

# Mosflm 7.0.1 and its new interface - iMosflm 0.5.3

Harry Powell, Andrew Leslie and Geoff Battye  
MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH  
Email: [harry@mrc-lmb.cam.ac.uk](mailto:harry@mrc-lmb.cam.ac.uk), [andrew@mrc-lmb.cam.ac.uk](mailto:andrew@mrc-lmb.cam.ac.uk)

## Introduction

CCP4 provided the funding to allow a new graphical user interface (GUI) to be developed for the diffraction image integration program Mosflm. The result, known as iMosflm, is a new interface that has a straightforward and logical flow through processing diffraction data. It has been designed so that a novice user can easily process datasets and it also has the functionality to allow more advanced users to deal with difficult data. The feedback to the user has been tailored to provide the most useful items in as clear a way as possible so that the various processing parameters can be optimized. Output which was only available in Mosflm in tables is now presented by iMosflm in the most appropriate form, e.g. pie-charts, histograms and line graphs. Variables being refined during processing are displayed as real-time graphs, which allow the user to see immediately if any problems arise.

iMosflm is written in TclTk, and can be installed and run on most modern computing platforms, including Linux, OS X and Microsoft Windows.

Underlying changes have been implemented in Mosflm to facilitate the development of iMosflm, and which also make processing more robust and less likely to cause the user problems. Mosflm itself can be run from the command-line (e.g. in a UNIX shell window or a MS-Windows command prompt), so it can be used in shell scripts, and the traditional X11 GUI is still available for all platforms for which it was previously available.

The vast majority of the work on iMosflm was carried out by Geoff Battye between 2003 and 2006; this culminated in the release of iMosflm 0.4.5 in August 2006, which required the installation of a beta-release copy of Mosflm version 6.2.6. The latest version (described here, v0.5.3), is the first to run with a full release copy of Mosflm (v7.0.1).

Both iMosflm and Mosflm can be obtained by following the links at <http://www.mrc-lmb.cam.ac.uk/harry>. A normal pre-requisite is that the CCP4 suite should be set up correctly on the target platform.

## Background

Mosflm is an integration program with a long history, with its origins dating back to the 1970's. A GUI was first suggested in 1992, and this was implemented using John Campbell's "xdl\_view" X-windows package shortly after. While this represented a great leap forward in usability, by the end of that decade it was beginning to look rather old-fashioned and its fairly rigid format was starting to restrict development; its use also prevented porting Mosflm straightforwardly to MS-Windows. The decision was made to develop a new, more portable, interface using modern object-oriented design principles; at the time, the scripting language TclTk with the [incr]Tcl and [incr]Tk extensions was well-

established and provided the required functionality. Object-oriented programming (OOP) is the modern standard method in software development for a variety of reasons, including straightforward re-use of code and modular implementation of functionality.

There are two main ways of interfacing a GUI with another program; communication can be carried out by parsing output in log files, or a more intimate connection can be made by using TCP/IP sockets and applying a consistent structure to the information passed between the components. A ubiquitous example of the latter is the use of encapsulating information in HTML to pass from a web server to a browser. We chose to use sockets for linking iMosflm/Mosflm, and developed a local XML (eXtensible Markup Language) for encoding information to be sent from Mosflm to iMosflm; all communication in the other direction is in the form of standard Mosflm commands. The dictionary of available commands was extended to allow for this new method of running Mosflm; a consequence of this has been that Mosflm itself has become more flexible in its command-line scripting.

## Installation of iMosflm 0.5.3

Full installation instructions are on the website.

The following components should be installed on your system;

- Current CCP4 distribution
- TclTk 8.4
- Mosflm version 7.0.1
- iMosflm source

Since the normal CCP4 installation process now gives the option of installing the ActiveTcl TclTk package, this does not need to be repeated for iMosflm. A copy of TclTk built for Tru64 UNIX is available on the iMosflm download site.

Mosflm 7.0.1 can be downloaded as a pre-built binary for the available platforms, or built from the Mosflm source code by the user.

iMosflm 0.5.3 is distributed as a tar'd and gzip'd archive containing over 200 files; these are mostly ASCII source files and no compilation or other building is necessary. The MS-Windows version is distributed as a zip file, containing the iMosflm source as well as a Mosflm 7.0.1 executable and a Tcl script file for launching the interface.

Installation should be straightforward. The archive should be extracted using

- `tar -xzf imosflm.tgz` (UNIX-based systems)
- double-click on the "imosflm.zip" icon (MS-Windows) and follow the instructions.

iMosflm 0.5.3 runs on Linux, Mac OS X, Tru64 UNIX and MS-Windows. There are problems with porting some of the TclTk components to Irix, so it is unlikely that iMosflm will run on SGI MIPS workstations.

### UNIX-based systems

You need to set up the environment variables MOSFLM\_WISH (to point to the "wish" you are using) and MOSFLM\_EXEC (to point to the Mosflm executable).

Run the program by typing the full path to the script (or put it in your path) "imosflm/src/imosflm"; this file contains a Bourne shell script to set up variables and perform a number of checks before trying to start the interface itself.

A new feature in version 0.5.3 is the addition of command-line parameters; start imosflm with the option "--help" for a full list, which will be added to in the future.

## MS-Windows

Double-clicking on the script "imosflm.tcl" in the top-level "imosflm" folder takes care of setting up the various environment variables for you (do not confuse this with the file of the same name in the folder "imosflm\src"), and starts iMosflm itself.

## Running imosflm

On any platform, you should be rewarded by the appearance of an interface with a large, mostly blank pane; this has a number of large buttons down the left-hand side (see Figure 1), and a number of options across the top. The best source of information on how to run the program is the tutorial available online in the iMosflm documentation at <http://www.mrc-lmb.cam.ac.uk/harry>, but it is worthwhile going through the options briefly here, to give an idea of how to run the interface.

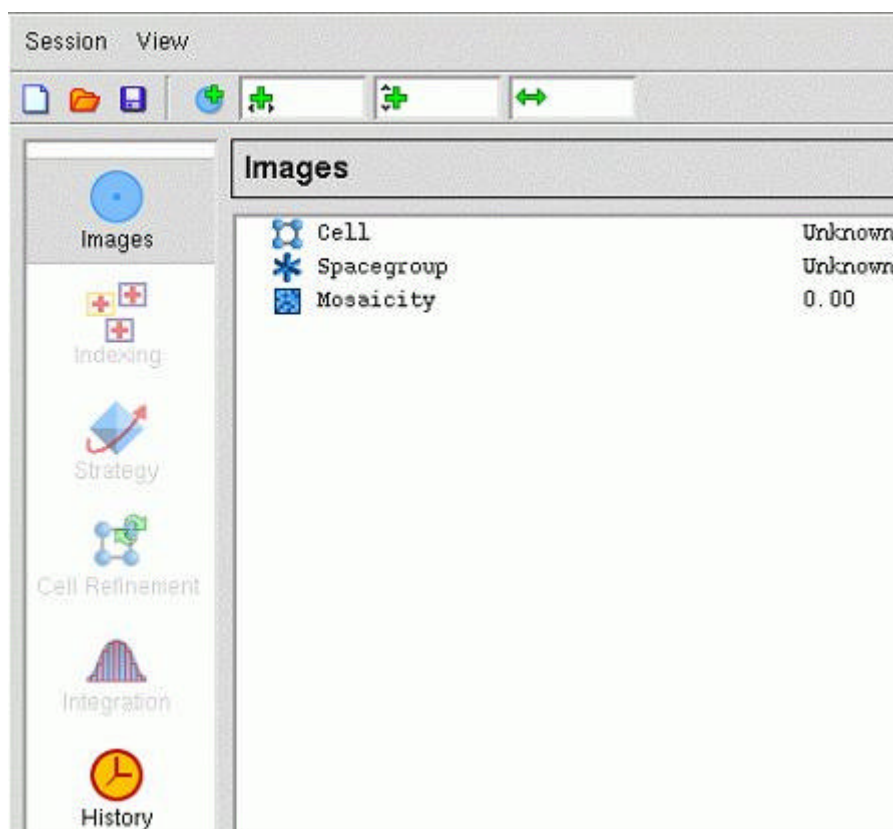
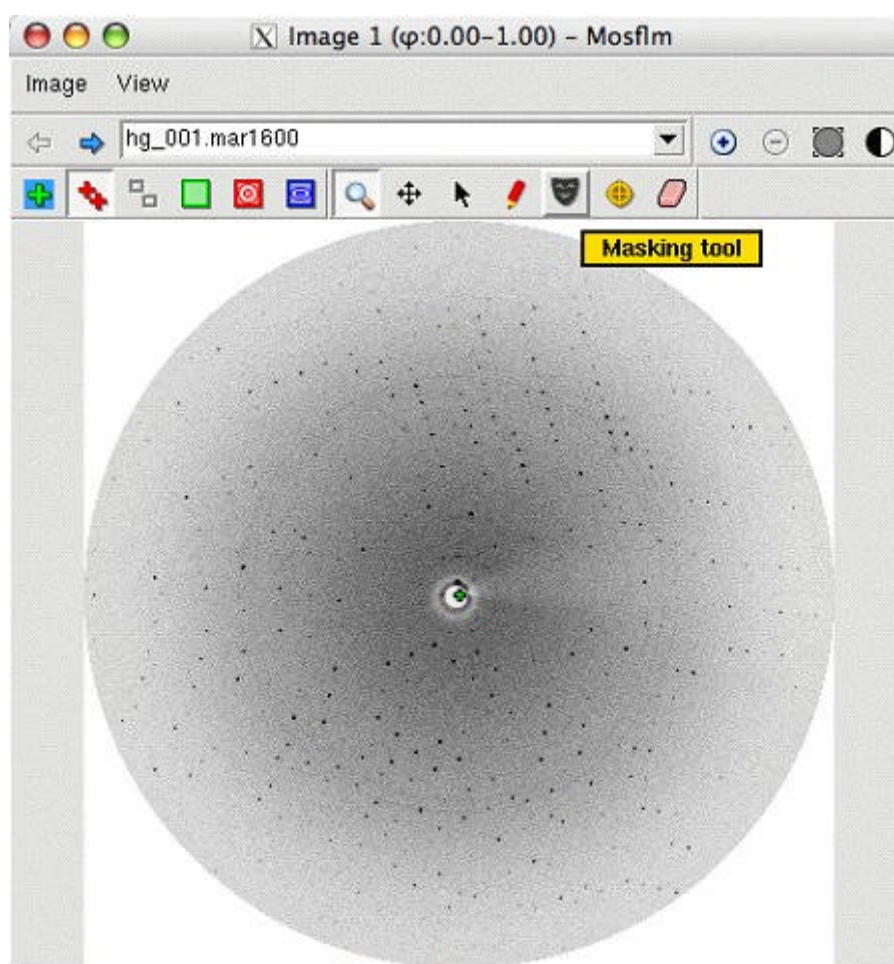


Figure 1. Appearance of the iMosflm GUI on starting the program.

Any options not available at the current stage of processing are greyed out, and cannot be selected. At the start of the process, therefore, the only options available from the list down the side of the GUI are to choose "Images" or to examine the "History". The other options, i.e. "Indexing", "Strategy", "Cell Refinement" and "Integration" cannot be done before

images have been selected so these are greyed out. Clicking the mouse on either "Session" or "View" (both on the top line) gives a drop-down menu with further options. The four buttons immediately below are (reading from left to right) to start a new session, read an existing session or save the current session (in a file which can be read in later) and to read in an image list for processing.

Once an image list has been read, the "Indexing" option becomes live since this is now a possible action. In addition, an image display window opens which displays the first image in the list chosen (Figure 2). This has a number of options to enhance the display in various ways, each of which can be chosen from the menu bar. Note that "tooltips" appear which describe the action of the buttons if the mouse cursor is left over them for a few seconds.



*Figure 2. The iMosflm image display window.*

Following indexing, all actions are possible, so the "Strategy", "Cell Refinement" and "Integration" buttons become live. As indicated above, the best place for further details on running the interface is the tutorial.

## Changes in Mosflm (ipmosflm) 7.0.1

The major change over previous versions (6.2.6 and earlier) is that the autoindexing code has been enhanced considerably to make it more robust; it can now index images of much lower quality than was possible previously in an entirely automatic way. In addition, the detector parameter refinement has been improved to make it more stable when processing weak images (it should no longer be necessary to "fix" parameters like "tilt" and "twist").

New detectors supported by Mosflm are the ADSC Q270 and the Dectris Pilatus pixel array detector (support for this is not yet included in iMosflm).

Other changes are enhancements to communicate with iMosflm, the DNA automation project and many small bug fixes. The change in major version number indicates that this is the first Mosflm version which deals properly with iMosflm.

The bugs that have been addressed mean that Mosflm will prove to be an even more robust program to process diffraction images. With the exception of the MS-Windows version, all Mosflm executables include the traditional X11 GUI that uses the xdl\_view libraries.

Mosflm will now build and run on MS-Windows, using either Visual-C and Visual Fortran or the Mingw gcc/g77 pairing; indeed, the MS-Windows executable distributed with iMosflm is built on a Macintosh running OS X and using a cross-compiler version of gcc/g77.

## Acknowledgements

In particular, we would like to thank François Remacle for his help in preparing the MS-Windows port using Visual C and Fortran. Also, we would like to thank all the users who have provided us with bug reports and suggestions for improvements and who have tested the software so thoroughly.