

**Instruction manual for the
UM-MCIP meteorological pre-processor**

G J Taylor

Draft

Document Version: 0.2

Date: July 2006

Title: Instruction manual for the UM-MCIP meteorological pre-processor

Author: G. J. Taylor

Aeolus
5 Thornton Close,
Guildford,
Surrey GU2 9XE, UK.

Tel: +44 (0)1483 505921
e-mail: geoff@aeoltech.co.uk

Date: 12 July, 2006

Prepared for: University of Hertfordshire

Summary: This document provides instructions on how to build and run the UM-MCIP meteorological pre-processor.

Acknowledgements: This work has been funded by the Universities Weather Research Network (UWERN) through a contract placed with, and administered by, the University of Hertfordshire.

The work is substantially based on the MCIP processor developed by the US EPA.

The following individuals have contributed to this contract:
Peter Clark, Richard Forbes (JCMM, University of Reading)
Ranjeet Sokhi, Ye Yu (University of Hertfordshire)
Kevin Marsh (BADC, Rutherford Appleton Laboratory)
Chang Wang (UWERN)

All intellectual property rights including without limitation all copyrights, database rights, patents, design rights, trademark rights and other intellectual property and proprietary rights in this document and its associated materials and appendices (the "Materials") belong to the University of Hertfordshire Higher Education Corporation ("UH") or its licensors.

The Materials must not be displayed, modified, adapted, distributed, transmitted, communicated, copied, transferred, lent, printed off, published or otherwise reproduced either in part or in whole in any form by any means without the written permission of UH.

UH will use every reasonable effort to ensure that the information contained in the Materials is accurate at the time you receive it. However, you acknowledge and agree that UH accepts no liability whatsoever in contract, tort or otherwise for any loss or damage caused by or arising directly or indirectly in connection with any use or reliance on the contents of the Materials except to the extent that such liability cannot be excluded by law.

Nothing in these terms shall exclude UH's liability to you for fraudulent misrepresentation by UH or its employees or for death or personal injury resulting from its negligence or that of its employees.

These terms shall be governed by and interpreted in accordance with the laws of England and Wales and you agree to submit to the exclusive jurisdiction of the courts of England and Wales in respect of any disputes which may arise in relation to the Materials.

Copyright, University of Hertfordshire Higher Education Corporation and its licensors. All rights reserved.

Contents

1	Introduction	4
2	Building the program	4
3	Running the program	5
4	Utilities	10
5	Input data	11
6	References	13
	Appendix 1	14
	Appendix 2	15

1. Introduction

The UM-MCIP meteorological pre-processor allows meteorological data produced by the UK Meteorological Office's UM model to be used directly by the CMAQ (Models-3) atmospheric transport model. As such, it replicates the functionality of the existing MCIP processor, supplied as part of the Models-3 system, which provides access to data produced by the US MM5 and WRF meteorological models.

The program is suitable for use with UM data archived at BADC and also with data produced by the ported version of the model (PUM). It may be used with both 'meso-scale' and 'global' data.

This document outlines the procedures for building and running the program and also provides information on the data requirements. A technical description of the pre-processor is available elsewhere⁽¹⁾.

2. Building the program

The pre-processor consists of a number of Fortran90 source files. These comprise various files created specifically to handle the UM data and also some routines from the existing MCIP processor developed by the EPA. The latter routines have been utilized to allow future changes to MCIP to be more readily incorporated into the pre-processor.

In addition to the pre-processor, a number of small utility programs is also supplied.

Both the netCDF and Models3-ioapi libraries are required to build the pre-processor. These are available from <http://www.unidata.ucar.edu/software/netcdf/> and <http://www.baronams.com/products/ioapi/> respectively. (The software has been developed and tested with the netCDF version 3.6.0 and ioapi version 3.0 libraries).

The software is supplied as source code in a single file 'ummcipv1.tar.gz'. This should be copied to a suitable source directory and then uncompressed:

```
tar -xzf ummcipv1.tar.gz
```

This will create files in the source directory as well the following directories:

```
mcip3
documents
utilities
run_script
```

Make-files are supplied to facilitate the building of both the main pre-processor and the utility programs. Before use, the make-files should be edited to select the relevant compiler and also to specify the associated compiler options. Entries are already supplied in the make-files for the Intel and Portland Group Fortran-90 compilers. In addition, the paths to the netCDF and Models3-ioapi libraries should also be specified in the make-file for the pre-processor.

The pre-processor may then be built from the source directory using:

```
make clean; make
```

This will create the executable ('ummcip.exe') in the source directory.

The utilities may be built in a similar fashion using the make-file in the /utilities directory.

(Note: Both of the above build operations create object files in the source directory. Before either make-file is used for the first time, ‘make clean’ should be used to delete these files).

Alternatively, all the programs may be built in a single pass using the ‘Build_all’ script in the source directory:

`./Build_all`

This builds all the programs and creates links to all the necessary files in the /exec directory.

3. Running the program

1. Specifying input parameters

All the necessary input to the pre-processor is provided in a single input file (‘namelist.ummcip’) which contains the information required to specify the model domain and user-selectable options. This could clearly be created and edited directly, but a simple script file, ‘run_ummcip’, is provided (in the /run_script directory) to facilitate the building of the input file and to control the subsequent execution of the processor.

This script file should be edited to reflect the requirements of the user. The script is designed to be self-explanatory; detailed information on the various parameters is, however, supplied below.

1.1 Identifiers and directories:

Run_Id	a brief character string to identify the current pre-processor ‘run’. The string is appended to the output file-names.
CoordName	a character string to identify the projection and co-ordinate system (defined in §3.1.8 below). The identifier and associated parameters are written into the GRIDDESC file. [maximum 16 characters]
GridName	a character string to identify the horizontal domain and grid (defined in §3.1.6 below) on which the output meteorological data are produced. The identifier and associated parameters are written into the GRIDDESC file. [maximum 16 characters]
MetDirIn	the path to the directory containing the UM meteorological files (see §3.1.2).
AnclDirIn	the path to the directory containing the UM ancillary files (see §3.1.3).
OpDirOut	the name of the directory to contain the output files.
WorkDir	the name of the working directory, where the input file (‘namelist.ummcip’) is to be created.
ExecDir	the name of the directory containing the ummcip executable.

1.2 Input meteorology files:

MetFilesIn	<p>a list of the UM meteorology files to be processed (normally relative to the MetDirIn directory). The files must be listed in chronological order.</p> <p>The list should be enclosed in parentheses, with the file-names space delimited and with lines separated by a back-slash (\).</p> <p>(The maximum number of UM files which may be included (25 by default) is defined by the parameter max_um in umfile_mod.F in the UMMCIP source code. This value should match that assigned to MaxUmFiles in the run script).</p>
------------	---

1.3 UM ancillary files:

The UM ancillary files contain 2-d, time-invariant data fields which supplement those provided in the main meteorological files. They must be provided on the same domain and at the same resolution as the meteorological files themselves. In addition they should be supplied in a big-endian, pp-format.

OrogIn	location of the file containing orographic data (i.e. terrain height [m]).
LandIn	location of the file containing land-use information. The file specifies the fractional coverage corresponding to each of 9 IGBP land-use categories in each grid cell.
ZOIn	location of the file containing surface-roughness (z_0) data [m] for each grid cell. If this field is left blank, the processor will look for the appropriate field amongst the data fields for the first time period in the UM meteorological files.

1.4 Run-time options:

These options specify how the boundary layer parameters, radiation fluxes and dry deposition velocities are determined within the processor.

The run-time options are identical to those utilized in the original MCIP program. In most cases Lpbl and Lrad should be set to 1, so that the boundary layer and surface radiation fields provided in the UM files are utilized.

Full details of the algorithms used to re-diagnose the boundary layer and radiation fields and to calculate the species-dependent dry-deposition velocities are available elsewhere⁽²⁾.

The dry deposition method specified should be chosen to be consistent with the chemical mechanism to be used in the subsequent CCTM calculations (i.e. it should be selected to produce deposition velocities for the appropriate chemical species).

Lpbl	determines how the boundary layer (pbl) parameters are determined: 1 - use pbl values from input meteorological fields. 2 - re-diagnose pbl values using pbl-similarity algorithm. 3 - re-diagnose pbl values using surface-similarity method.
Lrad	determines how radiation fields are specified: 1 - use radiation fields from input meteorological files. 2 - re-diagnose radiation fields using MCIPv1 algorithm.
Lddep	specifies the approach to use for calculating dry deposition velocities: 1 - use RADM dry deposition routine. 2 - use Models3 (Pleim) dry deposition method. 3 - use Models3 (Pleim) dry deposition method with Cl species. 2 - use Models3 (Pleim) dry deposition method with Cl and Hg species.

1.5 Start and end times for analysis:

Ump_start Ump_end	the times and dates of the first and last records to be contained in the output files. The dates and times should be provided in a character string of the form: YYYY-MO-DD-HH:MI:SS.SSSS. (i.e. 2006-05-28-01:00.00.0000) No temporal interpolation is carried out within the processor, so the chosen times should match those of records within the UM input files.
Intvl	the time interval [min] between successive output records. In most cases, this will just be the time interval between successive records within the UM files.

1.6 Specify the horizontal domain and grid on which data should be produced:

The processor produces data on a regular, rectangular grid located on a projected plane (rather than on the surface of the earth). The projection to be used is specified in §3.1.8 below. All locations are then referred to the (possibly false) origin defined by that projection. All dimensions here should be specified in km.

It should be noted that the processor produces data not only for a simple rectangular domain, but also on a boundary domain which encloses it. For current purposes this boundary domain is always 1 cell thick.

XOrig YOrig	define the location of the 'dot' point at the south-west corner of the total domain (including the boundary domain). These values should be in km., measured relative to the projection origin.
NxOut NyOut	define the dimensions of the total output domain (including the boundary domain) in terms of the number of 'dot' points in the x and y directions. (i.e. the number of grid cells is NxOut-1,NyOut-1).
DxOut DyOut	the dimensions of the output grid cells in the x and y directions [km]. Normally, DxOut = DyOut.

Clon	centre longitude (defines x-origin on projected plane)
<u>Mercator</u>	
Slat1	not used
Slat2	not used
Clat	latitude of true scale (defines y-origin of projected system)
Clon	longitude of central meridian (defines x-origin of projected system)
<u>Stereographic</u>	
Slat1	not used
Slat2	not used
Clat	latitude of true scale (defines y-origin of projected system)
Clon	the angle of rotation of the Y-axis relative to the Greenwich meridian, i.e., the longitude meridian which becomes the negative Y axis.
Feast	false easting, northing (km); re-locates origin so that projection origin is at (Feast,Fnorth).
Fnorth	

1.9 Define the vertical co-ordinate system:

As noted above, the program produces data on constant sigma-pressure levels on a hydrostatic, time-independent, vertical co-ordinate system. This co-ordinate system is discussed elsewhere^(1,3).

The pressure levels are defined with respect to a hydrostatic reference atmosphere. This is characterized by a surface pressure, P_{00} , at $z=0$ (normally chosen to be sea-level) and a corresponding reference temperature, T_{00} , along with a constant assumed adiabatic lapse rate.

It should be noted here that these parameters serve only to define a vertical co-ordinate system; unlike their use within models such as MM5, they otherwise have no particular physical significance. Thus, in principle, physically unrealistic values could be chosen for P_{00} , T_{00} and Alaps. In practice, though, typical values of $P_{00} = 10^5$ Pa, $T_{00} = 290^\circ$ K, Alaps = 50° K are normally appropriate.

P00	reference surface pressure [Pa].
T00	reference surface temperature [$^\circ$ K].
Ptop	pressure at model top [Pa]. This defines the vertical extent of the model domain.
Alaps	adiabatic lapse rate [$^\circ$ K].
Tsit	temperature of isothermal layer in upper part of domain [$^\circ$ K]; not currently used.
Zctm	surface height [m] to use when determining sigma levels which are consistent with the hybrid height levels utilized in the UM meteorological files.

1.10 Location of cell for diagnostic output:

The processor outputs diagnostic values at the specified array locations.

Lprt_col Lprt_row	(nx,ny) location in the output domain (excluding boundary cells) where diagnostic output is required. If either value is set to 0, the cell at the centre of the corresponding row or column will be assumed.
----------------------	---

1.11 Parameters controlling program behaviour:

The following parameters control the run-time behaviour of the program.

Mkgrid	determines whether the program will output the grid description file (GRIDDESC) and netCDF files (GRID_**) containing time-independent geographical and meteorological data. [Y,N]
<u>IOAPI Environment Variables</u> LOGFILE	name for the program log-file (optional). This will contain messages and error reports generated by the Models3 ioapi library.
IOAPI_CHECK_HEADERS	determines whether the ioapi headers are checked for validity [Y,N]. Note: setting this flag causes the program to fail within the ioapi library on some systems!
IOAPI_FILE_WRITE	determines whether message should be produced for each file-write from ioapi [Y,N].

2. Executing program

The processor may be executed simply by running the above script:

```
./run_ummcip
```

This will produce output in the directories specified in §3.1.1 above.

4. Utilities

A few simple utilities are supplied:

1. ll2xy and xy2ll

These programs calculate the projected (x,y) co-ordinates corresponding to supplied (lat,long) values (and vice-versa) for a specified map projection. The map projection should be defined in a text file 'prjparm.txt' (specimen provided). The projection parameters required here correspond exactly to those listed in §3.1.9 above.

2. read_ppheader

This program reads a UM meteorological file and lists various items of header information for each pp-formatted data block. This is useful if difficulties are encountered in accessing the data

through the processor. The meteorological filename may be supplied either as a command-line argument or via a user prompt.

5. Input data

1. UM meteorological data

The UM meteorological data should be supplied in one or more binary files. All the data corresponding to a particular time should be grouped together, with the data arranged in an increasing chronological order. It may be convenient to group the data together in separate files on a daily (or weekly) basis, but this is not essential.

The data should be in a binary, 32-bit pp-formatted form with big-endian byte-ordering. The above utility, 'read_ppheader', may be used to check that the data are correctly formatted. The X-window utility Xconv (available at <http://badc.nerc.ac.uk/help/software/xconv>) may also be used to examine and visualize the data.

The pp-format is a binary record-based format utilized by UK Met. Office as a standard format and is used in a number of BADC data archives. (Further information is available at <http://badc.nerc.ac.uk/help/formats/pp-format/index.html>). In this format a typical record would consist of the values of a particular field over a defined domain and at a specified hybrid height level. For present purposes, the required data input file may be considered as consisting of a series of such records.

The data fields required by UMMCIP are listed in Appendix 1.

1. Data from BADC archives:

Data fields calculated by the operational version of the Unified Model are archived in a packed form at BADC. Data are available for both the global and mesoscale domains.

Recent global data (post December 2005) are available on a domain consisting of 50 vertical layers (extending up to approximately 65km) on a 640 x 481 horizontal grid. The corresponding horizontal resolution is approximately 40km. Earlier data utilized 38 layers (extending up to about 40km) and 432 x 325 horizontal grid cells, yielding an horizontal resolution of ca. 60km. The temporal resolution of these data is 3h. [A utility will be provided to interpolate these data temporally to yield data at a 1h interval].

The mesoscale domain (<http://www.met.rdg.ac.uk/~uwern/UM/domains/index.html>) covers north-west Europe and utilizes 38 layers (extending up to 39km) and a 182 x 146 horizontal grid, yielding an horizontal resolution of approximately 12km. The interval between successive output data is 1h.

The operational model is run every 6 hours (i.e. at 0000, 0600, 1200, 1800). The records corresponding to the start of each forecast period are conventionally referred to as 'analysis' states and are denoted as 0ZT+0, 6ZT+0, 12ZT+0 etc. (In reality, each model run is initialized at T-2, but only the data from T+0 onwards are of interest for forecasting purposes and thus only such data are archived). The other stored records (i.e. 0ZT+1 ... 0ZT+6 etc.) are normally referred to as 'forecast' data.

The archived data may be downloaded from <ftp.badc.rl.ac.uk>. (The global data may be found at </badc/ukmo-um/data/global/ag> and the mesoscale data at </badc/ukmo-um/data/meso/lb>).

The data are archived on both model levels (i.e. hybrid height levels) and pressure levels for both the analysis and forecast periods. The 3-d data required by UMMCIP are those on the model levels. The only 'pressure level' data utilized are those corresponding to surface fields.

The procedures required to construct a suitable input file for the processor from the archived data sets are rather laborious. [A script to automate this process is currently being written]. An outline of the steps required to produce a suitable file manually is, however, included in Appendix 2.

The required binary pp-formatted data file should contain the data records ordered as 0Z (T+0 ... T+6), 6Z(T+0 ... T+6) etc. Clearly, 0Z(T+6) and 6Z(T+0) etc. refer to the same hourly period and data for either T+6 or T+0 may thus be utilized. (If both are included in the file, the latter item will just overwrite the former). The only exceptions here are the surface precipitation data. These archived fields contain cumulative precipitation values, from which must be deduced the incremental precipitations over each time interval. In this case, the T+0 data contain no directly useful information, but are required for initialization purposes. Both 0Z(T+6) and 6Z(T+0) etc. values must thus be supplied. Clearly, no meaningful precipitations may be derived for the first time period in the file and the precipitation fields are thus set to zero here.

2. Data produced by PUM:

Data produced by the Portable Unified Model may also be utilized. A suitable specimen PUM 'experiment' and instructions for re-formatting the data will be provided in the near future.

3. Ancillary files:

In addition to the time varying fields listed in Appendix 1, a number of additional time-invariant fields contained in 'ancillary' files are required (see §1.3 above). These contain (i) the orographical height data (ii) UM fractional land-use based on a sub-set of the IGBP categories data (vegetative fraction data).

The ancillary files at the required spatial resolution are included with the distribution package, but may also be downloaded from http://www.met.rdg.ac.uk/~uwern/um_ancil/.

Files containing surface roughness estimates for the global and mesoscale domains are also included with the distribution package. These are not standard 'ancillary' files but are included for convenience. If these are not available, the surface roughness fields in the BADC archives (stash code 00026), or the equivalent PUM fields, may be included in the input meteorology file.

6. References

1. Taylor, G.J., 2006, "Technical description of UM-MCIP", to be completed.
2. Byun, D.W., Pleim, J.E., Tang, R.T., Bourgeois, Al, 1999, "Meteorology-chemistry interface processor (MCIP) for Models-3 Community Multiscale Air Quality (CMAQ) modelling system", Models-3 Documentation, Chapter 12, EPA Document EPA/600/R-99/030
3. Jimmy Dudhia, Dave Gill, Kevin Manning, Wei Wang, Cindy Bruyere, 2005, "PSU/NCAR Mesoscale Modeling System Tutorial Class Notes and Users' Guide (*MM5 Modeling System Version 3*)" – Part 1: Introduction, available from <http://www.mmm.ucar.edu/mm5/documents/tutorial-v3-notes.html>

Appendix 1

UM variables required by processor

3D variables		Stash Code
u		2
v		3
w		150
P_θ	Pressure (theta levels)	408
P_ρ	Pressure (rho levels)	407
Q_v	Specific humidity	10
Q_w	Cloud liquid water	4205
Q_i	Cloud ice content	4206
T	Temperature	16004
2D Time Dependent Variables		
H_{pbl}	Boundary layer depth	25
z_0	Roughness length	26
H_{fx}	Sensible surface heat flux	3217
T_g	Surface temperature	24
P_{sf}	Surface pressure	409
$T_{1.5}$	Temperature at 1.5m	3236
$Q_v(1.5)$	Specific humidity at 1.5m	3237
σ_x	x component of surface shear stress	3219
σ_y	y component of surface shear stress	3220
	Net downward short wave radiation flux at surface	1201
	Total downward short wave radiation flux at surface	1235
	Net downward long wave radiation flux at surface	2201
	Downward long wave radiation flux at surface	2207
	Large scale rainfall rate	4201
	Convective rainfall rate	5201
	Snow amount over land	23

Appendix 2

Procedure for producing data files from data archived at BADC

The file required by UMMCIP is just a binary pp file containing the required global or mesoscale data in the order 0Z(0-6),6Z(0-6),12Z(0-6),18Z(0-6), where 0Z is the analysis hour, and (0-6) means the hourly forecast fields from that analysis.

The following outline manual procedure may be used to create suitable files. This example refers to mesoscale data; the global data may, however, be handled in an identical fashion. In this example it is assumed that all the available archived fields are to be included in the file. This is not necessary (only those fields listed in Appendix 1 are required). In practice though it is sometimes easier to adopt this approach.

- (1) download all the mesoscale data from the NWP archive (i.e. model and pressure levels, analysis and forecast fields). Thus for 27/01/06 retrieve all the files in:

```
/badc/ukmo-um/data/meso/lb/ap/2006/01/27/*.pp (analysis fields)
/badc/ukmo-um/data/meso/lb/ap/2006/01/27/*.pp " "
/badc/ukmo-um/data/meso/lb/fp/2006/01/27/*.pp (forecast fields)
/badc/ukmo-um/data/meso/lb/fm/2006/01/27/*.pp " "
```

- (2) concatenate all the analysis fields together for 0Z:

```
cat lba?2006012700* > 270106_00_00.pp
```

similarly, concatenate all the 0Z forecast fields together, for each forecast hour (1-6)

```
cat lbf?2006012700*_01.pp > 270106_00_01.pp
cat lbf?2006012700*_02.pp > 270106_00_02.pp
cat lbf?2006012700*_03.pp > 270106_00_03.pp
... etc
```

- (3) repeat the above process for 06Z, 12Z, etc., to give a set of files such as:

```
-rw-r--r-- 1 badc badcint 16196976 Feb 9 13:04 270106_18_06.pp
-rw-r--r-- 1 badc badcint 12314000 Feb 9 13:03 270106_18_05.pp
-rw-r--r-- 1 badc badcint 12339952 Feb 9 13:03 270106_18_04.pp
-rw-r--r-- 1 badc badcint 14197096 Feb 9 13:03 270106_18_03.pp
-rw-r--r-- 1 badc badcint 12403600 Feb 9 13:02 270106_18_02.pp
-rw-r--r-- 1 badc badcint 12428192 Feb 9 13:02 270106_18_01.pp
-rw-r--r-- 1 badc badcint 17342960 Feb 9 13:02 270106_18_00.pp
-rw-r--r-- 1 badc badcint 16306664 Feb 9 13:01 270106_12_06.pp
-rw-r--r-- 1 badc badcint 12386824 Feb 9 13:01 270106_12_05.pp
-rw-r--r-- 1 badc badcint 12402008 Feb 9 13:01 270106_12_04.pp
-rw-r--r-- 1 badc badcint 14245184 Feb 9 13:00 270106_12_03.pp
-rw-r--r-- 1 badc badcint 12418344 Feb 9 13:00 270106_12_02.pp
-rw-r--r-- 1 badc badcint 12435840 Feb 9 13:00 270106_12_01.pp
-rw-r--r-- 1 badc badcint 17360872 Feb 9 12:59 270106_12_00.pp
-rw-r--r-- 1 badc badcint 16315600 Feb 9 12:59 270106_06_06.pp
-rw-r--r-- 1 badc badcint 12399480 Feb 9 12:59 270106_06_05.pp
-rw-r--r-- 1 badc badcint 12421064 Feb 9 12:58 270106_06_04.pp
-rw-r--r-- 1 badc badcint 14288248 Feb 9 12:58 270106_06_03.pp
-rw-r--r-- 1 badc badcint 12463256 Feb 9 12:57 270106_06_02.pp
-rw-r--r-- 1 badc badcint 12476440 Feb 9 12:57 270106_06_01.pp
```

```
-rw-r--r-- 1 badc  badcint 17378640 Feb 9 12:57 270106_06_00.pp
-rw-r--r-- 1 badc  badcint 16303224 Feb 9 12:56 270106_00_06.pp
-rw-r--r-- 1 badc  badcint 12361936 Feb 9 12:55 270106_00_05.pp
-rw-r--r-- 1 badc  badcint 12359904 Feb 9 12:55 270106_00_04.pp
-rw-r--r-- 1 badc  badcint 14207208 Feb 9 12:55 270106_00_03.pp
-rw-r--r-- 1 badc  badcint 12351888 Feb 9 12:54 270106_00_02.pp
-rw-r--r-- 1 badc  badcint 12368000 Feb 9 12:54 270106_00_01.pp
-rw-r--r-- 1 badc  badcint 17295960 Feb 9 12:53 270106_00_00.pp
```

(4) combine these to give a single pp file:

```
cat 270106*.pp > 270106_all.pp.
```

(5) this file should then be byte-swapped prior to unpacking. Suitable byte-swapping utilities are ppflipper.pl and swapbytes, available from :

<http://badc.nerc.ac.uk/help/software/xconv/index.html#swapbytes> .

If using ppflipper, execute the command:

```
ppflipper.pl 270106_all.pp >270106_all_swap.pp
```

to give a file '270106_all_swap.pp' in little-endian form.

(6) this file should then be unpacked and renamed to give (say)

```
'270106_all_swap.pp.unpacked'
```

(7) finally, this file should be byte-swapped again to yield the required file in big-endian form.

[These notes are based on instructions provided by Kevin Marsh of BADC].