

EXPGUI*, a graphical user interface for *GSAS

Brian H. Toby

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

EXPGUI, a graphical user interface for GSAS

Brian H. Toby

NIST Center for Neutron Research, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8562, USA. Correspondence e-mail: brian.toby@nist.gov

A description and justification of the *EXPGUI* program is presented. This program implements a graphical user interface and shell for the *GSAS* single-crystal and Rietveld package. Use of the Tcl/Tk scripting language allows *EXPGUI* to be platform independent. Also included is a synopsis of how the program is implemented.

1. Introduction

Graphical user interfaces (GUIs) simplify use of computers by presenting information in a manner that allows rapid assimilation and manipulation. The use of visual constructs (widgets) that mimic physical objects such as 'switches' and 'buttons' can speed learning, by providing an intuitive method to provide input to the computer. A GUI is not always an improvement. As demonstrated by some widely used commercial programs, a poor GUI implementation can obscure functionality. If the GUI is organized in a counterintuitive manner, or if the menu contents are arranged haphazardly, or if commonly performed operations require several unexpected steps to be performed, then a user must typically invest a significant amount of time in learning how to use the program before the program can be used effectively. A good GUI design does not require users to memorize the steps needed to perform an action. This is particularly important for scientific applications, where the goal of the user should be to understand the theory behind a program rather than master the arcane steps needed to perform an action. There is one case where memorization may be unavoidable. It is appropriate that a GUI incorporate shortcuts that simplify multi-step tasks; use of these shortcuts may not always be intuitive and expert users may choose to commit them to memory to speed their work. There should also be an obvious way to perform the same tasks, albeit less efficiently, without use of the shortcut.

This article describes a GUI developed for the *Generalized Structure and Analysis Software (GSAS)* package (Larson & Von Dreele, 2000). The GUI was designed for both novice and expert users, with several goals: to aid scientists who visit the NIST Center for Neutron Research and need to learn Rietveld analysis (Rietveld, 1969), to extend the capabilities of the *GSAS* software, and to facilitate research projects, by allowing expert users to work more quickly.

1.1. GSAS

The *GSAS* package has been developed over a period of decades by Allen C. Larson and Robert B. Von Dreele. It is widely used and is arguably one of the most comprehensive packages of crystallographic software ever developed, as it can be used to fit crystallographic and magnetic structural models to X-ray and neutron single-crystal and powder diffraction data. It can be used with both time-of-flight (TOF) neutron and constant-wavelength (CW) neutron powder diffraction data, as well as laboratory and synchrotron CW X-ray powder diffraction data and even energy dispersive X-ray powder diffraction measurements. Up to 99 sets of data may be used in combination to determine a single crystallographic model. In powder diffraction, it is

common for a sample to contain more than one crystallographic phase. *GSAS* allows a crystallographic model to be composed of as many as nine crystallographic phases. In addition to structure determination, *GSAS* is also used for lattice-constant determination (even for materials with unknown structures), simulation of powder diffraction data, and for texture analysis. Recently, it was enhanced to enable the fitting of protein structures to powder diffraction data.

1.2. The native GSAS user interface

The *GSAS* package contains approximately fifty programs. The majority of the programs require minimal or no user input. The only program in the *GSAS* package that requires extensive user interaction is named *EXPEDT*, an acronym for experiment editor. The *EXPEDT* program is operated through a series of dialogs, where the user is prompted to answer a question with a series of one-letter options. Typing a question mark (?) causes *EXPEDT* to print a short explanation for each option. The information provided by the user is saved in a file called the experiment (.EXP) file. Thus, the experiment file contains a wide range of information: for example, atom coordinates and diffraction data file names, as well as least-squares and Fourier map options. The *GSAS* package is quite sophisticated, in that a large number of options are available, but this means that the user may have to transcend six or more levels of dialogs in *EXPEDT* to provide input for a particular option and this can be quite confusing to a novice user. Likewise, it can be very hard to review all the options and values stored in the experiment file, as many dialogs must be visited.

The *EXPGUI* program offers an alternate, GUI-based mechanism for reviewing and editing the *GSAS* experiment file. The *EXPGUI* program implements only a small fraction of the full complement of features available in *EXPEDT*, but most of the commonly used features for powder diffraction are present in *EXPGUI*. Since *EXPGUI* and *EXPEDT* can be used interchangeably, use of *EXPGUI* does not prevent access to any capabilities within *GSAS*. The *EXPGUI* program can be used in two ways. It can be run from one of the several platform-specific 'shell programs' that allow access to the various *GSAS* programs. More significantly, *EXPGUI* can function both as an experiment file editor and as a *GSAS* shell program, providing GUI access to initiate other *GSAS* programs.

1.3. Platform independence via Tcl/Tk

The *EXPGUI* program has been written as a series of Tcl/Tk scripts (Ousterhout, 1994). [See Urzhumtseva & Urzhumtsev (1997, 1999, 2000) for examples of other crystallographic codes developed

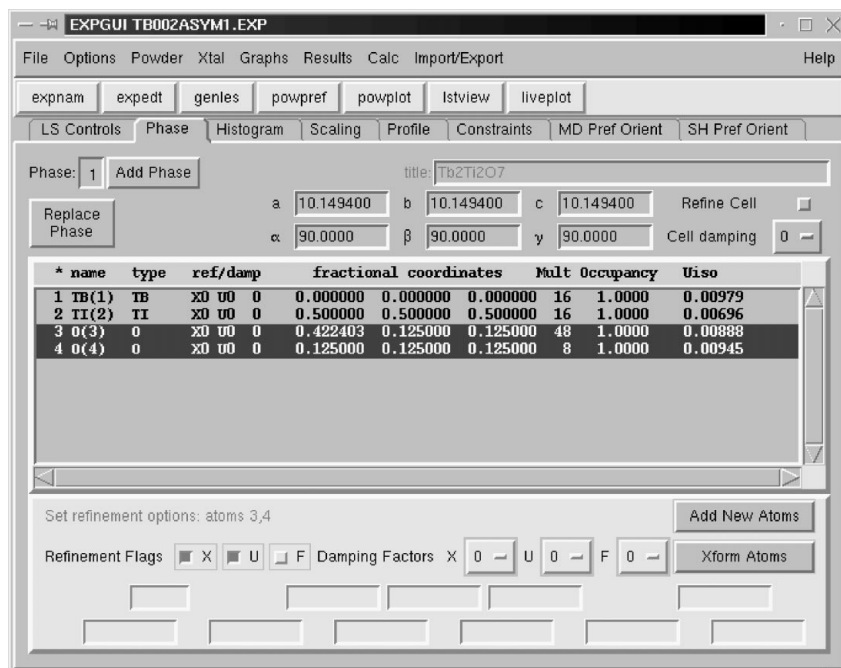


Figure 1
A screenshot of the main *EXPGUI* window.

using Tcl/Tk.] These Tcl/Tk scripts are not compiled, rather they are interpreted as the program is run. Since a Tcl/Tk interpreter is available for all common operating systems, the exact same version of *EXPGUI* can be run on all platforms where *GSAS* is available. The program has been tested under Windows 95 and Windows NT, as well as several variants of Unix. Most development is performed under Linux. Minor adaptations will be needed for the Macintosh; this work is in progress. A VMS implementation is possible, if interest warrants.

Interpreted languages, such as Tcl/Tk and Perl, run slower than compiled ones, such as Fortran and C. However, with modern computers, speed is not usually a critical factor for interactive programs, since human response times are usually much slower than programs. Performance of *EXPGUI* on Unix platforms is excellent. Windows performance is significantly slower than Unix, but is usually sufficient that response is nearly instantaneous.

Interpreted languages offer three advantages. One is that code can be written and tested in small sections, without compilation. This speeds program development. Second, since the scripts are distributed as source code, users can add new features. Indeed, several sections of *EXPGUI* were adapted from code submitted to the author by users. Third, the program can be customized by including code in user-supplied files that are read when the program is started.

2. The *EXPGUI* interface

The *EXPGUI* main window is divided into three sections, as seen in Fig. 1. The menu bar at the top is used to run all of the various *GSAS* programs. It is also used to select options within *EXPGUI* and can be used to run several utility functions built into *EXPGUI*.

Below the menu bar is a button bar. The button bar provides a quick way to invoke frequently used menu bar commands. The program documentation includes instructions and examples on how to customize the button bar, so that users can tailor the bar to contain their own most frequently used options.

The bottom section of the window is used to display and modify the contents of the experiment file. Since the experiment file can contain many different parameters, this space is shared through use of a notebook widget, and thus there are 'panes' for (i) least-squares options, (ii) phase and atomic parameters, (iii) diffractogram and background parameters, (iv) scale factors and phase fractions, (v) profile terms, (vi) constraints on profile terms and atomic parameters, (vii) March–Dollase model preferential orientation terms, and (viii) spherical harmonic expansion preferential orientation terms.

The structure of the experiment file to a large extent determines the layout of each pane. For example, for each phase in the experiment file there is a set of atomic parameters and (when used) a set of spherical harmonic terms. Thus the panes implementing access to atomic parameters and spherical harmonic terms include a set of buttons for selecting the phase. Since there is a set of background terms and diffractometer constants for each 'histogram' (as *GSAS* names diffractograms), the panes for those parameters have a list of histograms to be selected. For some other parameters, such as profile terms, phase fractions and March–Dollase preferred orientation terms, a set

exists for every diffractogram, as well as for every phase. For those parameters, a list of histograms is displayed. When one is selected, the parameters for all the appropriate phases are displayed.

When *GSAS* is used with several diffractograms, it is often convenient to perform a single action that affects parameters for many histograms, as opposed to repeating the action for each histogram. An example of this is that one may wish to change a set of diffractometer constants or phase fraction refinement flags for all histograms. A special mode in *EXPGUI*, the multiple-histogram selection mode, allows the parameters for histograms to be grouped together for quick access. This can be used by an experienced user efficiently to set refinement flags or to change initial parameter values in complex refinements. *EXPGUI* has two different types of multiple-histogram selection modes. One mode allows any combination of histograms to be selected together. The second set of modes restricts selection to a particular data type, for example, only CW X-ray data or only TOF neutron data. When selection of all histograms is allowed, certain panes are automatically disabled, as it would not make sense, for example, to group together different types of profile functions or diffractometer constants.

EXPGUI also offers additional modes that affect how data and results are presented. For example, histograms can be displayed in the order they are input to *GSAS*, or can be sorted by angle (wavelength), bank number or type. Atoms can also be displayed in the order they are input, or be sorted by atom type, x, y, z, site multiplicity, or fractional occupancy. Bond distance and angle output from the *DISAGL* program can be displayed in a separate scrollable window, or can be included in the *GSAS* listing (.LST) file, as is normally performed in *GSAS*.

3. *EXPGUI* utilities

The program adds additional features to *GSAS*. Most significant are two independent Tcl scripts, *LIVEPLOT* and *LSTVIEW*. The

LIVEPLOT program displays the Rietveld results by plotting an observed and computed diffractogram along with the background curve and the observed-minus-computed differences. As the Rietveld refinement progresses, the plot is updated. The mouse can be used to magnify ('zoom') a section of the plot by clicking on two corners of the desired region.

The *LSTVIEW* program is used to view the *GSAS* listing file in a scrolled window. Most *GSAS* programs append output to the end of this file. The *LSTVIEW* program updates the window to include this output as it is written. By default, *LSTVIEW* scrolls the window so that the most recent refinement cycle is displayed. Also, agreement factors are highlighted to make it easier to follow the refinement progress.

The original implementation of *GSAS* in VMS allowed users to keep an archive of previous versions of experiment files, so that it was possible to return to a previous stage of a refinement. This was lost when *GSAS* was ported to Unix and DOS, but has been reimplemented in *EXPGUI*. In Windows, archiving is performed by adding the current experiment file to a compressed archive named *expnam.ZIP*, where *expnam* is the root name of the *GSAS* experiment file. Compression requires that the shareware *PKZIP* program be installed in the *EXPGUI* directory. If this program is not available, the experiment file is archived by creating a copy named *expnam.EXP.xxx*, where *xxx* is a number between 000 and 999. In Unix, the experiment file is copied to a file named *expnam.EXP.xxx* or if the GNU *gzip* program is available, a compressed file named *expnam.EXP.xxx.gz*.

Multiple formats are supported for importing coordinates into *GSAS*, including the Crystallographic Information File (CIF) (Hall *et al.*, 1991). Capitalizing on the interpreted nature of Tcl, as the *EXPGUI* program is started, it incorporates any files named *import_XXXX.tcl* (where *XXXX* is an arbitrary name). In this way, a user can implement code to read a new coordinate format by simply creating a new short script.

Other utility scripts are incorporated into *EXPGUI* that (a) convert files to or from the appropriate direct-access format needed by *GSAS* on the particular platform where the program is used, (b) compute the composition of a unit cell, and (c) plot the peak widths as a function of 2θ .

4. Distribution

EXPGUI is available *via* the Internet from a NIST Web server (<ftp://ftp.ncnr.nist.gov/pub/cryst/gsas/>) as well as the Collaborative Computational Project, Number 14 (CCP14) regional mirror sites (Cockcroft, 2000), presently in the UK (<ftp://ftp.ccp14.dl.ac.uk/ccp14/ftp-mirror/briantoby/pub/cryst/gsas/>), Canada (<ftp://ccp14.sims.nrc.ca/ccp/ccp14/ftp-mirror/briantoby/pub/cryst/gsas/>) and Australia (<ftp://ftp.minerals.csiro.au/pub/xtallography/ccp14/ccp14/ftp-mirror/briantoby/pub/cryst/gsas/>). The code can be downloaded either as a compressed tar archive (*expgui.tar.gz*) or for Windows as a self-expanding zip archive (*expgui.exe*). The files contained in these two distribution archives are identical. *GSAS* and *EXPGUI* are also available on CD-ROM *via* the Crystallographic Nexus project: The International Union of Crystallography, the International Council for Science and the CCP14 collaborate to produce a CD-ROM that is available at no cost to developing world laboratories that lack access to the Internet (Cranswick, 2000).

The code is distributed without copyright protection. Thus, it can be modified and redistributed in any fashion. Users are asked to send

an e-mail address to the author, so that the program use can be documented and notice of program updates can be distributed.

New features are added to the program at irregular but often frequent intervals. As updates are made, revised versions of the program are placed on the Web immediately, as a minimally tested 'alpha' version. The updates are available in archive files named *expgui_ALPHA.tar.gz* or *expgui_alpha.zip*. A self-extracting file, *expgui_alpha.exe*, is updated less frequently. After the alpha version has been tested for some period of time, it is made the standard version.

5. Documentation

Documentation for *EXPGUI* is provided through more than a dozen Web pages. A Web page is provided to explain the use of each notebook pane. Other Web pages describe menu options, customization options, warning and error messages, and instructions for installation for Unix and for Windows. Context-sensitive help is available in most sections of the program. For example, the F1 key or the 'Help on current pane' item in the Help menu, will open a Web browser directed to the appropriate page. Most dialog windows, warning and error messages have a yellow 'Help' button. Pressing this button, or the F1 key, opens the documentation providing a more detailed explanation of the error message. Abbreviated help is also available from the 'Help summary' item in the Help menu.

The Web pages are included in the *EXPGUI* distribution and are available from a NIST Web server (<http://www.ncnr.nist.gov/xtal/software/expgui/>), as well as the CCP14 regional mirror sites presently in the UK (<http://www.ccp14.ac.uk/ccp/web-mirrors/briantoby/programs/crystallography/software/expgui/>), Canada (<http://ccp14.sims.nrc.ca/ccp/web-mirrors/briantoby/programs/crystallography/software/expgui/>) and Australia (<ftp://ftp.minerals.csiro.au/pub/xtallography/ccp14/ccp/web-mirrors/briantoby/programs/crystallography/software/expgui/>). Web pages for the alpha version are posted in a different directory (<http://www.ncnr.nist.gov/xtal/software/expgui/alpha/>) and are also mirrored by CCP14.

6. Program internals

It is convenient to keep the Tcl/Tk scripts in a number of smaller files, so that code can be tested more simply and so that some sections of the script are loaded only as they are needed, which speeds initialization of *EXPGUI*. A few of these files are worthy of special note, since they are of interest to users wishing to customize or extend the program. The file *gsasmenu.tcl* determines the menu and button bar contents, as well as the functions to be invoked and much of the help summary information. While this file can be modified by users, the preferred way to customize this file is to include information in a file named *localconfig* or *~/gsas_config* (*c:\.gsas_config* in Windows), as this will override the *gsasmenu.tcl* contents and will not be overwritten when *EXPGUI* is upgraded.

When the *EXPGUI* program is running, the contents of the experiment file being used is read into a Tcl array named *exparray*. The experiment file contents are divided into a large number of indexed records, where each record contains a specific type of information. Each record is stored in *EXPGUI* as a separate array element so that it can be accessed *via* the record's index. A separate array, *expmap*, provides lists of which phases, histograms and atoms are in use. After changes have been made to the copy of the experiment file in local memory, the contents are written back to the experiment file when another program is run or *EXPGUI* is ended.

Different routines have been written to extract and change information in the experiment file. As an example, the Tcl command `phaseinfo 3 gamma` is defined to return the value for the unit-cell angle γ and the Tcl command `phaseinfo 3 gamma set 120` is defined to set this angle to 120° . To simplify debugging of *EXPGUI*, all access to experiment file information is made through routines in the file `readexp.tcl`. Since these routines simplify reading and changing a *GSAS* experiment file, they may also be of value to anyone wishing to write Tcl programs that interact with *GSAS* experiment files, for example to process a large number of related refinements on similar data files.

The *EXPGUI* program implements a simple mechanism so that changes made on the various input panes are checked for errors, are correctly formatted and are then inserted into the appropriate element in the `exparray` array. Each text entry box, `checkboxbutton`, *etc.*, is mapped to an element in the Tcl array `entryvar`. For example, `entryvar(gamma)` is used for the unit-cell parameter γ . For every such array element, a corresponding element in array `entrycmd` provides the Tcl command needed to save this value in `exparray`. Thus, every time the variable `entryvar(gamma)` is changed, the command stored in `entryvar(gamma)` will be invoked. Values are formatted where needed and this provides a mechanism for error checking. Should an invalid value be specified for a numeric value, for example a string containing letters and numbers, the value is not inserted in array `exparray`. Also, the color of the displayed entry is changed to red, to indicate an error is present. Since this array update procedure is performed with every keystroke, it is invoked far more often than is needed, but this operation is fairly rapid compared to human typing speeds, so the impact on perceived program speed is minor.

In some program sections, changes are not made on an immediate basis. For parameters such as background terms and atomic and profile constraints, editing is performed by creating a new dialog window where values are entered. At the bottom of these windows there are buttons labeled with some variant of 'Save' and 'Cancel'. Changes are made after 'Save' is pressed; no changes are made if 'Cancel' is pressed. Error checking of input is made only after the 'Save' button is pressed.

7. Conclusions

An expandable graphical user interface for the *GSAS* program suite has been developed. While the interface does not directly support all of the versatility of the entire *GSAS* suite, it does not preclude use of

any *GSAS* features. The majority of features commonly used with Rietveld refinements are included; for these features the GUI both aids novice users learning the package and speeds use of *GSAS* by more expert users.

The Tcl/Tk scripting language provides an excellent mechanism for implementing such an interface, since it offers platform independence, facilitates customization and has a programming environment suitable for rapid program development. Despite the relatively large size of the project (>12000 lines of code), the program starts and responds reasonably quickly.

I am indebted to Jonathan Wasserman, who coded sections of the initial GUI, and to Dr Pamela Whitfield, who contributed a draft of the March–Dollase and profile constraints sections and reviewed this manuscript. I would also like to thank Lachlan M. D. Cranswick of CCP14, who has made many useful suggestions for improving *EXPGUI* and has created several Web-based tutorials on the use of *EXPGUI*, as well as the Engineering and Physical Sciences Research Council of the UK, which funds the CCP14. I am grateful to Dr Diane P. Toby, Dr Whitfield and Mr Cranswick for reading and commenting on early versions of this manuscript. Many people have contributed indirectly to this effort, through their contributions to Tcl/Tk, but in particular I would like to thank Dr Przemek Klosowski for originally introducing me to Tcl/Tk, Mumit Khan for his work on Windows implementations of `gcc`, `g77` and Tcl/Tk, Michael I. Schwartz for his *WINEXEC* package, and Dr John K. Ousterhout for creating Tcl/Tk and publicly distributing the source code. Finally, this work could not have been possible without the efforts of Drs Allen C. Larson and Robert B. Von Dreele to create and maintain *GSAS*.

References

- Cockcroft, J. K. (2000). Chairman, Collaborative Computational Project, Number 14 (CCP14), <http://www.ccp14.ac.uk/>.
- Cranswick, L. M. D. (2000). Crystallographic Nexus Software Library on CD-ROM, Daresbury Laboratory, Warrington, WA4 4AD, UK. E-mail: lcranswick@dl.ac.uk.
- Hall, S. R., Allen, F. H. & Brown, I. D. (1991). *Acta Cryst.* **A47**, 655–685.
- Larson, A. C. & Von Dreele, R. B. (2000). *General Structure Analysis System (GSAS)*, Los Alamos National Laboratory, Report LAUR 86–748.
- Ousterhout, J. K. (1994). *Tcl and the Tk Toolkit*. London: Addison-Wesley.
- Rietveld, H. M. (1969). *J. Appl. Cryst.* **2**, 65–71.
- Urzhumtseva, L. M. & Urzhumtsev, A. G. (1997). *J. Appl. Cryst.* **30**, 402–410.
- Urzhumtseva, L. M. & Urzhumtsev, A. G. (1999). *J. Appl. Cryst.* **32**, 376–377.
- Urzhumtseva, L. M. & Urzhumtsev, A. G. (2000). *J. Appl. Cryst.* **33**, 992.