Appendix B Adding File Types

Overview

This appendix outlines the procedures you should follow when adding new file formats to NCSA Reformat. If you do add any new file types, NCSA developers would appreciate it if you could send them your code which they could then include in future distributions. (Refer to this document's README or Bugs pages for the NCSA address.)

Procedure

1. Write an appropriate conversion routine. Look at any of the current conversion routines to see what sort of interface is required of the top-level conversion routine. The routine must return a value of type ErrorCode. You'll need to include "`error.h`", and take a look at "`error.c`" for the strings that map to the codes. If you want to issue any error messages, add a string for the name of your routine to the table '`names`' in "`error.c`", and an entry (NOT the same as the function name - mine are all close, but not identical) in the '`FuncName`' enumerated type in "`error.h`". Make sure to add the string and the enumerated type value in the same position - the latter is used as an index to find the former.

2. Decide on a short (3-8 letter) name for your new type. Add this name to the enumerated type 'DataType' in "types.h". Also, make a new #define statement for the new type. Now, find the table '`FileTypeMap`' in "`cmdline.c`". Add a new entry to that table for the new type.

3. Locate the table '`cvtFuncs`' in "`reformat.c`". This is a large matrix of function pointers. Each entry (i, j) in the matrix is the actual name of the function used to convert from type i to type j. A NULL entry means that no such routine exists. Note that most of the entries are NULL, except those in the HDF column. Add a new row to the table, making sure to place it such that its index corresponds to the index of the new type (as defined in the enumerated type in "`types.h`"). If your new type requires the PBM library, make an entry in the PBMTypeMap list.

4. Now find the table '`writeFuncs`' in "`reformat.c`". Add an entry (the name of the routine) for the procedure that writes out data of your new format. Even if you don't need to write out data in that format, you MUST have such a routine. Look at some of the examples in the distribution for the interface. This is a design drawback. Make sure that the new entry is in the correct position - this table is also indexed by the enumerated type of file types. Write a stub for the routine. Add the name and a new entry in "`error.h`" and "`error.c`" (as explained

above) for error reporting.

5. Add the name of the file to the list of object files in the Makefile. Compile it and try it out. The new type will be available as a command line option in reformat, and as a selectable type in XReformat.