Journal of
**Applied
Crystallography**
ISSN 0021-8898

CIF applications. XIV. Reporting of Rietveld results using pdCIF: *GSAS2CIF*

Brian H. Toby, Robert B. Von Dreele and Allen C. Larson

Journal of
Applied
Crystallography

ISSN 0021-8898

Received 7 April 2003
Accepted 29 July 2003

CIF applications. XIV. Reporting of Rietveld results using pdCIF: *GSAS2CIF*

Brian H. Toby,[a]* Robert B. Von Dreele[b] and Allen C. Larson[c]

[a]NIST Center for Neutron Research, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8562, USA, [b]IPNS/APS Divisions, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439-4814, USA, and [c]14 Cerrado Loop, Santa Fe, New Mexico 87508-8248, USA. Correspondence e-mail: brian.toby@nist.gov

A discussion of the process of creating powder diffraction CIF documents (pdCIF) from Rietveld results is presented, with particular focus on the computer program *GSAS2CIF*. The data structures used within *GSAS2CIF* are described, as well as how the program implements template files for descriptive information. Two graphical user interface utilities are also discussed.

1. Introduction

The crystallographic information file (CIF) was created as a descriptive language for the exchange and archive of single-crystal structure determination data and results (Hall *et al.*, 1991). Later developments expanded the scope of CIF for macromolecular crystallography, mmCIF (Bourne *et al.*, 1997), as well as powder diffraction, pdCIF (Toby, 2003). In comparison with single-crystal diffraction, powder diffraction places additional demands on CIF: the types of instrumentation commonly used for powder diffraction data are quite diverse, powder diffraction studies frequently utilize more than one data set, and materials studied by powder diffraction may contain more than one chemical species.

This article describes *GSAS2CIF*, a recently completed computer program that uses CIF to document Rietveld refinement results (Rietveld, 1969). This article does not intend to explain how the software is used, as that topic has been covered in a web document (http://www.ncnr.nist.gov/xtal/software/expgui/gsas2cif.html). This web page is distributed with the *EXPGUI* package (Toby, 2001) or can be found on the CCP14 mirror sites (http://www.ccp14.ac.uk). The aim of this article is to document the methodology behind the program, with the goal of aiding development of other software for generating pdCIF output. The Fortran code written to implement *GSAS2CIF* has been placed in a separate series of web documents, along with extensive annotations (http://www.ncnr.nist.gov/xtal/software/cif/gsas2cif/gsas2cif_doc.html).

The *GSAS2CIF* program is part of the *GSAS* crystallographic software package, a widely used set of programs (Larson & Von Dreele, 1986). *GSAS* is arguably the most versatile crystallographic structure refinement package in existence, as it can perform Rietveld refinements using constant-wavelength (CW) and time-of-flight (TOF) neutron diffraction data, as well as data from X-ray instruments ranging from common laboratory to very high resolution synchrotron instruments, including energy-dispersive X-ray (ED) instruments. *GSAS* can also fit models to neutron and X-ray single-crystal data. *GSAS* simultaneously fits models to any combination of these data types, utilizing the advantages of each technique. In fact, *GSAS* can use up to 99 data sets (histograms, in *GSAS* nomenclature) to model as many as nine chemical species simultaneously. The *GSAS2CIF* program is intended to report any study that can be performed within *GSAS*, so it must be able to create CIFs with any type of data, any number of chemical species and any number of data sets.

Another complexity of CIF generation must be addressed by the *GSAS2CIF* program: a well documented CIF requires a wealth of descriptive information, in addition to the raw data and refinement parameters that are found in existing *GSAS* data files. This descriptive information must be manually entered by the investigator. *GSAS2CIF* uses a series of template files for this descriptive information; these files are automatically integrated with the data and results from *GSAS* when the CIF is generated. By editing the template files rather than the final CIF, user input is retained when the CIF is updated. Further, these template files can be customized and reused. Template files will be discussed further in §4 below.

The *GSAS2CIF* program requires very little, if any, user input to run. If input is needed, the user is prompted to type information using the text-based interface employed throughout *GSAS*. For input of descriptive information into template files, a text editor can be used. Alternatively, a graphical user interface (GUI) tool, *FillCIF*, has been created for this purpose within the GUI interface to *GSAS*, *EXPGUI* (Toby, 2001). A second GUI tool, *CIFselect*, is also provided to set the 'publish' flag for interatomic distances and angles. These GUI tools are described briefly in §6.

2. Rietveld data structures

The data structures of pdCIF are rather complex and require attention when pdCIF software is written. To aid others in the design of software that creates or reads pdCIFs, these data structures will be defined in detail in §3. To understand why such complexity arises, it is worthwhile to consider the types of data and parameters generated from powder diffraction in relation to the underlying structure of CIF. A CIF consists of carefully defined data names that are associated with either a single value or, if contained in a loop, a series of values. CIF dictionaries, *i.e.* the core CIF dictionary and the pdCIF dictionary, define the data names as well as the range of allowed values (http://www.iucr.org/iucr-top/cif). CIF loops often contain more than one data name, so a CIF loop structure is best envisioned as a two-dimensional spreadsheet or table.

One restriction on CIF data structures arises because the CIF dictionaries define a category for each CIF data name and the data definition language (DDL) requires that all data names in a loop must belong to the same category. Further, if data items in a category will be split into more than one loop, each loop must contain a 'reference item'. This increases the complexity of pdCIF software and, as will be seen in §3, the data items used in loops will vary. In a few cases, the assignment of CIF categories is not yet completely consistent with the intended use of CIF data names; such conflicts are also noted.

2.1. Block id labels

The CIF data items (data names and values) are logically grouped together in data blocks, where a data name may be specified in a block only once. This means that a single CIF data block can describe only one data set and only one chemical species. Initially, CIF was envisioned for a paradigm where an individual crystallographic determination would be reported in a single CIF data block. For a series of related structures, several blocks might be reported together in a single file, but the standard did not initially implement any data structures to relate one block to another. Thus, each block was logically independent. For powder diffraction, where multiple data sets and/or multiple chemical species are present, a CIF will contain several data blocks that are logically interrelated. The pdCIF dictionary implements a data item, `_pd_block_id`, to provide a unique 'tag' for each data block. Further, pdCIF provides data structures to document the relationships between blocks, by defining 'pointers' to these block id tags. These structures are described in §3.2, §3.3.1 and §3.4.1, and are illustrated by an example provided in Appendix *A*.

2.2. Multiplicity of Data Items

Data relationships in pdCIF can be conceptualized by the number of values or sets of values needed to describe a Rietveld refinement. Consider the case where there are D data sets and P chemical species. There will be P sets of cell and atomic parameters, D sets of background parameters and individual data set R factors, and $D \times P$ sets of profile parameters and species scale factors (phase fractions). There are also $D \times P$ sets of reflections. Other information will appear only once in a CIF, for example overall R factors and publication information. The number of items can vary, depending on circumstance. For many experiments, a single table of scattering factors will suffice. However, in an anomalous dispersion experiment, f' and f'' values differ for each wavelength, so there can be D sets of these values. Other parameters will vary depending on implementation. For example, preferred orientation corrections might result in P or $D \times P$ sets of parameters, depending on how the correction is parameterized in software.

3. Data structure organization within *GSAS2CIF*

The above discussion of data structures suggests an organization for pdCIFs. As mentioned before, each data block can contain a single data set and a single species. There will be a many-to-one relationship between chemical species and data sets, when both $D > 1$ and $P > 1$, as there must be $D + P$ blocks. Even in the case where $D > 1$ and $P = 1$, it would not make sense to include the chemical species information in just one of the D data blocks, since the structural parameters were derived from all of the D data blocks. The same argument applies when multiple chemical species contribute to a single data set. The CIF will also contain overall information, such as global parameters and overall R factors. Since this information is connected to all species and data sets, it is best placed in a separate block. This information could be combined with species information in the case where $P = 1$ or with data-set information where $D = 1$, but implementing this would be complex; the *GSAS2CIF* program always keeps the overall information in a separate block when either $D > 1$ or $P > 1$. Further, *GSAS2CIF* separates publication information into a separate block from overall information for the refinement, as this simplifies collating several Rietveld analyses into a single large electronic submission. Thus, when either $D > 1$ or $P > 1$, *GSAS2CIF* creates a total of $D + P + 2$ blocks.

In the case where $D = P = 1$, *e.g.* where one diffraction data set is used to determine the structure of a single species, one could retain the previous 'multiblock' structure and use four data blocks. However, this complexity is not needed. Further, while all CIF-compliant software should be able to read files that contain multiple blocks, in practice some programs cannot. When possible (*e.g.* when $D = P = 1$), *GSAS2CIF* places only one block in the CIF.

The following paragraphs outline the information to be found in each block of a multiblock pdCIF, with particular focus on the data loops in each block. Single-block pdCIFs will contain much of the same information.

3.1. Publication block

The first block in the CIF contains information used in a publication, primarily `_audit_*`, `_publ_*` and `_journal_*` data items. This information is copied directly from the publication template file and thus the user will need to specify most of this information, if it is to be included. Other data items that specify descriptive information about the sample, as well as some descriptive information about the refinement, also appear in this block.

3.2. Overall information block

The overall information block specifies refinement statistics, primarily using `_refine_*` and `_pd_proc_ls_*` data items. This block does not use any information from a template file. To provide a logical connection between the overall information and the data sets and chemical species, the overall information block contains two loops, one for `_pd_block_diffractogram_id` with D values and one for `_pd_phase_block_id` with P values. These loops are comparable with those found in the chemical species blocks and the diffraction data blocks, discussed in §3.3.1 and §3.4.1. These two loops are omitted in a single-block pdCIF.

3.3. Chemical species blocks

As described previously, a block is created for each of the P chemical species contained in the experiment. Each of these blocks defines the crystallographic parameters for the chemical species, containing items that define the unit-cell dimensions, atomic parameters, *etc*. This block also contains descriptive information from a species template file, as will be discussed further in §4. Most of the loops found in a chemical species block would be found in any CIF, such as symmetry (`_symmetry_equiv_pos_as_xyz` and `_symmetry_equiv_pos_site_id`), atomic parameters (`_atom_site_fract_x`, *etc.*), unit-cell contents (`_atom_type_symbol` and `_atom_type_number_in_cell`) and, when appropriate, anisotropic displacement parameters (`_atom_site_aniso_U_11`, *etc.*). This block will also contain loops for interatomic distances (`_geom_bond_distance`, *etc.*) and for interatomic angles (`_geom_angle`, *etc.*).

3.3.1. Data-set pointers loop. The only loop found in a chemical species block that is unique to pdCIF is one that contains values for `_pd_block_diffractogram_id`. This is used in multiblock CIFs to connect logically the chemical species block to the data sets that were used to obtain the crystallographic parameters. Thus, there will be up to D block id values in this loop. It is possible to have fewer than D values in this loop, as a chemical species may be present in some but not all the data sets used in a refinement. This loop is omitted in a single-block pdCIF.

3.4. Diffraction data blocks

In a multiblock CIF, there will be D blocks containing powder diffraction data, as well as information describing the diffraction instrument and how the data were recorded. This descriptive information is copied from an instrument template file, as will be discussed further in §4.

The contents of this block vary, depending on the type of data used for the experiment and how those data are processed. While powder diffraction data are intensity values as a function of an independent variable, the actual independent variable varies with the type of instrument in use. For example, 2θ is the independent variable for CW instruments. Both TOF instruments and ED instruments measure intensity *versus* a channel number, where this channel number corresponds to time or energy, respectively. Both the intensity values and perhaps the independent variable may require scaling or calibration before the data are used in the Rietveld refinement. For this reason, pdCIF offers both `_pd_meas_*` and `_pd_proc_*` data names, for as-collected and processed data, respectively.

In addition to the observed intensity values, the block reports the intensity values computed from the structural model, as well as computed background values. Standard errors for the observed intensities and least-squares weights may also be included.

Where possible, *GSAS2CIF* places all of the observed and computed diffraction data and related items in a single data loop. However, *GSAS* allows for both averaging and sampling of data; these options break the one-to-one correspondence between the input observations and the intensities used for refinement. This requires that the as-input and processed data items be reported in separate loops (see §3.4.3 and §3.4.4).

There are D sets of parameters and results related to the D diffraction data sets. Some examples of these include background parameters and profile R factors for individual data sets. Thus, these values are placed in the diffraction data blocks. Parameters that differ by both data set and chemical species, *i.e.* where there are $D \times P$ sets of values, are also placed in the diffraction data blocks. Profile parameters and phase fractions are examples of these types of parameters.

3.4.1. Phase table. As was mentioned in §2.1, pointers between diffraction data and chemical species blocks are used in multiblock CIFs. In the diffraction data block, a loop is included that provides a reference to the blocks that define each of the P species found in the current data set. Since this loop enumerates the chemical species referenced in every diffraction data block, the loop is also used for parameters where there are $D + P$ values, *i.e.* where values differ both by species and by data set.

In *GSAS2CIF*, the phase table contains data names `_pd_phase_id`, `_pd_phase_block_id`, `_pd_phase_mass_%`, `_pd_proc_ls_profile_function`, `_pd_proc_ls_peak_cutoff`, and, when appropriate, `_pd_proc_ls_pref_orient_corr`. The `_pd_phase_id` data item is simply the *GSAS* phase number. This value links the chemical species to each reflection in the reflection loop (§3.4.5). The `_pd_phase_block_id` contains the block id for the phase block where the phase information is found, and thus provides a pointer to the crystallographic information.

It should be noted that `_pd_proc_ls_profile_function` and `_pd_proc_ls_peak_cutoff` are not in the same category as the other data items in the loop. This problem will need to be addressed in a new version of the pdCIF dictionary. This loop is omitted in a single-block pdCIF.

3.4.2. Multiple-wavelength loop. Multiple wavelengths are a necessary evil in powder diffraction. The Cu $K\alpha$ doublet ($K\alpha_1$ and $K\alpha_2$) is commonly used in laboratory instruments. Multiple wavelengths may also be seen in other circumstances. For example, higher-order wavelengths ($\lambda/2$, *etc.*) may pass through monochromators, or impurities in the X-ray tube anode may result in additional wavelengths. Where more than one wavelength is present in the data set, this information is defined in a loop with the following entries: `_diffrn_radiation_wavelength`, `_diffrn_radiation_wavelength_wt`, `_diffrn_radiation_type` and `_diffrn_radiation_wavelength_id`. It should be noted that `_diffrn_radiation_type` is not in the same category as the other CIF data names, despite the original intent that these data items have their current usage. This will need to be resolved in a future version of the core dictionary. The wavelength loop is omitted when only one wavelength is present or an energy-dispersive technique is used.

3.4.3. Raw data loop. As mentioned previously, when the numbers of raw and processed intensity values differ, there must be a separate loop for raw data. When this occurs, *GSAS2CIF* calls subroutine *WRITERAWDATA* to create this loop, which will contain either `_pd_meas_intensity_total` or `_pd_meas_counts_total`. The loop will include the data item `_pd_meas_point_id` to satisfy the DDL 1.4 category requirement. If the data are recorded in units of 2θ with fixed steps, then the minimum, maximum and step will be recorded as single values with `_pd_meas_2theta_range_*` and the loop will have only two data names. Otherwise, the loop will also contain a third data name, `_pd_meas_time_of_flight`, `_pd_meas_detector_id` or `_pd_meas_2theta_scan`, for TOF, ED or CW data, respectively.

3.4.4. Observed and computed data loop. The entries in the observed and computed data loop will vary tremendously, depending on the type of data used in the refinement. If there is a separate raw data loop present for the data-set block, then the loop will contain `_pd_proc_*` values. Otherwise, the observed data will be reported in this loop as `_pd_meas_*` values. The independent variable can be one of the following data names: `_pd_meas_time_of_flight` (TOF), `_pd_proc_energy_detection` (ED), or `_pd_meas_2theta_scan` or `_pd_proc_2theta_corrected` (CW). Alternatively, for CW data with a constant 2θ step size, the independent variable may be omitted from the loop and may be implied implicitly through use of the `_pd_meas_2theta_range_*` or `_pd_proc_2theta_range_*` data items, as was the case for the raw data loop. For TOF and ED data, `_pd_proc_d_spacing` will also be included. For ED data, `_pd_meas_point_id` is included as well, unless a separate raw data loop has been used.

When the raw data appear in a separate loop, the intensity values in this loop are reported as `_pd_proc_intensity_total` and the `_pd_meas_*` are reported in the raw data loop. Otherwise, the input intensity points will be defined using either `_pd_meas_intensity_total` or `_pd_meas_counts_total`. The latter data item is used when the uncertainties are the square-root of the intensities. Note that *GSAS* sometimes applies corrections to input data, for example to correct for the incident spectrum correction. In this case, data item `_pd_proc_intensity_total` is used to report these scaled values.

The pdCIF definitions do not require this, but *GSAS2CIF* will always include data names `_pd_proc_ls_weight`, `_pd_proc_intensity_bkg_calc` and `_pd_calc_intensity_total` in this loop as well. Finally, if the raw data have been placed in a separate loop, the data item `_pd_proc_point_id` will also be included in the observed and computed data loop to satisfy the DDL 1.4 category requirement.

In many cases, there will be observed data points that are not used in the refinement. This may be because no reflections are observed in the region, or because the signal-to-noise level or the resolution are too poor for the data to be useful. When such points are present, the observed intensity value is reported, but the value '.' (a CIF placeholder meaning 'not determined') is reported for the `_pd_proc_*` and `_pd_calc_*` data items.

3.4.5. Reflection loop. Also in the diffraction data block is a loop containing a table of reflections. This loop contains the reflection indices, `_refln_index_h`, `_refln_index_k` and `_refln_index_l`, as well as the observed and computed structure-factor information, `_refln_F_squared_meas`, `_refln_F_squared_calc` and `_refln_phase_calc`. In a single-crystal diffraction data block, the `_refln_F_squared_meas` values are the 'observations', but in the powder diffraction case, these values are extracted from the diffraction pattern using the Rietveld method (Rietveld, 1969). This method relies on the ratio of computed structure factors to estimate intensities for overlapped reflections. The loop will also always include `_refln_observed_status`, although this will always be 'o' (meaning 'observed') for powder diffraction. Powder diffraction loops will also contain a *d*-space value for each reflection, `_refln_d_spacing`, to aid in plotting the peak positions, and a locally defined data item, `_gsas_i100_meas`, which defines relative reflection intensities. If powder data have been collected with two wavelengths, then each reflection will be listed twice, with `_pd_refln_wavelength_id` used to indicate the wavelength. Finally, when there is more than one chemical species in the powder diffraction data, data name `_pd_refln_phase_id` is also included in the loop to define the species responsible for each reflection.

4. CIF templates

As the previous paragraphs have described, a CIF contains observed and computed diffraction data, program settings and the results of the crystallographic fitting. However, this information is of limited value unless accompanied by descriptive information that defines, for example, sample preparation and data collection conditions. This sort of information is not available within Rietveld software and thus the investigator must manually input this information for it to be included in the CIF; specification of this requires a significant level of effort. Thus, providing a mechanism where data entry for one project can be reused is very important. For example, one would prefer to describe a diffractometer once and reuse these data items in future studies rather than input this information each time a new CIF is generated. Another important aspect to be considered with respect to this descriptive information is that a mechanism is needed to update the items in a CIF that are obtained from refinement results, should additional refinements be performed, without requiring a user to re-input descriptive information manually.

One idea that was considered to address the latter goal would be to have the *GSAS2CIF* program read an existing CIF when updates are performed, and then copy the descriptive information from the older version of the CIF. This would require use of a CIF parser. A Fortran-based parser, *CIFtbx*, exists, but it cannot read and write CIFs simultaneously (Hall & Bernstein, 2002). Also, *CIFtbx* would not preserve comments in the CIF and these have value to users who may add information to a CIF *via* a text editor. This approach would not address the reuse of descriptive information.

Instead, *GSAS2CIF* was designed to obtain descriptive information from separate files, which we call template files. The CIF file created by *GSAS2CIF* contains the contents of these template files combined with CIF items generated from the Rietveld refinement. By having users add descriptive information to these template files, rather than to the output CIF, the *GSAS2CIF* program can be rerun at any point to create a new CIF file that incorporates the latest information from the refinement as well as the descriptive information from the template files. Template files from one project can be reused for other projects, saving time for users.

Three separate types of template files were created for *GSAS2CIF*, based on the contents of the *Acta Crystallographica Section C* template for powder diffraction submissions (ftp://ftp.iucr.org/pub/rietform.cif). The three types of *GSAS2CIF* template files are:

(*a*) *Publication template*, which contains publication and overall information, as well as information about how a sample was prepared.

(*b*) *Species template*, which describes a specific chemical species.

(*c*) *Instrument template*, which documents a specific powder diffractometer and its configuration.

When the *GSAS2CIF* program is run, the program looks for copies of template files that have been created for the current project. If any of these files are not found, these files are created from generalized versions of the template files. The *GSAS2CIF* documentation describes the search protocol used to locate template files. This offers the ability to have different customized template files, with versions specific to projects and to diffraction instruments. If no customized version template files can be located, a universal version of the template file from the *GSAS* distribution is used. Note that each project will require *P* copies of the species template and *D* copies of the instrument template, plus a single copy of the publication template.

The *GSAS2CIF* program names each of the $D + P + 1$ template files according to the project and, where appropriate, either the species or the instrument, as described in the documentation. Information can be added to these template files using a standard text editor program. Alternately, the *EXPGUI* package contains a GUI-based program, *FillCIF* (described in §6.2 below), which has been written specifically to add information to template files. Once a template file has been populated with information on a particular project, sample or instrument, it can be reused as the basis for a customized template file.

5. Interatomic distances and angles

Interatomic distances and angles, along with estimated uncertainties, are generated in the *GSAS* program *DISAGL*. To allow the output of this program to be used in *GSAS2CIF*, the *DISAGL* program was modified to save the interatomic distances and angles in an ASCII file. Simple refinement statistics are also recorded in the file, so that *GSAS2CIF* can verify that the distance values match the coordinates read from the experiment file. When *GSAS2CIF* generates loops for the interatomic distances and angles, it includes a 'publication flag' that indicates if the value should be included in a publication. This flag value is read from the *DISAGL* output file. By default, the *DISAGL* program sets the flag value to 'n' for 'do not publish'. However, the flag can be changed with a text editor or an *EXPGUI* utility program, *CIFSelect* (described in §6.1 below). If *DISAGL* is rerun, the publication flags from a previous *DISAGL* output file are read and then reproduced in the updated output file.

6. GUI utilities for *GSAS2CIF*

As described before, the *GSAS2CIF* program reads two sets of ASCII files, namely the template files and the distance and angle file from *DISAGL*. Two GUI utility programs have been added to *EXPGUI* to aid the manipulation of these files. The capabilities of these routines are briefly presented below; more detailed discussion on how these programs are used is available in the *GSAS2CIF* documentation (http://www.ncnr.nist.gov/xtal/software/expgui/gsas2cif.html).

6.1. *CIFSelect*

The *CIFSelect* utility displays the distances and angles around a selected atom. The mouse can be used to select which distances and angles will have their publication flag set to ‘yes’. In one mode of operation, angles and distances are selected separately. In the other mode of operation, angles are automatically selected for publication when each of the two interatomic distances from that vertex atom are selected to be published.

6.2. *FILLTemplate*

The *FILLTemplate* utility is used to enter or change information in the CIF template files. This routine is used to open all the template files for a project for editing. It can locate CIF items where the value is undefined, so that a user can enter an appropriate value. The utility ensures that valid CIF syntax is maintained and checks that numerical values specified by users are within the allowed ranges specified in the CIF dictionary. When a data name is selected, the *FILLTemplate* utility program can optionally display the definition for that name from the appropriate CIF dictionary.

APPENDIX *A*
Block pointer example

This appendix provides an example of the block structure of a powder diffraction CIF resulting from the refinement of a mixture of two chemical species, utilizing two powder diffraction data sets. The example (Fig. 1) shows the use of the block id (`_pd_block_id`) and the data items that serve as pointers to this id (`_pd_phase_block_id` and `_pd_block_diffractogram_id`). Note that, for simplicity, `_pd_proc_ls_profile_function` has been removed from the phase table.

```
data_NISI_publ
#=============================================================================
# this block describes the project, paper, authors, sample prep info
#=============================================================================
_pd_block_id   2002-12-22T17:32|NISI|Brian_H._Toby|Overall

data_NISI_overall
#=============================================================================
# this block has refinement statistics (from GSAS)
#=============================================================================

# pointers to the phase blocks
loop_   _pd_phase_block_id
      2002-12-22T17:32|NISI_phase1|Brian_H._Toby||
      2002-12-22T17:32|NISI_phase2|Brian_H._Toby||
# pointers to the diffraction patterns
loop_   _pd_block_diffractogram_id
      2002-12-22T17:32|NISI_H_01|Brian_H._Toby|GPD
      2002-12-22T17:32|NISI_H_02|Brian_H._Toby|GPD

data_NISI_phase_1
#=============================================================================
# Information about phase 1
#=============================================================================
_pd_block_id   2002-12-22T17:32|NISI_phase1|Brian_H._Toby||

# Pointers to histograms that defined phase 1
loop_   _pd_block_diffractogram_id
  2002-12-22T17:32|NISI_H_01|Brian_H._Toby|GPD
  2002-12-22T17:32|NISI_H_02|Brian_H._Toby|GPD

data_NISI_phase_2
#=============================================================================
# Information about phase 2
#=============================================================================
_pd_block_id   2002-12-22T17:32|NISI_phase2|Brian_H._Toby||

# Pointers to histograms that defined phase 2
loop_   _pd_block_diffractogram_id
  2002-12-22T17:32|NISI_H_01|Brian_H._Toby|GPD
  2002-12-22T17:32|NISI_H_02|Brian_H._Toby|GPD

data_NISI_p_01
#=============================================================================
# Information about data set #1
#=============================================================================
_pd_block_id   2002-12-22T17:32|NISI_H_01|Brian_H._Toby|GPD

# phase table for data set #1
loop_     _pd_phase_id
          _pd_phase_block_id
          _pd_phase_mass_%
          _pd_proc_ls_peak_cutoff
  1  2002-12-22T17:32|NISI_phase1|Brian_H._Toby||  51(49)  0.00500
  2  2002-12-22T17:32|NISI_phase2|Brian_H._Toby||  49(49)  0.00500

data_NISI_p_02
#=============================================================================
# Information about data set #2
#=============================================================================
_pd_block_id   2002-12-22T17:32|NISI_H_02|Brian_H._Toby|GPD

# phase table for data set #2
loop_     _pd_phase_id
          _pd_phase_block_id
          _pd_phase_mass_%
          _pd_proc_ls_peak_cutoff
  1  2002-12-22T17:32|NISI_phase1|Brian_H._Toby||  51.38       0.00500
  2  2002-12-22T17:32|NISI_phase2|Brian_H._Toby||  48.62(28)   0.00500

#--eof--eof--eof--eof--eof--eof--eof--eof--eof--eof--eof--eof--eof--eof--eof--
```

Figure 1
Block pointer example.

The authors would like to thank all users of this software who have taken the time to document and report problems, and Brian McMahon for assistance with CIF and comments on an early draft of this paper. Certain commercial products are identified in this report in order to describe the documented software adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that these products are necessarily the best available for the purpose.

References

Bourne, P. E., Berman, H. M., McMahon, B., Watenpaugh, K. D., Westbrook, J. D. & Fitzgerald, P. M. D. (1997). *Methods in Enzymology*, Vol. 277, pp. 571–590. New York: Academic Press.
Hall, S. R., Allen, F. H. & Brown, I. D. (1991). *Acta Cryst.* A**47**, 655–685.
Hall, S. R. & Bernstein, H. J. (2002). *CIFtbx*, http://www.bernstein-plus-sons.com/software/ciftbx/.
Larson, A.C. & Von Dreele, R.B. (1986). *General Structure Analysis System* (*GSAS*). Report LAUR 86-748. Los Alamos National Laboratory, New Mexico, USA.
Rietveld, H. M. (1969). *J. Appl. Cryst.* **2**, 65–71.
Toby, B. H. (2001). *J. Appl. Cryst.* **34**, 210–213.
Toby, B. H. (2003). *International Tables for Crystallography*, Vol. G, *Definition and Exchange of Crystallographic Data*, edited by S. R. Hall & B. McMahon, ch. 3.3, *Classification and Use of Powder Diffraction Data*. In the press.