1 Introduction

This describes the functionality required by the labelit.screen wrapper. Note to self - this may be superceeded by the wrapper for labelit.index. The latter appears to have the same functionality as labelit.screen - though this still attempts to run mosflm so maybe it's not worth the bother.

2 Modifications

2.1 13/JUN/06

Added getting of the mosaic spread (under) estimate.

2.2 Notes from the Documentation

On the setting of the beam centre etc. as input:

```
Normally, critical model paramters such as the beam position are taken
from the file header.  If for some reason these parameters are wrong,
it may be impossible to index the dataset unless LABELIT is instructed
with override values.  In the current working directory, create a file
called dataset_preferences.py with any or all of the following items:

autoindex_override_beam = (93.2,100.4) #your xy beam coordinates in mm
autoindex_override_distance = 250.00   #your detector distance in mm
autoindex_override_wavelength = 0.9793 #your X-ray wavelength in Angstroms

...Now labelit.reset followed by labelit.index will use these new parameters.

The file dataset_preferences.py is just one of three locations where
configurable parameters can be set. Parameters can be defined in any
of the following locations, with definitions taking precedence when
they occur in locations further down this list:

labelit_sources/labelit/labelit/site_preferences.py.

A location where the system administrator can set parameters shared
by all users.

HOME/.labelit_preferences.py.

A place where a particular user can define settings applicable to all
work done under that unix account.

dataset_preferences.py.
Placed in the current working directory, this file contains parameters
applicable only to the particular dataset being analyzed.
```

3 Use Cases

3.1 UC 1: Determine Beam Centre

This is simply a case of running labelit.screen –index_only to determine the appropriate centre for the direct beam position. This would be used for instance when I want to perform the autoindexing with Mosflm but want to refine the beam centre first.

3.2 UC 2: Perform Autoindexing

This is the default case - autoindexing with no prior knowledge of what the beam centre is. This should include the possibility to provide "correct" values for the beam centre, distance and wavelength (see above for details.)

This will provide the correct beam centre, unit cell and lattice but will not in itself compute the matrix file for mosflm. This will be done externally via labelit.mosflm_script.

3.3 UC 3: Perform Autoindexing with Provided Lattice

As for UC2 but with the additional functionality that the "correct" lattice can be assigned beforehand. If this is in the list of possible lattices then this solution should be returned. If not, an exception should be raised.

4 Implementation

4.1 Notes

May be worth running labelit.reset first to delete all of the bad solutions. Is it possible to switch off the beam centre refinement? When running with only one image this can actually give the wrong result and takes a while. This can be achieved by setting the following:

```
beam_search_scope = 0.0

# put the exactly correct values in here
autoindex_override_beam = (x, y)

=> dataset_preferences.py
```

It's worth noting that this makes the indexing much faster! When autoindexing the stats from distl can be obtained from labelit.stats_distl. This is now also implemented as a wrapper.

4.2 Useful Test Images

The following images illustrate some useful features:

- ESRF TEST SUITE:3.1 - shows nearly tetragonal mP crystal example. This has `;(` records in it when run with Labelit!