UNIFIED MODEL DOCUMENTATION PAPER NO. F5

Unified Model File Utilities

BY

D.M. Goddard

VERSION 13

30/12/99

MODEL VERSION 4.5

Numerical Weather Prediction
Meteorological Office
London Road
BRACKNELL
Berkshire
RG12 2SZ
United Kingdom

Modification Record		
Document version	**Author**	**Description ....**
2	A. Dickinson	(i) Description of how to access utilities clarified (ii) ieee utility extended to allow conversion of model files to IEEE 64-bit format
3	A. Dickinson	Environment variable UMDIR used to generalise file paths
4	A. Dickinson	(i) New utilities *convpp*, *cray2hds* and *hds2hp* described (ii) File transfers to workstations covered
5	A. Dickinson	New utility *bigend* described
6	D.M. Goddard	(1) New utilities *merge* and splitf described (ii) New directory structure used (iii) File transfer from ECMWF updated for new FR HP network
7	D.M. Goddard	*Convpp* extended to process ocean dumps
8	V. Blackman	New utility *qxumthin1* described
9	D.M. Goddard and I.Edmond	Explanation of how to use user-STASHmasters added. *fieldop* described.
10	D.M. Goddard I Edmond	Utilities modified for use on MPP systems Note on exec qxfieldcos added. fieldop/qxfieldmod/qxfieldcos modified for use on MPP systems
11	D.M. Goddard	*ieee* extended to convert IEEE 64-bit format to CRAY PVP 64-bit format
12	D.M. Goddard	Corrections to *ieee* to convert boundary files and compressed ocean dumps *pumf* extended to print out acobs and varobs files
13	I.Edmond	Run length encoded fields covered

1. Introduction

This paper documents utilities for manipulating Unified Model files which conform to the format described in Documentation Paper F3. In addition, two further utilities are also described which allow binary files to be transferred from the Met Office Cray to a HP workstation in such a way that they retain the same image as on the Cray. This facility is required for transferring UM files to workstations, but may be used for any file where the exact Cray structure needs to be retained on a workstation. For example, when Cray binary files need to be sent out via the email node onto internet.

The following UM file utilities are available:

NAME	SYSTEM	PURPOSE
cumf *	Cray/ws	compare UM files
pumf *	Cray/ws	print out the contents of a UM file
ieee *	Cray T3E	convert Cray UM files to IEEE format and vice versa
convpp*	Cray/ws	convert UM file into sequential PP format
bigend	Cray/ws	Toggle big endian and little endian data representation
merge*	Cray/ws	merge two UM files
splitf	ws	split a Fortran program into separate files
qxfieldmod	Cray	Thin/scale/convert/select/reject fields in a fieldsfile
fieldop*	Cray/ws	Simple mathematical functions
qxfieldcos	Cray	Converts dumps/fieldsfiles -> sequential IBM formatted pp files

The following file transfer utilities are available:

NAME	SYSTEM	PURPOSE
cray2hds	Cray	transfer UM file from Cray to HDS
hds2hp	ws	transfer UM file from HDS to HP workstation

2. UM file utilities

All of these utilities use small amounts of memory and so can be used interactively. UM files of any size or resolution can be processed.

The utilities (marked with *) are accessed via scripts which are stored on directory $UMDIR/vn4.5/utils. The environment variable UMDIR is automatically initialised at logon time. **Note that the full pathname must be specified to run these scripts.**

3. File transfers

3.1 From Cray PVP machines to Met Office Cray T3E

UM files archived from the C90 and UM files on any Cray PVP machine may be transferred to the Met Office Cray T3E for further processing or as input to running the UM system. The sequence of tasks required to convert a UM file to IEEE format and complete the transfer is described below.

Copy UM file to the Cray T3E using **ftp** or some other suitable file transfer.

On Cray T3E using UM Vn 4.5:

```
        MYPATH=$UMDIR/vn4.5/utils
        $MYPATH/ieee -64 umdump.pvp umdump.ieee
```

3.2 From Met Office Cray T3E to Cray PVP machines

UM files may be transferred from the Met Office Cray T3E to a CRAY PVP machine for further processing or as input to running the UM system on the CRAY PVP machine. The sequence of tasks required to convert a UM file to CRAY PVP format and complete the transfer is described below.

On Cray T3E using UM Vn 4.5:

```
        MYPATH=$UMDIR/vn4.5/utils
        $MYPATH/ieee -64c umdump.ieee umdump.pvp
```

Copy UM file to the Cray PVP machine using **ftp** or some other suitable file transfer.

3.3 From Met Office Cray T3E to a workstation

UM files may be transferred from the Met Office Cray T3E to workstation systems for further processing or as input to running the UM system on a workstation. The sequence of tasks required to convert a UM file to IEEE format and complete the transfer is described below.

On Cray using UM Vn 4.5:

```
        MYPATH=$UMDIR/vn4.5/utils
        $MYPATH/ieee -32 umdump umdump.ieee
        $MYPATH/cray2hds umdump.ieee ’mxx.file’
```

On HP server workstation:

```
        hds2hp mxx.file umdump
```

Alternatively, if the data set is not wanted on the IBM, The **ftp** utility may be used instead if **cray2hds** and **hds2hp.** The **ftp** utility may be initiated from either the Met Office CRAY T3E or a workstation.

3.4 From ECMWF Fujitsu to a workstation

UM files may similarly be transferred from the ECMWF Fujitsu directly to workstations or the T3E at the Met Office using the TCP/IP link. There is a firewall in between which controls access, meaning that access is only available from the following nodes:

NWP division: fr0110
Hadley centre: hc0020
Cray T3E : crt3e1.

To access this facility a secureID card is required. Once connected, the user can choose between **telnet**, **ftp** and **eccopy** for effecting the transfer. This facility is primarily for returning PP output to a workstation for analysis.

In the following example, a fieldsfile called *ffile* created on the ECMWF Fujitsu is transferred to an NWP Division HP workstation at the Met Office using eccopy and converted into a form suitable for input to the PP-Package. The file is returned to directory *$HOME/eccopy*

The following commands need to be entered:

On ECMWF Fujitsu:

```
eccopy --u hp_userid -h ip_address -f data.pf ffile
```

where **hp_userid** is the user's id on the destination workstation network
and **ip_address** is the ip address of the gateway server on the destination workstation network. For example in NWP the ip_address is 151.170.5.6 (fr0110)

On HP workstation:

```
convpp $HOME/eccopy/data.pf data.pp
```

Logical, integer and real data types are recognised and converted into their corresponding 32-bit or 64-bit IEEE forms. Fieldsfile with WGDOS packed or run length encoded fields may optionally have the packed/encoded fields unpacked by the utility.

Options:

One of the following must be specified

-32 Convert UM file on T3E to 32-bit IEEE format

-32e Convert UM file on T3E to 32-bit IEEE format and expand WGDOS or run length encoded fields

compressed fields

-64 Convert Cray PVP format UM file to 64-bit IEEE format

-64e Convert Cray PVP format UM file to 64-bit IEEE format and expand WGDOS compressed fields

-64c Convert 64-bit IEEE format UM file to Cray PVP format

-64ce Convert 64-bit IEEE format UM file to Cray PVP format and expand WGDOS compressed or run length encoded fields

4. User-STASHmaster files and the file utilities

Some UM files may contain user defined fields. Details of which are not stored in the standard STASHmaster files. If the user requires the name of the user defined field included in the file utility output, then a user-STASHmaster file must be supplied.
Otherwise the phrase "NON STANDARD OUTPUT" will be printed instead.
Prior to version 4.1, a user-STASHmaster file could be read in on unit 2. This is done by typing the following before executing the utility script.

```
UNIT02 = filename
EXPORT UNIT02
```

Where *filename* is the full pathname for the user-STASHmaster

Form version 4.1 onwards, there is an option in the scripts to attach user-STASHmaster files by supplying a namelist. The namelist is similar in format to the **USTNUM** namelist, which appears in the *RECONA*, *RECONO* and *STASHC* files in the job libraries produced by the UMUI.

The definition of the namelist here is as follows:

&STNUM	
N_USTASH	Number of user-STASHmaster files provided
NRECS_USTASH	Number of records in each user-STASHmaster file
USTSFILS	Full path name of user-STASHmaster files
&END	

Example: Consider two user-STASHmaster files, ***file1*** *and* ***file2*** *in our home directory.* ***File1*** *contains three records record1,record2 and record3.* ***File2*** *contains one record recorda.*

If both ***file1*** *and* ***file2*** *are required, the namelist is as follows;*

```
&STNUM
N_USTASH = 2, NRECS_USTASH =3,1,
USTSFILS= '$HOME/file1','$HOME/file2',
&END
```

If only ***file1*** *is required*

```
&STNUM
N_USTASH = 1, NRECS_USTASH =3,
USTSFILS= '$HOME/file1',
&END
```

If only ***file2*** *is required*

```
&STNUM
N_USTASH = 1, NRECS_USTASH =1,
USTSFILS= '$HOME/file2',
&END
```

Once the user-STASHmaster has been created, it is input to the utility following a **-uSTASH** flag. *eg pumf -uSTASH $HOME/stnamlst filename*, where filename is the operand. The default is no user-STASHmaster file

NAME

cumf - Compares Unified Model files

SYNOPSIS

cumf [-uSTASH stnamlst] *file1 file2*

DESCRIPTION

The **cumf** command carries out an intelligent compare of Unified Model files *file1* and *file2*. Comparison of atmosphere, ocean, ancillary, boundary and observation files as well as fieldsfiles is supported.

Three output files are written to your $TMPDIR directory.

cumf_summ.nnnnn	This contains a summary of differences
cumf_full.nnnnn	This contains all differing values for each header, the first 10 differing values are printed out for each field and the value and location of the maximum difference between the fields.
cumf.diff.nnnnn	This contains difference charts for each field.

***nnnnn** is the job number of the cumf job*

Options:

-uSTASH stnamlst

File **stnamlst** contains details of user-STASHmaster files to be used in the utility, *See section 4 for more details*.

NOTES

Well formed io was not part of the version 4.3 release but has been used operationally since July 1997. It was consolidated into version 4.4.

For well-formed files, users should use version 4.4 or later.

NAME

pumf - Prints out contents of a Unified Model file

SYNOPSIS

pumf [-o nprint] [-uSTASH stnamlst] *file1*

DESCRIPTION

The **pumf** command prints out a summary of the contents of the headers and each field stored on Unified Model file file1. The display of atmosphere, ocean, ancillary, boundary and observation files is supported as well as fieldsfiles.

Two output files are written to your $TMPDIR directory.
pumf_head.nnnnn This contains the contents of each header.
pumf_field.nnnnn This contains a summary for each data field.

***nnnnn** is the job number of the pumf job*

Options:

-o nprint

For observation files argument **nprint** contains the number of observations to be printed out.

-uSTASH stnamlst

File **stnamlst** contains details of user-STASHmaster files to be used in the utility, *See section 4 for more details.*

NOTES

(1) Well formed io was not part of the version 4.3 release but has been used operationally since July 1997. It was consolidated into version 4.4.
For well-formed files, users should use version 4.4 or later.

(2) Printouts of acobs and varobs files are only available at vn4.5 or later.

NAME

ieee - Converts a Cray Unified Model file into IEEE format

SYNOPSIS

ieee -32|-64[c][e] [-uSTASH stnamlst] *file1 file2*

DESCRIPTION

The **ieee** utility has three functions :-

1) To convert *file1*, a Cray format Unified Model file from the C90 or any PVP machine into 64-bit IEEE format and stores the results in *file2*.

2) To convert *file1*, a 64-bit IEEE format Unified Model file e.g. a CRAY T3E Unified model file into 64-bit CRAY PVP format and stores the results in *file2*.

3) To convert *file1*, a Cray Unified Model file already in 64-bit IEEE format e.g. a Cray T3E Unified Model file into 32-bit IEEE format and stores the results in *file2*.

Logical, integer and real data types are recognised and converted into their corresponding 32-bit or 64-bit IEEE forms. Fieldsfile with WGDOS or run length encoded fields may optionally have the packed fields unpacked by the utility.

Options:

One of the following must be specified

-32	Convert UM file on T3E to 32-bit IEEE format
-32e	Convert UM file on T3E to 32-bit IEEE format and expand WGDOS or run length encoded fields compressed fields
-64	Convert Cray PVP format UM file to 64-bit IEEE format
-64e	Convert Cray PVP format UM file to 64-bit IEEE format and expand WGDOS compressed or run length encoded fields
-64c	Convert 64-bit IEEE format UM file to Cray PVP format
-64ce	Convert 64-bit IEEE format UM file to Cray PVP format and expand WGDOS compressed or run length encoded fields

The following is optional

-uSTASH stnamlst

File **stnamlst** contains details of user-STASHmaster files to be used in the utility, *See section 4 for more details.*

NOTES

(1) The c and e flags are not case sensitive.

(2) Well-formed boundary files require version 4.5 or later.

(3) Conversion of Old (pre-version 4.2) compressed ocean dumps to 64-bit IEEE format requires vn4.5 or later.

NAME

cray2hds - Transfers a binary file from the Cray to the HDS

SYNOPSIS

cray2hds *crayfile ’ibmfile’*

DESCRIPTION

The **cray2hds** command transfers a binary file to the HDS in a format suitable for onward copying to a HP workstation using the utility **hds2hp**. The target file is copied to the generic disk CRDISK. The HDS file name should be specified in single quotes in the normal way. If the target file already exists, then **cray2hds** will fail.

Options:

None.

NAME

hds2hp - Transfers a binary file from the HDS to a HP workstation

SYNOPSIS

hds2hp *ibmfile hpfile*

DESCRIPTION

The **hds2hp** utility copies a binary file from the HDS system to a unix workstation, removing blocking bytes in the process. This utility is required if **cray2hds** has been used on the Met Office Cray to copy a file to the HDS.

Options:

None.

NOTES

This utility can only be run on a server node supporting DECNET access to the HDS.

NAME

merge - Combines two Unified Model files

SYNOPSIS

merge *n*|-*t file1 file2 file3*

DESCRIPTION

The **merge** command combines two unified model files *file1* and *file2* writing the output to *file3.* This utility is primarily designed to merge boundary files, but has been extended to support atmosphere, ocean and ancillary files as well.

Options:

-*t* The files are automatically merged at the point where they merge temporally. This option should only be used for timeseries. It is also the option to use for boundary files.

n >=0 The first *n* records are taken from *file1,* then the whole of *file2* appended.

-uSTASH stnamlst

File **stnamlst** contains details of user-STASHmaster files to be used in the utility, *See section 4 for more details.*

NOTES

See Unified model documentation paper F51 for further details.

NAME

bigend - Converts data between big endian and little endian formats

SYNOPSIS

bigend -*32*|-*64 file1 file2*

DESCRIPTION

The **bigend** utility copies *file1* to *file2* reversing the order of the bytes in each word. This utility allows unformatted data in little endian format to be converted to big endian format or vice versa.

Options:

One of the following must be specified

-32 Data held in 32-bit IEEE format

-64 Data held in 64-bit IEEE format

NAME

splitf - Splits a Fortran program into separate files.

SYNOPSIS

```
splitf  fortran_file
```

DESCRIPTION

The **splitf** command splits a Fortran program into separate files, such that each file contains just one subroutine, function, main program or block data segment. Each file takes the name of the segment it contains, and is suffixed by a .f .

Options : none

NAME

qxfieldmod - Thin/scale/convert/select/reject fields in a fieldsfile

SYNOPSIS

```
export UM_SECTOR_SIZE=2048
export UNIT07="Diagnostic filename"
export UNIT10="Input fieldsfile name"
export UNIT11="Output fieldsfile name"
export UNIT12="Orography filename"
$UMDIR/vn4.5/exec/qxfieldmod < " Namelist filename"
```

DESCRIPTION

qxfieldmod operates on fieldsfiles and can :-

(a) Select a subset of fields.
(b) Reject a subset of fields.
(c) Scale fields.
(d) Replace 10m winds by the level 1 wind scaled using wind_10m_scale (specified in the namelist).
(e) Thin fields. Fields are thinned by extracting every nth point where n is specified for both X and Y directions in the namelist input.
(f) Output fields can be unpacked, WGDOS packed, Run length encoded, CRAY32 or GRIB format (GRIB format has a PP header added).

NOTE

(1) Thinning does not interpolate.
(2) UM sector size was increased from 512 to 2058 at version 4.5 and operationally on 15th April 1998

Full details of this utility can be found in UNIFIED MODEL DOCUMENTATION PAPER No. F52, FIELDMOD - Modifying Unified Model Output Fields.

NAME

fieldop - Performs simple mathematical functions on fieldsfiles/dumps.

SYNOPSIS

$UMDIR/vn4.5/utils/fieldop [[-asm] [-d integer]] [-n string] [-uSTASH "filename"] *file1* [*file2*] *outfile*

DESCRIPTION

The **fieldop** command reads in two model dumps or direct access fieldsfiles with unpacked or packed (wgdos,grib,run length encoded,cray32 bits) data and write out to a new file, the difference, sum or product of the data values. Alternatively if a single dataset is read the data may be divided by an integer.

Important Note: A new dumpfile format was introduced at vn4.4 to make the addressing "well-formed" for Cray IO . For well-formed files, users should use version 4.4 or later.

Options:

-a addition of equivalent fields in *file1* and *file2*. The new data fields are written out to *outfile.*

-s subtraction of each data field in *file2* from the equivalent field in file1 to produce a new field written out to *outfile.*

-m multiplication of the data fields in *file1* and *file2* with the product written out to *outfile.*

-d divides the data in file1 by an integer number. New data values written out to *outfile.*

-t <string> character string protected by single or double quotes contains a list of full stashcodes, separated by spaces, indicating which fields **to** operate upon. The remaining fields are transferred directly from *file1* to *outfile.*

-n <string> character string protected by single or double quotes contains a list of full stashcodes, separated by spaces, indicating which fields **not to** operate upon. These fields are transferred directly from *file1* to *outfile.* The remaining fields are operated upon.

-l <string> character string protected by single or double quotes contains a list of levels (LBLEV from **UMDP F3**), separated by spaces, indicating which levels **to** operate upon. The remaining levels are transferred directly from *file1* to *outfile.*

-T If set, data/validity times are taken from *file2* instead of *file1*.

-uSTASH <string> pathname of file containing UM namelist USTSNUM specifying user STASHmaster details. e.g.

```
&USTSNUM
N_USTASH=1 ,   NRECS_USTASH= 1 ,
USTSFILS="/u/m11/user3/t11ie/user_STASHmaster"
&END
```

Environment variables:

A temporary working directory may be specified by initialising and exporting the environment variable TMPDIR. Otherwise /tmp is used to store intermediate files.

Note:

The program searches through each file and performs the specified arithmetic operation on fields with same stash code, level code and length of data record. *File1* and *file2* may therefore have different numbers of fields with unique field types and equivalent field types may be stored at different addresses.

The output file is a copy of *file1* with the fields common to both input files, being operated upon. Those which are unique to *file1* are unchanged.

A change is made to element 15 of the fixed header to record the type of the last arithmetic operation:

addition	- fixhd(15) = 100
subtraction	- fixhd(15) = 200
multiplication	- fixhd(15) = 300
division by integer	- fixhd(15) = 400

Examples:

The following example differences the data in files *file1* and *file2*, writing the differenced fields to *outfile*. Fields, "U component of wind" and "x-comp of surf and bl wind stress" are written directly to *outfile* from *file1*.

```
fieldop -s -n "2 3219" file1 file2 outfile
```

To divide the data in *file1* by an integer e.g. 10 except those fields with stashcodes 2 and 3219 use:

```
fieldop -d 10 -n "2 3219" file1 outfile
```

NAME

qxfieldcos - To read a model dump or direct access fieldsfile and convert it to a sequential pp file ready for transfer to IBM.

SYNOPSIS

The calling script contains the unit declarations

export UNIT07="Diagnostic filename"
export UNIT10="Input dump/fieldsfile name"
export UNIT11="Output dump/fieldsfile name"

Run executable

$UMDIR/vn4.5/exec/qxfieldcos < " Namelist_filename"

Example of namelist file, Namelist_filename

&PACK UNPACK=.FALSE., FORMAT_OUT=’IBM ’ /
&TYPE OPER=.TRUE. /

DESCRIPTION

The fieldcos executable requires 3 namelist variables

UNPACK	(logical)
FORMAT_OUT	(character string)
OPER	(logical)

read in from namelists UNPACK and TYPE as in the above example.

Variable UNPACK applies to fieldsfiles only (UNPACK is set .true.for dumpfiles) and unpacks the 32bit/wgdos/grib packed or run length encoded fields before converting this unpacked data into the format indicated by namelist variable FORMAT_OUT.

Variable FORMAT_OUT is set to ’IBM ’ ...is the default type and converts fieldfiles/dumps into a sequential pp file in IBM number format. The pp file may be copied over to the IBM using putibm
e.g. on the T3E

```
UI=ruser=t11ie,racct=’"(m11,ie,fc15z)"’
export IBMFILE=ms15.ie.ppfile
vbstext=’text="dcb=(lrecl=x,blksize=10000,recfm=vbs)’
vbstext=$vbstext,’space=(cyl,(100,50),rlse)’
vbstext=$vbstext,’storclas=scdatprk,mgmtclas=mcsnc4"’
putibm /tmp/t11ie/pp_ibm_fmt  $IBMFILE,stat=new,df=tb,$UI,$vbstext
```

Variable FORMAT_OUT is set to ’IEEE ’ ....conversion of 64 bit IEEE fieldsfile/dump into a 32 bit IEEE sequential file may be achieved. However, as the BCW’s inserted by the unformatted FORTRAN WRITE on the T3E differ from those used by the HP unformatted FORTRAN READ/WRITE, the file cannot be copied onto the HP’s and used with PP Package/WAVE etc. To produce a pp file on the HP which is usable with WAVE etc follow the procedure:

```
T3E
 ieee -32 dumpname outfile   # produce 32 bit dumpfile
 rcp outfile frxx@fr0???:~frxx
```

HP
convpp outfile ppfile   # converts dumpfile into pp file format

Variable FORMAT_OUT is set to 'VAX   ' .... the C90 conversion utility CRAY2VAX is not supported on the T3E and therefore the conversion from 64bit IEEE -> VAX numbers cannot be done.

Variable FORMAT_OUT is set to 'GRIB   ' ..... the option to convert fieldsfile/dump to a pure GRIB file ready to be passed to IBM or HP has not been tested yet.

TYPE Namelist variable OPER.
For an accumulation or time mean (ie lbtim.ne.0), the start and end time are in a different order to the data and verification time for a snap shot type field. The result is that the PP Package fails with accum/time mean fields for operational fieldsfiles. To overcome this anomaly, if variable OPER is set .true., fieldcos will recalculate the data time from the end time and fcst period and reset pp header elements 1-12.

Important Note: A new dumpfile format was introduced at vn4.4 to make the addressing "well-formed" for Cray IO. Fieldcos vn4.4 or later should be used with "well-formed" dumpfiles.

```
       Met Office Cray                    IBM                         Workstation

UM DUMP --------------> ieee ---> cray2hds ---> hds2hp ------------------> UM DUMP
   |                                                                          |
   v                                                                          v

  UM <--   ANCILLARY --> ieee ---> cray2hds ---> hds2hp ---> ANCILLARY -->   UM
System        FILE                                                       System
   |                                                                          |
   |                                                                          v
   |
   |                                                                    FIELDSFILE
   |                                                                          |
   v                                                                          v

FIELDSFILE -----------> ieee ---> cray2hds ---> hds2hp -----------------> convpp
   |                                                                          |
   v                                                                          v

convpp                                                                     PPDATA
   |
   v

PPDATA
```

Figure 1: Overview of supported file transfers. Uppercase names represent files. Italics represent file utilities.