

# IT at Inforum in 2007

Ronald Horst<sup>1</sup>

## Abstract

*This article briefly summarizes work at Inforum this past year on information technology. In the past year, Inforum launched a redesigned and extended web site. In addition, development has continued on G7, the database and econometric software package developed by Inforum and its partners. Finally, extensions and improvements have been made to other software packages developed and maintained by Inforum and its partners.*

## 1. Introduction

Since the report on Inforum software presented last year at the 14<sup>th</sup> Inforum World Conference in Traunkirchen, work has continued on several fronts. The most significant change is that the Inforum web site that updated, heavily revised, and extended. Significant improvements also have been made to Inforum's flagship software program, *G7*, though in contrast to the sweeping changes to the web site, the software modifications consist of refinements to the graphical interface, extensions to the scripting language, and bug fixes. Finally, large-scale implementation of Inforum's revised interindustry modeling package, *Interdyme*, has begun, bringing with it needed development of the model-building code and supporting software.

We begin with a review of the new web site, going through each of its major components and features. The following section presents the most recent version of *G7* and the accompanying help files and other documentation. We conclude with a brief review of other work, and in particular work on the collection of C++ classes known as *Interdyme*.

In short, recent developments correct flaws and strengthen weaknesses in previous work. While much remains to be done, the software developments strengthen the products and services offered by Inforum and its partners. The improved web site should prove more attractive to potential customers, and interested parties easily can find extensive and current information on Inforum and its literature, products, and consulting services. A key improvement is that visitors now have ready access to information about Inforum partners around the world.

## 2. [www.Inforum.umd.edu](http://www.Inforum.umd.edu)

In August 2007, Inforum launched a new edition of its web site. As the first significant revision in many years, the work brought much-needed improvements to this critical tool for advertising the capabilities of Inforum and its partners, for distributing software, and for disseminating information about the organization and its research.

---

<sup>1</sup> Ronald Horst, Inforum, University of Maryland, College Park, USA. [Ronald.Horst@gmail.com](mailto:Ronald.Horst@gmail.com).

The site is organized in five major parts. They are the Organization, Research, Products and Services, EconData, and Software.

The Organization block contains information about the organization and its staff. Most importantly, the site now features web pages for each of Inforum's partners. These pages were made possible by the contributions and cooperation of many colleagues. The partner pages provide contact information, links to partners' web sites, and links to papers and materials produced by partners and hosted on the Inforum site. These pages will be updated indefinitely as Inforum partners provide corrections and extensions. Finally, the Organization block features pages for each of the 15 Inforum world conferences.

The Research block provides links to several tables. Each table presents a collection of papers and other materials. The tables are Working Papers; Published Works (including books, dissertations, and other work); Other Studies (work published by other organizations with contributions by Inforum); Inforum World Conference presentations, papers, and other materials; and archives of materials from several Input-Output Association conferences. Each table lists the contributor, the title of the work, and a link to the PDF or zip file.

The Products and Services block currently includes three sets of information. The first two describe Inforum's Forecasting work and its Consulting and Policy Analysis services. The final collection of pages describe various models developed and maintained by Inforum.

EconData remains largely unchanged from the version that featured on the previous site, but the data has been updated and again is maintained regularly. The list of data series now featured is somewhat more limited than before, but the value of consistently updated key series was deemed more important than presentation of a wider array of data.

The Software block currently features three sets of pages: one each for *G7*, *Build*, and *Interdyme*. The *Build* page now offers the small U.S. model *AMI* for free download. The *Interdyme* page currently does not offer code for download, but it does offer a brief summary with links to other documents. The *G7* page offers *G7* and related software, along with a collection of documentation, for download in the form of an installation package. In addition, the *G7 Reference Manual* may be downloaded separately. The most recent versions of *G7* and its help files, *Build* and *IdBuild*, *Fixer* and *MacFixer*, and *Compare* can be downloaded separately. The *G7* section also provides a page for sample scripts that demonstrate tools and techniques; the list of available demonstration programs will grow as submissions are received.

An invaluable addition to the site is a search tool. This tool indexes HTML, PDF, and Word documents hosted on the site. The addition should simplify and speed the hunt for particular topics or work by a particular author.

Additional work and new features are planned. An important possibility is the addition of dedicated search tools for the documents hosted on the site. The planned tool should offer more extensive capabilities, though over a more narrow range of materials, than the current search tool.

### 3. G7

*G7* is the flagship tool offered by Inforum for data construction and analysis, econometrics, and for the formation of large-scale structural econometric models. Together with earlier versions of the software, *G7* has been under continual development and in real-world use for many years. We summarize here work on a variety of extensions to *G7* that recently have been completed, along with a number of other improvements. While even collectively the results are less than revolutionary, the work leaves *G7* more reliable and powerful, simultaneously lessening the burden on the model builder while extending his capabilities. In addition, 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.<sup>2</sup>

Recent improvements to *G7* are presented in four following sections. First, we present extensions to the *G7* scripting language. Second, we summarize extensions and revisions to the *G7* graphical interface. Third, we report extensions and improvements to the documentation. Finally, we conclude by listing a few of the problems that have been fixed.

#### 3.1 Extensions to the Language

While a handful of basic tasks can be completed through *G7*'s graphical interface, the full extent and power is contained in its scripting language. This large collection of commands and routines allows the user to create data banks and fill them with data, construct regression routines, and put together pieces of 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. Others commands and tools are brand new, and still others extend some of the many lesser, but still essential, 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 previous section in this paper, [www.Inforum.umd.edu](http://www.Inforum.umd.edu), for more details.

We begin by reviewing a series of new functions for one of *G7*'s key routines, the *f* command. This arithmetic tool constructs a time series of data based on the equation provided by the user, and the result is stored in the workspace data bank for later use.

---

<sup>2</sup> For the *G7* software and for additional documentation on the material introduced here, please visit the Inforum web site at [www.Inforum.umd.edu](http://www.Inforum.umd.edu). A copy of this document and related materials may be found on the "Inforum World Conference XV" page. The current *G7* version number is 7.373.

**@if( <expression1> <{< <= == >= >} > <expression2>, < result if true>, < result if false>)**

This new function for the *f*, *fex*, and similar data routine allows the calculated data series to be conditional on a period-by-period comparison of two expressions. For periods in which the condition is true, the first of two following results is returned for the corresponding period. If the condition is false, then the second result is returned. Both expressions and both results can be any legitimate script for the right-hand side of *G7*'s *f* command. Hence, each term either can be a constant or a time series. At this point, only <, <=, ==, >=, and > operators are recognized as legitimate. Additional operators, such as negation, AND, and OR will be supported in the future. An example is `f y = @if( x+z>0, x+z, -(x+z) )`

**@yoy( <expression> )**

The year-on-year growth rate function computes growth rates of the series provided. In contrast to the existing `@gr()` routine, which computes growth rates based on the current and previous periods, this routine computes growth rates based on the current period and the period one year ago. The routine accounts for various frequencies and supports monthly, quarterly, and annual data. An example is `f gdp_yoy = @yoy( gdp )`

**@ggr( <expression>, <start> <end> )**

The `@ggr()` computes the geometric growth rate (a constant) between the two specified periods. The result is recorded for each period within the *fdates* parameters set elsewhere. An example is `f gdp_ggr = @ggr( gdp, 1990, 2005 )`

**@min( <expression> [, <start>, <end>] ), @max( <expression> [, <start>, <end>] )**

These functions compute the minimum or maximum values for an expression, where the expression is evaluated either between the two specified parameters or between the *fdates* if no parameters are given. The result is recorded for each period within the *fdates*. An example is `f peak = @max( investment, 1990, 2005 )`

**@qchwt( nominal, quantity, desired base date [, zero])**

**@pchwt( nominal, quantity, desired base date [, zero])**

These routines are variations on the existing tools to construct chain weighted indexes. The `@qchwt()` function returns an aggregate quantity index, and the `@pchwt()` function returns an aggregate price index. Two groups of input series are specified: a group of series in nominal levels, and a group of series that are chain-weighted indexes. Note that both this routine and the original chaining routines now support groups of up to 1000 data series. A base date must be provided in order to scale the result, but it need not be consistent with the base date of the source data. Finally, a “zero” option indicates whether missing values should be interpreted as true zeros. At present, it also controls the routine's ability to skip zero aggregates that may precede the actual data and following the actual data; this problem occurs when the *fdates* interval is too wide. If these features can be made dependable, then the detection routine need not be optional and may be made standard. The routine adds three values to the workspace: *chwqi*, *chwpqi*, and *chwni*, which are the aggregate quantity index, the aggregate price index, and the nominal

aggregate, respectively. Please keep in mind that this routine is brand new, that it needs additional testing, and that the syntax and specification for this command are subject to change. An example is `pce_real = @qchwt( pcez(1-92), pce(1-92), 2000, z )`

The following text provides documentation for an assortment of new and modified capabilities for *G7*.

`vf <vector series> = <expression>`

`vf <vector series>{<date1> [ - <date2>]} = <expression>`

The *vf* command has been extended to allow the temporary setting of *fdates*. If the vector series on the left-hand side of the equation is followed by a date or a range of dates surrounded by brackets, then the operation is carried out for those dates only. This capability already was available for the *f* command. An example is `vf q1{1980-2005} = 1`

`catch [-a|-w] <off | filename>`

`save [-a|-w] <off | filename> [option]`

The *catch* and *save* routines store a copy of result printed in the *G7* output window and a collection of equations for building a model, respectively. It now is possible to append text to an existing file by including the "-a" flag. By default, the "-w" option is assumed and a new file is created. An example is `catch -a ExtendExistingFile.txt`

`matty [file <file_name>] [<date1> [<date2>]] [<option>]`

`<ser1> <ser2> .... <serN> ;`

`save [-a|-w] <off | filename> [option]`

The *matty* and *save* routines have been given a new option called "dump" to facilitate printing of data in a compact format. When numerical formatting options are set properly using the *format* command, the printed text may be read easily by humans and imported easily as text by spreadsheet programs. An example is

```
format 15 3 5
save output.dat dump
...
save off
```

An additional format has been made available for the *save* command. The "vupdate" option prints ASCII data suitable for updating data vectors in VAM banks.

XL mkseries <frequency> [text] <column> <row> [text]

The set of *G7* functions that provide an interface with Microsoft Excel has been extended with the *mkseries* command. This “make series” function allows a new series to be created in the *G7* workspace by reading a name (a character string) from a spreadsheet. When combined with the “XL read” command, a user can read a series names from one column and corresponding data from the same row, for example. Previously, the series needed to be created in the workspace by some other means before any data could be read. In the following example, a series name is provided in cell A1 of the spreadsheet, and corresponding data begins in cell B1. Note that the series name is omitted in the “read” command. The frequency of the new series is 12 (the data is monthly), and is available from January 1976 to December 2006.

```
XL mkseries 12 A 1
XL read B 1 right 1976.001 2006.012
```

Sometimes the text provided by the spreadsheet is not quite the desired name for the new data series. The *mkseries* (or *mks*) command allows optional text to be affixed before or after the root provided by the cell contents. In the following case, the names provided by the spreadsheet actually were sector numbers for a forecast made one year ago. The objective was to read the forecast that was recorded in a spreadsheet in order to compare last year's forecast to current work. The following example describes reading information for sector two, which occupies the second row of the spreadsheet and where the cell content of A2 is “2”. The following line of code creates a series called “frs2old” in the workspace. (The rationale is that “frs” is the name of the bank and “old” distinguishes the series from the current work.) Note that the data to be read is annual, so the specified frequency is 1.

```
XL mks 1 "frs" A 2 "old"
XL read B 2 right 1998 2012
```

Finally, note that the syntax of these commands usually allows the commands to be embedded in a loop. In the example above, row and column values were replaced with arguments, and a *do* loop was employed to create series and to read data quickly for nearly 300 sectors.

*fdates* <±*n1*> <±*n2*>

*lim* <±*n1*> <±*n2*> [<±*n3*>]

The *fdates* and *limit* commands have been extended to improve scripting capability. For each command, each date can be adjusted one or more periods forward or backward. The capability is useful for subsets of code for which the appropriate dates are similar to, but not exactly the same as, the dates generally appropriate. The following example adjustments regression limits for the observation lost when computing growth rates. Note that valid dates must be set before they can be adjusted in this way.

```
fdates 1980 2006
lim 1980 2006 2010
r y = x
```

```
f y_gr = @gr( y )
f x_gr = @gr( x )
lim +1 +0 +0
r y_gr = x_gr
```

**try{...} catch{...}**

The *try* command must be followed by the *catch* command. These routines loosely clone similar routines in C++. Within the brackets following each command may be any collection of *G7* routines, including other *try-catch* blocks. If execution of any of the commands within the *try* block produces an internal error signal, then execution of the code in the *try* block ceases and execution of the code in the *catch* block begins. Otherwise, the *catch* block is ignored. For example, consider the link series command *ls*. The routine will fail if the value of the guide series is zero in the base period. Ordinarily, this will cause *G7* to stop execution of the script and report an error. If, on the other hand, the *ls* command is nested within a *try* block, then if an error occurs the alternative block of code following the *catch* command will be executed. In the following example, suppose that we prefer to use guide series *y* when it is available, and when it is not we rely on guide series *z*. We first try to extend series *x* using series *y* beginning in 2005, and if the value of *y* is zero in 2005, then we repeat the exercise to extend *x* using series *z*:

```
try{
  ls x y 2005 f
}
catch{
  try{
    ls x z 2005 f
  }
  catch{
    ic Sorry. No data in y or z.
    pause
  }
}
```

The *try-catch* routine works well with many other commands, and has a wide variety of useful applications. Note again that *try-catch* blocks can be nested, so that several alternatives can be tried before *G7* gives up in despair.

### **eqpunch <filename> | “off”**

The *eqpunch* command is for writing equation results to a file in a tabular format. We follow the precedent of giving those files an “.eqp” extension, but of course any file name will do. The use of the *eqpunch* command is analogous to the *ipch* command, and in fact both are often used in the same regression file. The *eqpunch* command sets up a file for subsequent writing by the *titpch* and *tpch* commands. The *titpch* writes out a header for the table, and the *tpch* command writes out regression results, one line per sector. See the documentation on the *tpch* command for a complete example.

**titpch [-<options>] [<stub\_len>] ["str1" "str2" .."strn"]**

The *titpch* command defines a header line that will be provided for a regression table, and also defines the regression output that will be displayed on subsequent “tpch” commands. Before using either the “titpch” or “tpch” commands, an equation table punch file must first be opened using the “eqpunch” command.

The <stub\_len> is a number that indicates how long should be the “stub” or title for the line. You should ensure that the width you specify in the “tpch” command is the same, so that the table will line up properly.

The option string starts with a ‘-’ character, and may contain one or more of the following:

- b: rbar-squared
- o: rho
- r: r-squared
- s: see
- d: double-line format
- f: floating point f format, instead of g format.

"str1", "str2", .., "strn" are strings to be printed to explain the coefficients. Therefore, "str1" might be "intercept", "str2" is the second explanatory variable name, and so on.

**tpch [<sur which>] <sector> [<"label">] [str\_len] [(coef numbers)]**

The *tpch* command writes out a line of an equation table file. In order to use this command, a file first should have been opened using the *eqpunch* command and a header definition supplied by the *titpch* command.

[<sur which>] is used only when an equation has been estimated with the *stack* or *sur* commands to that the coefficients are in *rcoef1*, *rcoef2*, etc. <sector> is the number of the sector or category for which the equation is estimated. <"label"> is a sector or category title in quotes. If used, you also should provide the <str\_len> argument to specify a length for printing. <str\_len> is the length of the sector or category label in the printout. Note that this should be equal to the “stub length” specified in the *titpch* command. [(coef numbers)] are used when there is a superset of regression parameters possible, and each equation uses some subset of that.

An example is

```
eqpunch ven.eqp
titpch -rsf 30 const use usedif
```



```

add vena.reg 1 "Oilseed farming"
...
eqpunch off

```

where the contents of the “vena.reg” file are

```

ti %1 %2
subti Inventory Change Regression
f usedif = use%1-use%1[1]
r ven%1 = use%1, usedif
ipch ven %1 a
tpch %1 "%2" 30

```

## zip [off]

The simple syntax of the humble *zip* command, which determines whether *G7* should pause after each graph is created, has not been changed. Its behavior, however, has been changed slightly. In the past, if the *zip* setting was “on” then *graph* commands would be ignored; no graphs would be created. This allowed *G7* to work at top speed by avoiding needless processing of *graph* commands. Unfortunately, trouble sometimes arose if *zip* was on and a *graph* command was followed by a *gsave* command, where *gsave* attempts to store a graph to disk. If no graph were created, then *gsave* had little alternative but to fail. Otherwise legitimate scripts could be made to fail simply by setting turning on the *zip* setting. For this reason, and because modern computers scarcely are slowed by the “few” extraneous calculations demanded by the *graph* routine, *G7* now produces graphs regardless of the *zip* setting. However, *G7* halts to display each graph only if the *zip* setting is is “off.”

## line <number>

The *line* command is used to specify the color, style, and other settings for lines in *G7* graphs. The newly available syntax for this command allows the user to recover the current settings. The *number* specifies the setting for the desired line; up to seven lines may be drawn on a single graph, and so values from 1-7 are legal.

The capability of *G7* is enhanced greatly because of the ability of scripts to execute other scripts, and for one script to pass arguments to a second. Similar capability is employed elsewhere, such as with the *do* looping command. In the past, the number of arguments that could be passed at one time was limited to 9. Each argument could be employed within the script as %1 for the first argument, %2 for the second, and so on up to %9. While the restriction to only nine arguments seldom was found binding, sometimes it proved troublesome for more sophisticated routines. The number of arguments that can be sent to a *do* routine, and *add* file, or similar routines has been increased to 99. The extension to the syntax is obvious: %10, ..., %99.

Finally, several other routines were developed but are not sufficiently complete to report here. Perhaps the most important is a nonlinear regression routine to estimate the parameters of multinomial logit functions. These functions are very useful when calculating and predicting shares. Documentation will be provided when development is complete.

### 3.2 Interface Improvements

We next review a handful of modest improvements to the graphical user interface. These improvements speed and standardize users' control over the program. Indeed, many of the improvements introduce one or two-key shortcuts for existing routines.

Last year, a “Window” menu item was introduced to the main *G7* window. This menu listed all open windows and allowed the user to bring any one to the front. Two new items have been added to the menu in order to speed closure of multiple windows: “Close All Editors” and “Close All Other.”

The *step* command has been added to the *G7* scripting language. The command brings up a window that allows the user control over each following command. For each command, the user can execute the command or modify it, cancel all subsequent operations, or continue without pause. This can be useful when the behavior of *G7* is not clear. In particular, the user has the ability to modify the line of code that currently is in the input buffer. This allows the user to see the code as *G7* sees it, after any arguments (such as %1) have been replaced with values. The feature remains fairly crude, but additional capability will follow.

The *G7* editor now features the *F10* shortcut which will execute selected text. This especially is useful when adding code, line by line, to the end of a long script. Rather than repeatedly executing the entire file, the user now can select several lines and press *F10* in order to generate results.

Finally, a “zip” button has been added to the graph control bar. This allows the user to view several graphs and then speed through remaining graphs without pause. A “Find Again” shortcut (*F3*) has been added to the *look* window, along with the standard *Ctrl+F* shortcut to control the *Find* routine in the *look* window.

### 3.3 Documentation Descriptions

Two important pieces of documentation have been revised and extended. They are the *G7*

*Reference Manual*<sup>3</sup> and the *G7* help files. The help files include documentation for the new and revised routines reported here. See the “New for 2007” list under the “New Features” section on the “Contents” tab for links to recent major changes. The help files include descriptions and step-by-step instructions for sophisticated data construction techniques developed at Inforum.

### 3.4 Stability Matters

As reported at previous conferences, work has continued to detect and fix problems with *G7*. I am happy to report that while *G7* remains imperfect, the number of reported bugs reported in the past year has dwindled. *G7* grows increasingly stable and its ability to handle problems intelligently and reliably is much improved. As always, reports of any difficulties experienced by users are sought rather eagerly, and such reports are essential in order to continue our work to perfect the software.

Many other problems were fixed and minor improvements made. For example, some capability of the *show* command mysteriously had been lost, but full capability has been restored and the features claimed in the documentation again exist. While such improvements are not listed here, we trust that *G7* users will appreciate the improved stability, greater flexibility, and increased power.

### 4. Other Software

In work reported at recent conferences and offered on the Inforum web site, Dr. Almon has extended the capability of Inforum's interindustry modeling software *Interdyme* to allow optimal values to be found for certain parameters. Such capability first was developed for macroeconomic models.<sup>4</sup> Work to extend the capabilities to interindustry models has taken several years, and progress was reported earlier at several points.<sup>5</sup> The work is described in greater detail as part of Dr. Almon's three-volume series on economic modeling.<sup>6</sup>

The first application of this software on a fairly large model was made by Somprwin Manprasert in his interindustry model of Thailand.<sup>7</sup> An application on a still larger scale has begun as Doug

---

3 The reference manual is available on the Inforum web site at <http://www.inforum.umd.edu/papers/inforum/software/GBook.pdf>.

4 See *Experience with Optimization in Inforum Models*, available at <http://www.inforumweb.umd.edu/papers/wp/wp/2001/wp01005.pdf>

5 See the material presented at the 2006 conference at <http://www.inforumweb.umd.edu/papers/conferences/2006/Craft.zip>, the *Inforum Software for Building Dynamic, Interindustry Macroeconomic Models* paper presented at the 2005 conference and available at <http://www.inforumweb.umd.edu/papers/conferences/2005/AlmonIIOA.pdf>, in *Interdyme Pedagogy* presented at the 2004 conference and available at <http://www.inforumweb.umd.edu/papers/conferences/2004/almon.zip>, and finally in the 2003 paper *Progress Towards Optimization in Interdyme Models* that is available at [http://www.inforumweb.umd.edu/papers/conferences/2003/s\\_clop01.pdf](http://www.inforumweb.umd.edu/papers/conferences/2003/s_clop01.pdf).

6 Dr. Almon's work may be found on the Inforum web site: *The Craft of Economic Modeling: Part III* is available at <http://www.inforumweb.umd.edu/papers/publishedwork/books/craft3.pdf>.

7 *A Thai Interindustry Dynamic Model With Optimization* is available at

Meade leads the updating and revisions to Inforum's flagship U.S. model, *IdLift*. Such applications provide essential experience to test such software, both in its fundamental design and in the specification of each line of code.

Another piece of software, known as *IdBuild*<sup>8</sup>, translates equations written by *G7* into the C++ programming language. This C++ code can be compiled along with the *Interdyme* libraries and user-build code in order to produce a full executable model. In addition, *IdBuild* collects macroeconomic data needed by the model from other *G7* data banks. The software recently was improved, and the updated software is available on the web site. We list here a few of the changes. First, various improvements make the parsing capabilities more robust. In particular, longer strings can be read, and the reading of text is more reliable. Second, the square root, exponential, and log functions now are encoded in C++ to rely on Interdyme “wrapper” functions. These functions call corresponding C++ functions indirectly in order to improve error handling, leaving the user's model more robust. Third, path names in the *IdBuild* scripts now may be placed within quotation marks. This allows paths to begin with '.' so that “..\filename.dat” now is a legal argument. Finally, internal error handling is improved, so that problems are less likely to go unnoticed. Other changes, such as extended support for *G7* functions, will be documented later. Still other changes will not be documented and will be obvious only to astute users.

The software for building macroeconomic models is known as *Run*. Unfortunately, several reports have been received about difficulties in building and compiling macroeconomic models with Inforum software. The difficulties seem related to the compiler. More precisely, *Run* depends on the free version of the Borland C++ compiler. On machines that also feature Borland's Builder 6 software, problems sometimes arise. In our experience, the software runs without trouble on machines without Builder 6. The problems have not been resolved just yet, but we hope to release in the near future a more robust version of the software.

## 5. Conclusions

While much progress has been made in the past year, more work lies ahead. Additional tools for *G7* already are in development stages, and more projects will be launched in the coming year. As always, work to improve stability will continue. Please monitor the Inforum web site for future software updates.

The new web site offers many improvements and new features, but more work is needed. A variety of small problems need to be identified and fixed. Some possible extensions to the site were mentioned, and many others are possible. As they are identified and as time allows, key extensions and refinements will be pursued.

---

<http://www.inforumweb.umd.edu/papers/publishedwork/dissertations/manprasert.pdf>.

8 This version of *IdBuild* is available at <http://www.inforum.umd.edu/software/materials/optim/Idbuild.exe>. The version number is 6.103.

Both the software and the web site can be improved with the help of Inforum partners. Reports of problems, suggested improvements, and other questions or ideas can be sent through the webmaster's address or directly to the author. If any readers are interested in joining a mailing list to be informed of the latest work, please send your contact information to either of those addresses.