1 Introduction

This describes the (rather trivial) interface to the CCP4 program Sortmtz. This program sorts unmerged reflection files, and may be used to include a number of reflection lists in a single data set.

2 Use Cases

2.1 Simple: Sort a Reflection File

This is simply a case of sorting a reflection file into a form suitable for scaling with Scala. The default sort order is "H K L M/ISYM BATCH".

This will need to:

- Trap non-mtz file input.
- Trap missing input files.
- Handle input of exactly one input file.
- In the event of an error; delete the offending output file.

2.2 More Complex: Sorting Many Reflection Files

This is a more complex use case where the objective is to sort a number of reflection files together. These could result from collection of a low and high resolution pass (and hence using `rebatch`) or from processing the data from a single sweep in batches.

This will need to:

- Verify that the reflection indices will be unique e.g. no repeated batches.
- Handle more than one input file.
- Handle the above cases.

3 Resulting API

3.1 UC1

```
hklin = os.path.join(xia2core,
                     'Data', 'Test', 'Mtz', '12287_1_E1.mtz')
s = Sortmtz()
s.setHklin(hklin)
s.setHklout(os.path.join(os.environ['CCP4_SCR'],
                         'temp-test-sortmtz.mtz'))
status = s.sort()
```

In this example the status will be "Normal termination", which is good. If hklin is a non-mtz file a RuntimeError exception will be raised with "File not identified as MTZ (Error)" as the text. If the file does not exist then a RuntimeError exception will be raised with argument "Cannot find input file (Success)". The "success" here comes from the CCP4 MTZ library.

3.1.1 Unit Tests

Requires data from XIACore unit tests. This is in Data/Test/Mtz. The tests are implemented in TestSortmtz.py:

```
testdefault() -> test with the 12287 data; should work
testnotmtzfile() -> test with text file in file path; should fail
testnofile() -> test with missing input file; should fail
```

3.2 UC 2

For this use case I implemented a "addHklin()" method which manages it's own list of reflection files. If this is empty then the default CCP4 getHklin() method will be assumed.

3.2.1 Unit Tests

Requires data from DPA unit tests (this is ok) small data sets. The tests merge two separate runs then the same run twice. The second should raise an exception.

```
testmultiple() -> test merging two reflection files
testmultiplebadbatches() -> test merging two reflection files with the
                            same batch numbers; should fail
```