UNIFIED MODEL DOCUMENTATION PAPER NO. F51

Unified Model Merge Utility

BY

D.M. Goddard

VERSION 2

01/12/98

MODEL VERSION 4.5

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

Modification Record		
Document version	**Author**	**Description...................**
2	D.M. Goddard	Updated to UM version 4.5 and T3E.

1. Introduction

This utility allows the user to combine two unified model files into one file. The point at which the second file overwrites the first file is defined in a namelist. For t*imeseries*[1] there is an additional option to merge the files where they overlap temporally, *ie file2 overwrites file1 from the point where their validity times first coincide.*

This utility was written primarily for merging boundary files for input into limited area models using the time series facility. The utility has been extended to cope with dump and ancillary files. Fieldsfiles and observation files will follow at the next release.

2. How to use utility

The 'mergeum' utility uses a small amount of memory and so can be used interactively. UM files of any size or resolution can be processed.

NB On T3E ,the mergeum utility must be run at vn4.5 or later

The 'mergeum' utility can be accessed via a script which is stored on directory $UMDIR/vn4.5/utils. The environment variable UMDIR is automatically initialised at logon time. The script expects an **option**, followed by three filenames. The option can be either **-t** or a positive number. The positive number is used when the user wants to merge the files after a **fixed** number of records. **-t** can only be used for time series.It is used for merging files where they temporally overlap. *For example, Consider the two operational mesoscale model boundary files. One containing data over the period from 18Z 07/10/93 until 03Z 08/10/93 and the other from 21Z 07/10/93 until 00Z 10/10/93. These overlap between 21Z 07/10/93 and 03.08/93. The merged dataset would contain data from file1 between 18Z and 20Z inclusive on 07/10/93,*
and then to whole of file2. The filenames are the two input files followed by the output file.

NB T**he order of the parameters is important !**

[1] ***timeseries*** are files with data fields at consecutive times, *i.e.* one time per header as opposed to STASH timeseries where the header covers data with different times.

To execute the merge utility using the timeseries option enter

```
$UMDIR/vn4.5/utils/qxmergeum -t file1 file2 file3
```

To append **file 2** to **file 1** after a specified number of records simply replace -t with a positive number.

For example to append **file 2** after 15 records of **file 1** enter

```
$UMDIR/vn4.5/utils/mergeum 15 file1 file2 file3
```

3. How the utility works

The utility consists of a script, a namelist and an executable. The script **$UMDIR/vn4.5/utils/mergeum** converts the option parameter into the namelist, assigns fortran unit numbers and executes the executable $UMDIR/vn4.5/exec/qxmerge. If the option is a positive number it is passed directly into the namelist. If the option is **-t** then the namelist parameter is set to **-1**. If the option is anything else then the script will abort.

The source code for the main routines can be found in deck **MERGE1A**. This opens and reads the namelist and the two input files, together with **PPXREF.** The headers of both files are checked for consistency, aborting the program if two inconsistent files (files of different resolution or type) have been input. Messages are written stating any differences between the headers. These messages by themselves do not constitute an error. There will be differences unless you merge a file with it's identical twin.

If **-t** has been set (*ie NRECF1=-1*), the lookup table of **file1** is searched for the first record of **file2**, a record is kept of how many records in **file1** that have been checked. If a match has been found a message is printed and a variable **IDIFF** is set to the number of records in **file1** bypassed to reach the matching point. If no match is found the program aborts with an error message. If **-t** has not been set, **IDIFF** is set equal to the positive integer in the namelist.

The output file is now created taking the first **IDIFF** fields from **file1** then appending the whole of **file2**. Certain header and lookup elements are then adjusted to take account of the altered position of all **file2** fields. The output file is then written to **file3**.