1 XDS Indexer Implementation

1.1 Introduction

The XDS indexer works in a slightly complicated way, as in the XDS processing pipeline there is a substantial amount of processing which is performed *before* the crystal lattice is indentified (XYCORR INIT.)

The initial processing steps in XDS define the layout of the detector, assess the gain and look for "dead" areas of detector. Although all of these are prerequisite to indexing, the results are also required for integration. The results of all processing in XDS is files, which means that these need to either be picked up and carried in memory or linked - I have chosen the former for simplicity for all non-reflection files.

The input and output files for each XDS task are clearly defined in the XDS documentation - identifying them is therefore no problem. The "flow" of these files is shown in [FIXME create figure here.]

Fortunately the Indexer specification includes a payload - this will be used.

1.2 Solution Selection

The solution selection is based on the 0-999 penalty, and the assertion in the documentation that the penalty for a good solution is usually less than 20 - the highest symmetry solution with a penalty if less than 30 is therefore selected.

XDS gives possible indexing solutions for 44 "lattice characters". However, in this process I am interested only in the best solutions, so I am selecting for each lattice the solution with the lowest penalty as the "correct" one. This presumes that the penalty function behaves itself, but this is reasonable.

The indexing solution management then proceeds as described in the general documentation.