Journal of
Applied Crystallography

ISSN 0021-8898

Received 18 November 2008
Accepted 15 December 2008

OLEX2: a complete structure solution, refinement and analysis program

Oleg V. Dolomanov,* Luc J. Bourhis, Richard J. Gildea, Judith A. K. Howard and Horst Puschmann*

Department of Chemistry, Durham University, South Road, Durham DH1 3LE, UK. Correspondence e-mail: oleg.dolomanov@durham.ac.uk, horst.puschmann@durham.ac.uk

New software, *OLEX2*, has been developed for the determination, visualization and analysis of molecular crystal structures. The software has a portable mouse-driven workflow-oriented and fully comprehensive graphical user interface for structure solution, refinement and report generation, as well as novel tools for structure analysis. *OLEX2* seamlessly links all aspects of the structure solution, refinement and publication process and presents them in a single workflow-driven package, with the ultimate goal of producing an application which will be useful to both chemists and crystallographers.

1. Introduction

More and more frequently, chemists are expected to perform many aspects of the X-ray structural analysis process, including structure solution and structure refinement, without expert help. This requires not only a basic knowledge of crystallography, but also, crucially, a reasonable understanding of the quirks of the associated software. The majority of the crystallographic programs currently in use were written a long time ago and are often lacking in user-friendliness and/or comprehensive graphical user interfaces (GUIs). This leads to a situation in which the optimal use of both staff time and available crystallographic techniques is not achieved. We have approached this problem by the creation of simple-to-use software with the aim of providing both numerical and visual feedback to the user. The main purpose of *OLEX2* is to provide a comprehensive GUI which will not require any specialist training and which will be accessible and usable to both knowledgeable chemists and experienced crystallographers.

2. Program description

OLEX2 provides three-dimensional graphical visualization of the crystallographic structure as well as the means to manipulate and analyse it. *OLEX2* reads several chemical/crystallographic file formats and displays the structure on the screen. The user can interact with the structure model using either the mouse or the command line. A screenshot of *OLEX2* is shown in Fig. 1.

2.1. OpenGL: three-dimensional graphics

OpenGL (Khronos Group, 2008) is an application programming interface (API) to hardware-accelerated graphics. This enables the generation of high-quality and fully customizable displays of three-dimensional objects on a standard computer monitor. In particular, up to eight uniform or directional light sources can be employed, different visual appearances (*e.g.* ambient or diffuse colours) for the displayed objects can be set and various lighting models (which define ambient scene light intensity and how light reflection angles are calculated) can be defined. The program provides a user interface to customize these settings and the visual appearance of all graphical objects.

2.2. Atom display

OLEX2 provides all of the conventional representations of molecules: displacement ellipsoids, balls and sticks, and wire-frame and sphere-packing drawings. The displacement ellipsoid drawing for an isotropic model displays atom radii proportional to U_{iso} and helps with the visual observation of errors in the current model. The translucent representation of the residual electron-density peaks proportional to the peak heights helps to highlight missing atoms on the background of the difference electron map artefacts.

2.3. Design and implementation

OLEX2 is fully portable and runs on Windows, MacOSX Leopard and Linux platforms. The *OLEX2* GUI is written in an extended HTML language and provides the user with a set of GUI controls supporting event-driven execution, allowing the user to perform structure solution, refinement and report generation utilizing a clearly laid out and easy-to-follow workflow path. The GUI can be easily modified and extended to suit the individual user's requirements. With the automatic updates option switched on, the *OLEX2* development framework allows the updating of the GUI in an incremental fashion on program start-up.

OLEX2 is designed as a set of libraries, which can be re-used separately to build applications with as few dependencies as possible.

OLEX2 uses wxWidgets (2008) for its GUI. However, the underlying libraries are designed to avoid this dependency. These libraries provide most of the API for *OLEX2* and manage crystallographic file input/output operations and symmetry-related tasks.

The program is built as a Unicode (Unicode Consortium, 2008) program and the *OLEX2* GUI has multilingual capabilities with website support. Users interested in translation of the *OLEX2* GUI can register on the *OLEX2* website (Puschmann, 2008) and provide translation of given phrases into the language of interest. We update this information on a regular basis but we cannot guarantee the consistency or correctness of the translation.

2.3.1. Implementation languages. The core of *OLEX2* is written in C++, while some of its extended functionality is provided by the Python programming language (Python Software Foundation, 2008).

Most of the utilities provided by the *OLEX2* core are accessible from the command line as well as from the task-oriented GUI.

The *OLEX2* executable and underlying libraries export some functionality into the Python scripting language. These functions are used in the GUI to interact with the *OLEX2* core and they also provide the basis for extending functionality while taking full advantage of Python: ease of development, fast prototyping, readability and portability of code. There is also a facility in *OLEX2* to register Python functions to be called on certain events (call-back functions) or to expose them to the user through the console/HTML.

A Python layer allows the integration of the *smtbx* (Bourhis *et al.*, 2007) structure determination and refinement procedures into the *OLEX2* GUI. This layer exploits the *OLEX2* functionality to communicate data between *OLEX2* and the *smtbx*. *OLEX2* exposes the structure model and relevant information, such as available restraints and atom treatment, to the Python layer, which makes it easy to control, read and write the model from/to *OLEX2*.

A simple macro language, designed for users to develop their own console commands, is also provided. These commands are intended to group together other commands and can be used from the GUI or accessed directly through the command line. A user-editable macro file is provided with *OLEX2* and can be used by the user to customize the behaviour of certain operations, such as file loading, application start-up *etc.* These macros are also used internally to synchronize the states of Python, the *OLEX2* core and the GUI, and provide a powerful tool to intervene in various stages of structure solution and refinement.

2.3.2. User interface. The *OLEX2* main window consists of the graphical display and an HTML panel of adjustable width and position. The main window also has a customizable menu bar and an input console (Fig. 1).

OLEX2 provides a mouse-driven interface to interact with the graphical model as well as a console to input commands using the keyboard. Most of the console commands are accessible from the task-oriented GUI. The syntax of the command line input resembles that of the *XP* program [part of the *SHELXTL* system of computer programs (Sheldrick, 2008)], whilst providing more functionality and flexibility through the use of built-in or user-defined functions and a reference to the graphical objects *via* the selection.

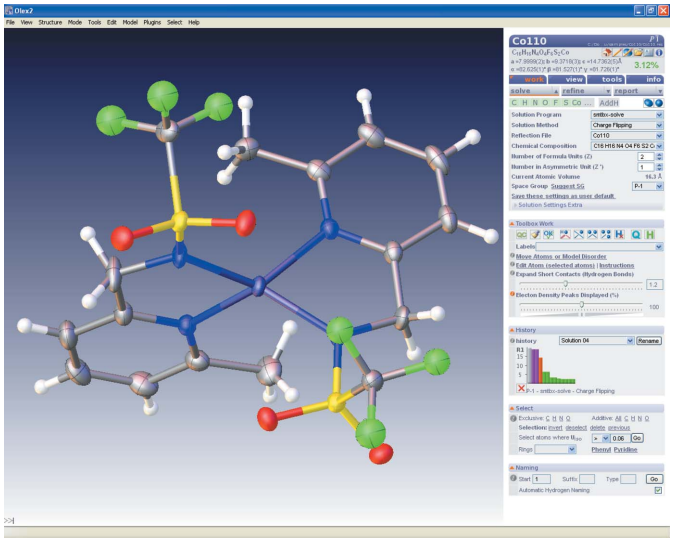

Figure 1
View of the *OLEX2* main window. Structure from Congreve *et al.* (2003).

The GUI is dynamically adjusted according to other known crystallographic software that is installed on the system and allows the user to select any particular software for the structure solution and refinement stages. All default parameters of these programs can be inspected and modified; any relevant help information is provided when needed.

2.4. Functionality

OLEX2 aims to be an all-encompassing tool for performing the analysis of small-molecule crystal structures. Following the three-step workflow (structure solution, refinement and report generation), there are some further tools which assist with structure analysis and publication.

2.4.1. History. Some of the extended *OLEX2* features include History, which records every structure solution/refinement step and provides visual feedback of the related *R* factors. The History module facilitates easy switching between different structure solution and refinement steps.

There is a CIF (crystallographic information file; Hall *et al.*, 1991) metadata module in the *OLEX2* GUI, which helps with data harvesting and report generation. This module analyses the files in the structure directory and extracts all possible information, and this information is later used to resolve missing items in the CIF and structure report. The metadata GUI provides the user with comprehensive controls to input data into the CIF. This information, as well as harvested data, is stored internally and the CIF is updated every time the report is created, thus ensuring that the information provided by the user is not lost in the structure refinement or solution steps.

2.4.2. File I/O and file manipulations. The software can import and export structural data *via* a number of crystallographic file formats [*SHELXL* model files (Sheldrick, 2008), CIF, MDL MOL, PBD, XYZ, XD master files], generate the extended structure if required, produce an output image and provide a structure refinement report.

2.4.3. Data analysis. The user can open a P4P (proprietary Bruker AXS file format; file containing experimental details such as unit-cell dimensions, asymmetric unit content, data collection temperature and radiation wavelength used), HKL or raw data reflection file (provided the cell dimensions are given) to initiate the *OLEX2* space group determination routine. If the space group determination routine comes to a single choice of the space group, *OLEX2* attempts to solve the structure automatically and displays the result to the user. Otherwise, the user can input the known space group or investigate the output of the *OLEX2* space group analysis procedure to make a decision.

The space group determination routine implemented in *OLEX2* uses analysis of systematic absences and merging statistics complemented by the Wilson (1949) statistics for the determination of the centre of symmetry.

2.4.4. Structure determination. Currently *OLEX2* features our own charge-flipping (Oszlányi & Sütő, 2008) structure solution routine based on the *smtbx*, as well as supporting *SHELX* (Sheldrick, 2008) structure solution programs. We are also working towards the support of other available structure solution programs such as *SIR97* (Altomare *et al.*, 1999).

In addition to our own *smtbx*-based refinement (Bourhis *et al.*, 2009), *OLEX2* also supports the *SHELXL* refinement program (Sheldrick, 2008). *OLEX2* provides tools to visualize the output of the refinement (*e.g.* occupancies) on-screen, as well as tools to manipulate the model, restrain refinement parameters, model disorder and perform structure analysis. The refinement module

provides a completely transparent mechanism of changing the reflection file against which the structure is refined.

2.4.5. Structure analysis. The structure analysis capabilities include calculation and visualization of voids and Fourier maps. Different drawing styles, such as two-dimensional projection onto the screen or three-dimensional surface/mesh drawing, are available. Other tools include fragment matching and alignment (Kearsley, 1989). The alignment either works automatically for fragments with the same connectivity or can be interactively driven by the user. A number of tools and modes are available for structure growing and packing. Other standard functions include distance and angle calculation between atoms, bonds or user-defined planes; standard uncertainties of these parameters can also be estimated when a variance–covariance matrix is available from the refinement. Tools to calculate the molecular composition and isotope pattern distribution of the current structure or provided formula are also included in *OLEX2*.

3. Software environment and system requirement

OLEX2 for Windows is pre-compiled and runs under 32- and 64-bit Windows 2000/XP/Vista. Some Linux distributions are available at the Debroglie repository (http://www.debroglie.net); others can be compiled using the scripts and documentation provided [currently tested for Fedora Versions 7 and 8, SUSE Versions 10.1 (32-bit) and 10.3 (64-bit), and Ubuntu Version 7.10 (32-bit)]. The user interface is developed using the wxWidgets library. The source code of the program is developed using Microsoft Visual Studio Free Edition 2005/2008.

The system requirements for running *OLEX2* are 128 Mb of free RAM, 50 Mb of disk space and a graphics card that supports OpenGL.

4. Documentation, support and availability

OLEX2 is in the ongoing development phase and we are striving to document all aspects of *OLEX2* comprehensively. There is a web portal at http://www.olex2.org which contains descriptions of the *OLEX2* functionality and some of its functions. Anybody interested in development of *OLEX2* or its GUI is welcome to join the portal.

OLEX2 is an open source project and executables are available in pre-compiled form at http://www.olex2.org. The source code resides at the SourceForge website (http://www.sourceforge.net) and can be freely downloaded and compiled for any system. *OLEX2* is distributed under a BSD-style licence

The authors thank the EPSRC (UK) for financial support. We are grateful to our testers and contributors (Sebastian Cirkel, Helena Shepherd, John Warren, Ilia Guzei) for their time, attention and thorough evaluation of the program, to Ehmke Pohl for assistance in the preparation of this publication and to Aileen Congreve for provision of the structure in Fig. 1.

References

Altomare, A., Burla, M. C., Camalli, M., Cascarano, G. L., Giacovazzo, C., Guagliardi, A., Moliterni, A. G. G., Polidori, G. & Spagna, R. (1999). *J. Appl. Cryst.* **32**, 115–119.

Bourhis, L. J., Dolomanov, O. V., Gildea, R. J., Howard, J. A. K. & Puschmann, H. (2009). In preparation.

Bourhis, L. J., Grosse-Kunstleve, R. W. & Adams, P. D. (2007). *IUCr Commission on Crystallographic Computing Newsletter*, No. 8, pp. 74–80. http://www.iucr.org/resources/commissions/crystallographic-computing/newsletters/8.

Congreve, A., Kataky, R., Knell, M., Parker, D., Puschmann, H., Senanayake, K. & Wylie, L. (2003). *New J. Chem.* **27**, 98–106.

Hall, S. R., Allen, F. H. & Brown, I. D. (1991). *Acta Cryst.* A**47**, 655–685.

Kearsley, S. K. (1989). *Acta Cryst.* A**45**, 208–210.

Khronos Group (2008). *Open-GL – the Industry Standard for High Performace Graphics*, http://www.opengl.org.

Oszlányi, G. & Sütő, A. (2008). *Acta Cryst.* A**64**, 123–134.

Puschmann, H. (2008). *The OLEX2 Website*, http://www.olex2.org.

Python Software Foundation (2008). *Python Programming Language*, http://www.python.org.

Sheldrick, G. M. (2008). *Acta Cryst.* A**64**, 112–122.

Unicode Consortium (2008). *The Unicode Standard*, http://www.unicode.org.

Wilson, A. J. C. (1949). *Acta Cryst.* **2**, 318–321.

wxWidgets (2008). *wxWidgets – Cross-Platform GUI Library*, http://www.wxwidgets.org.