data can be read from Excel files and written to the G7 workspace data bank or to a *VAM* file. Data also can be copied from *G7* databanks to existing Excel files. There currently are five demos that will assist the user in gaining a mastery of the Excel commands.

Balancing Matrices with Uncertain Totals: Mike Field has created a *G7* script that demonstrates its ability to balance matrices with uncertain totals using the *psras* command. The following is an excerpt from his description. "With the special use of proportional-scaling RAS outlined below, it is possible to balance a matrix for which one or more row and\or column totals is not known with any degree of certainty. The method, which allows these uncertain totals to change during the balancing process, opens up a row and\or column and moves the uncertain totals inside the matrix, so that these totals are treated as part of the matrix to be balanced. This depends critically on the ability of proportional scaling to scale rows and\or columns that sum to zero. Finally, it is possible to use the precondition option in the *psras* command to control the change in any of the uncertain totals."

The current set of demos provide only a brief glimpse of *G7*'s capabilities. More scripts will be added as they become available. We welcome suggestions for demo topics and submissions of completed scripts.

4. Other Software

In addition to *G7* software, Help files, and other documentation, the *G7* page on the Inforum web site provides current editions of other modeling software. Recent updates include bug fixes for *Compare*, a program for publishing data, and updates for *Banker*, a program to compile databases.

5. Conclusions

A number of extensions have been introduced recently for *G7*. The latest version of *G7* eases the task of data development and manipulation with new tools for data aggregation, trend extraction, conditional code execution, and more. In addition, most problems that have been reported also have

[5] The page may be found at www.inforum.umd.edu/software/demos.html.
[6] Note that the *G7 Tutorial* also is available on the *G7* web page.

been fixed.

Our partners in Italy, Japan, and Poland deserve particular recognition for submitting reports of their experiences and for suggesting improvements. Mike Field again has contributed in many ways, including the contribution of extensions to the *vmake* routine, creation of documentation and code for extended scaling and balancing routines, and submission of demo scripts for the web site. Thanks also to Troy Wittek for his help in preparing this document. We welcome the suggestions and participation of Inforum partners in the continued development of these tools that are essential to our work and to the development of the models we share.