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.

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.
- 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.

3.2 UC 2