
Unified Model Documentation Paper F3

Format of Atmospheric Dumps, Fieldsfiles, Ancillary Data Sets, Boundary Data Sets, Observation Files and Post Processed (PP) files for the Unified Forecast / Climate Model, Met Office Data Analysis Applications and Met Office Storage Systems.

Date of Issue: January 2013
Unified Model Version: 8.3

Met Office
FitzRoy Road
Exeter
Devon EX1 3PB
United Kingdom

© Crown Copyright 2013

This document has not been published. Permission to quote from it must be obtained from the Intellectual Property Rights manager at the above address.

Contents

1	Introduction	4
1.1	UM system fieldsfiles	4
1.2	Post-Processing (pp) files	5
1.3	Related Formats	5
1.4	Additions and Changes to the file formats	5
1.5	History and layout of this document	6
2	List of data set components	7
2.1	FIXED_LENGTH_HEADER	7
2.2	INTEGER_CONSTANTS	7
2.3	REAL_CONSTANTS	7
2.4	LEVEL_DEPENDENT_CONSTANTS	7
2.5	ROW_DEPENDENT_CONSTANTS	8
2.6	COLUMN_DEPENDENT_CONSTANTS	8
2.7	FIELDS_OF_CONSTANTS	8
2.8	EXTRA_CONSTANTS	8
2.9	TEMP_HISTORYFILE	8
2.10	COMPRESS_FIELD_INDICES	8
2.11	LOOKUP	8
2.12	DATA	8
3	Details of the data set contents	9
3.1	FIXED_LENGTH_HEADER	9
3.2	INTEGER_CONSTANTS	13
3.3	REAL_CONSTANTS	15
3.4	LEVEL_DEPENDENT_CONSTANTS	17
3.5	ROW_DEPENDENT_CONSTANTS	18
3.6	COLUMN_DEPENDENT_CONSTANTS	19
3.7	FIELDS_OF_CONSTANTS	19

3.8	EXTRA_CONSTANTS	19
3.9	TEMP_HISTORYFILE	19
3.10	COMPRESSED_FIELD_INDEX	20
4	LOOKUP table and PP headers	21
4.1	DATA	21
4.2	The LOOKUP table: PP header content	21
5	Atmosphere Observation Files	32
A	Model level Definition	33
B	WGDOS packing in the UM	34
B.1	COMPRESSION AND EXPANSION ROUTINES	34
B.2	Format of Compressed data	35
C	FORMAT OF DATA RECORD FOR PP-FIELDS	37
C.1	Order of grid points	37
C.2	Extra data	38
D	Towards Portable PP files	39
D.1	Records	39
D.2	Data types	39

Chapter 1

Introduction

This note describes the format of the data files used within the Unified Model forecast/climate model system, the scientific desktop analysis environment, and the MASS archive. The note describes two main kinds of file: the UM fieldsfile and the post processed or pp file. Although these are distinct formats they share a 64 word metadata structure (the lookup header in a fieldsfile and the pp header, or field descriptor record [FDR], in the pp format) and can both use WGDOS packing. This document describes both the common aspects and the differences.

1.1 UM system fieldsfiles

In designing the UM fieldsfile format the aim has been to avoid any features which would make the format inflexible with regard to changes in number of levels, number of field types or even the order in which fields are held, as well as being adaptable for different applications to which the UM might eventually be put. This has led to the construction of a file format which consists of a primary header record with pointers to a series of secondary header records and also records pointing to and describing the data areas as well as the data areas themselves.

Each field has an associated 64 (128 for observation files) word header. The 64 word version of the header is the common structure between fieldsfiles and pp files. The header describes the field, and helps determine how the values in the field are located in space and time. The content of this header is described in detail in section 4. In a UM fieldsfile the 64 word header is contained within a lookup table. The location of any one particular field is obtained by a search on the appropriate header element(s) (most often that giving the STASH code or PP field code) of the lookup table. Full details of pp field codes are available online at http://www-hc/umdoc/pp_package_ibm_docs/@fcodes, while STASH codes are found in the STASH-master file(s) associated with a UM run. The UM system STASHmaster files for different versions of the UM can also be viewed at <http://www-nwp/umdoc/pages/stashtech.html>.

Note that the adoption of the fieldsfile type lookup table implies a mixed integer/real array which has conventionally been decoded by EQUIVALENCING. This non standard convention is allowable within CONTROL routines only.

In the basic version of the model there is an assumption when searching the lookup that the levels for a given variable are contiguous. This can still be assumed but it is a programming restriction not a data set restriction. All authors of code which generate dumps, fieldsfiles or ancillary data sets should obey the restriction. However, one cannot assume that the first occurrence of a particular field is the primary model data field (it might be a time series, zonal mean diagnostic etc), so a more comprehensive check on several LOOKUP words is required to uniquely identify a field. In practice, the primary model data fields will occur first in the dumps before other fields sharing the same primary PP code. Also in practice, there will be a well defined order in which fields are held within the dump.

It would be too restrictive to lay down such an order precisely at this stage, but if we accept that prognostic variables will precede diagnostic variables which will precede time series then it will be a much simpler process to compare related dumps. fieldsfiles will not have such a well defined order of fields. Ancillary data sets contain much smaller amounts of information with a conventional order of fields for each type of data set.

1.2 Post-Processing (pp) files

The PP file format was designed for use with data analysis and data archiving. Initially there was a suite of FORTRAN routines called the pp package that use pp files as input and output. These FORTRAN routines were largely superseded by the UKMO PV Wave (now TIDL) routines.

The pp format is also used in the MASS storage system (MOUSE and now MOOSE). The pp format is essentially sequential, rather than direct access. The sequential pp file can be read relatively efficiently from tape in the MASS storage system.

A pp file is made up of a number of pp fields. Each pp field contains a pp header and is followed by a data field (i.e. H1 D1 H2 D2 H3 D3 ... HN DN). The content of the pp header is largely the same as that of the lookup table in the fieldsfile.

Although not enforced, there has been an implied rule about the use of pp fields, which derives its philosophy from GRIB. This is that each field in a pp file can be used independently of any other pp fields. A pp file can be split on fields or two pp files can be concatenated together without breaking the basic 'usability' of the data. This means that for complex grids each field should carry its coordinates with it. It also means that compression methods such as land mask compression are not supported by pp (although they are by UM fields files).

In practice, however, as the complexity of fields and their coordinates has increased the ideal of independent fields has been relaxed in some limited cases. For instance although pp fields carry enough information to locate each data point horizontally in space (at least for fixed resolution grids), they do not always carry enough information to locate each data point in the vertical. For example any field on hybrid height levels needs an orography field to locate the height of the field.

1.3 Related Formats

There are other formats used in and around the Met Office that share a similar pedigree to fieldsfiles and pp files. These include GPCS (IBM mainframe) fieldsfiles, HORACE fieldsfiles, NIMROD files, and FRASIA pp files. The details of these formats is not currently documented here. The GPCS fieldsfiles share the same 64 word lookup/pp header as UM fieldsfiles and pp files so this documentation should describe the use of this 64 word header in GPCS fieldsfiles.

Very old data may be stored in pp fields with a 42 word header rather than a 64 word header. These are not documented here.

1.4 Additions and Changes to the file formats

Occasionally model developments or new data products are made that can not be supported by the current fieldsfile and pp formats. When considering how to accommodate any new data products into the file format please consult widely. Any changes you make will have potential downstream impact in other applications including MASS archiving, TIDL and IRIS data analysis and visualisation, and the operational and product suites.

1.5 History and layout of this document

The fieldsfile format and pp file format used to be documented independently. Due to the formats sharing much in common this lead to a document maintenance problem with the definition of the 64 word lookup/pp header becoming inconsistent in the two documents. To avoid this in future the two documents have been merged. The choice was made to keep this F3 document which described the UM fieldsfile format as the main source of information, and absorb the documentation on the pp header into it.

The main structure of the original F3 document has been preserved. This means that sections 2 and 3 continue to describe features only relevant to the fieldsfile format. Section 4 describes the 64 word lookup/pp header which is common (with some small exceptions noted in that section) to both fieldsfiles and pp files. Section 5 describes some extensions to fieldsfiles for observations files.

Chapter 2

List of data set components

The full list of components of the data set format is given below. This looks a formidable list but the option should be available of having empty records if they are not required. Note that an empty record is defined as having a negative start address. The dimensioning of that missing record will be 1 to avoid problems with zero dimension arrays, but there is no requirement to read that single word.

2.1 FIXED_LENGTH_HEADER

A primary record of 256 integer words whose main purposes are to locate the secondary records and identify the type of dump. There should be nothing in this header which is specific to any one application. All the remaining records have a variable number of words which can be configured differently for different applications.

2.2 INTEGER_CONSTANTS

An integer array containing model variables which further identify the data, or model variables which need to be passed between runs or tunable model variables which although fixed might usefully be held in the dump to identify the run for future reference. This record should not contain any arrays.

2.3 REAL_CONSTANTS

As [2.2](#) above but for real variables.

2.4 LEVEL_DEPENDENT_CONSTANTS

Within the code this is anticipated to be a two dimensional array containing variables for which values are required at every level. It should not be assumed that the first dimension of the array is the number of levels. Since we need the flexibility of holding constants which might apply to different sets of levels (eg all levels, wet levels, soil levels etc). Real or integer values may be specified.

2.5 ROW_DEPENDENT_CONSTANTS

As 2.4 above but for variables which are required for each row

2.6 COLUMN_DEPENDENT_CONSTANTS

As 2.4 above but for variables which are required for each column (ie point along a row)

2.7 FIELDS_OF_CONSTANTS

As 2.4 above but for variables which are required for each grid point in the field

2.8 EXTRA_CONSTANTS

An unstructured one dimensional array containing other essential 'constants' that do not conveniently fit into 2.2 to 2.7 above.

2.9 TEMP_HISTORYFILE

No longer used.

2.10 COMPRESS_FIELD_INDICES

No longer used.

2.11 LOOKUP

A 64 or 128 word header table (usually one header per 2D field) describing and pointing to the model's real and integer data fields including timeseries. This is essentially an adaptation of the pp headers. Full details are given in section 4.2.

2.12 DATA

Real, integer or logical data fields or observations including timeseries.

Chapter 3

Details of the data set contents

3.1 FIXED_LENGTH_HEADER

1. Data Set Format Version Number.

15 Data set format : pre-vn3.1.

- Since vn3.1, the use of Word 1 in the fixed header has been dis-continued and is set to -32768 (IMDI). Any format change is now connected with a new version of the UM.

20 Data set format for MASS storage purposes : vn5.2 onwards.

- The use of Word 1 has been re-instated. It will cater for dataset changes required for MASS storage purposes. When used, the initial value will be 20 which will be incremented by one for each dataset change. Any dataset changes connected with new UM releases should continue to be done through the use of Word 12 of the fixed header.

2. Indicator for Sub-Model :

- 1 Atmosphere
- 2 Ocean Removed at vn7.0
- 4 Wave vn4.1 onwards / Removed at vn7.0

3. Indicator for vertical coordinate type :

- 1 hybrid
- 2 sigma
- 3 pressure
- 4 depth
- 5 Charney-Phillips on radius levels needed for dynamics grid, used by variational assimilation tangent linear dumps. vn3.4 onwards
- 6 Wave model direction and frequency pseudo-levels. vn4.1 Onwards

4. Indicator for horizontal grid type :

- 0 Global
- 1 Northern Hemisphere
- 2 Southern Hemisphere
- 3 LAM(No Wrap)
- 4 LAM(Wrap)

- For rotated grids : Add 100 to grid type.
5. Indicator for dataset type :
 - 1 Instantaneous dump
 - 2 Mean dump
 - 3 FIELDSfile
 - 4 Ancillary dataset
 - 5 Boundary dataset : Not available with 5.0 and 5.1
 - 6 AC observation file
 - 7 VAR observation file
 - 8 Cx file (model columns at observation location)
 - 9 Covariance File
 - 10 OPS Obstore File
 - 11 GPCS FIELDSfile
 6. Run identifier :
 - 0 Undefined
 - 1 Main
 - 2 Update
 - or programmer code (>100)
 7. Experiment Number e.g. identify version of model (1) or as for climate model?
 8. Indicator for calendar :
 - 1 standard Gregorian calendar
 - 2 360 day calendar
 9. Indicator for grid staggering e.g.(1-5 for Arakawa a-e grids)
 - 2 Arakawa B grid. Used in UM pre vn 5.0.
 - 3 Arakawa C grid. From vn 5.0
 10. For ancillary files, indicator for times at which data is provided:
 - 0 Single time
 - 1 Time series
 - 2 Periodic time series
 - For IAU increment files, the TimeID provided by the Variational Analysis scheme (see UMDP31 for details):
 - 0 Standard analysis increment
 - 3 Constant error tendency
 11. Projection Number
 12. Model version number x 100 + release number. E.g. vn7.7 is stored as 707
 14. Type of Observation File 1/2/3/4 Atmosphere/Ocean/SST/Wave
 15. Type of arithmetic operation last done in fieldop. From vn 4.3

- 100 Addition
- 200 Subtraction
- 300 Multiplication
- 400 Division by Integer

<ul style="list-style-type: none"> 21. Year 22. Month 23. Day of Month 24. Hour 25. Minute 26. Second 27. Day number 	}	<p>Dumps: Initial data time; Observation Files: Validity Time; Ancillary File, Boundary dataset or FIELDSDFILE: First Validity time</p>
---	---	--

<ul style="list-style-type: none"> 28. Year 29. Month 30. Day of Month 31. Hour 32. Minute 33. Second 34. Day number 	}	<p>Dumps: Validity time of instantaneous fields (Start time of mean fields). MODEL BASIS TIME Observation file: Validity Time Ancillary Files, Boundary datasets, FIELDSDFILE: Last validity time.</p>
---	---	--

<ul style="list-style-type: none"> 35. Year 36. Month 37. Day of Month 38. Hour 39. Minute 40. Second 41. Day number 	}	<p>Dumps and observation files: Wall clock time file generated. Ancillary Files and Boundary datasets: Interval between validity times of data. Set to zero if such data present for one time only.</p>
---	---	---

- 100 start of INTEGER_CONSTANTS
- 101 dimension of integer constants array
- 105 start of REAL_CONSTANTS
- 106 dimension of real constants array
- 110 start of LEVEL_DEPENDENT_CONSTANTS
- 111 first dimension of level dep constants array
- 112 second dimension of same
- 115 start of ROW_DEPENDENT_CONSTANTS (IMDI if RDC not used)
- 116 first dimension of row dep constants array (IMDI or 1 if RDC not used)
- 117 second dimension of same (IMDI or 1 if RDC not used)
- 120 start of COLUMN_DEPENDENT_CONSTANTS (IMDI if CDC not used)
- 121 first dimension of col dep constants array (IMDI or 1 if CDC not used)

- 122 second dimension of same (IMDI or 1 if CDC not used)
- 125 start of FIELDS_OF_CONSTANTS
- 126 first dimension of fields of constants array
- 127 second dimension of same
- 130 start of EXTRA_CONSTANTS
- 131 length of this array
- 135 start of TEMP_HISTORYFILE
- 136 length of this array
- 140 start of COMPRESSED_FIELD_INDEX1
- 141 length of this array
- 142 start of COMPRESSED_FIELD_INDEX2
- 143 length of this array
- 144 start of COMPRESSED_FIELD_INDEX3
- 145 length of this array
- 150 start of LOOKUP table
- 151 first dimension of this lookup table (64 or 128)
- 152 second dimension of this lookup table
- For dumps and ancillary files, this will correspond to the total number of fields. There is one lookup entry for each level of multi-level fields
 - For fieldsfiles, it also corresponds to the total number of fields, with a notional limit of 4096, although this can be over-ridden via the umui pp-file initialization panel. Another option is to use reinitialisation of fieldsfiles.
 - For boundary datasets, each lookup entry corresponds to a model variable. Data for multi-level variable goes into a single data field. The number of levels stored is given by LOOKUP(17). See section 4.
- 153 number of prognostic fields in dump. vn 3.4 onwards
- (Only used for instantaneous dumps FIXED_LENGTH_HEADER(5) = 1)
- 160 start of DATA : corresponds to start of sector (2048 words) for a well-formed dataset - from vn 4.3.
- 161 dimension of DATA - except for: Fieldsfile where not set correctly
- LBC files vn5.2 onwards where set to 0
- 162 Maximum length of all fields in DATA
- (Only used by observation file so far)

3.2 INTEGER_CONSTANTS

Location and dimensions defined by words 100-101 of FIXED_LENGTH_HEADER

Full list of words reserved for Atmosphere use

1. Number of timesteps since start of run
2. Meaning interval for the mean fields (hours)
3. Number of instantaneous dumps used to generate mean field
4. User defined if non-contiguous period used for meaning. (0) if contiguous otherwise no. of hours between neighbouring contiguous. sections of means
5. User defined if non-contiguous period used for meaning. (0) if contiguous otherwise no. of hours between end of one contiguous. section and start of next
6. Number of points E-W (x direction or 1st dimension)
7. Number of points N-S (y direction or 2nd dimension)
8. Number of levels (P_LEVELS) (numbered away from the earth's surface)
9. Number of wet levels (Q_LEVELS) (ditto)
10. Number of soil levels (ST_LEVELS)
11. Number of cloud levels (CLOUD_LEVELS)
12. Number of tracer levels
13. Number of boundary layer levels
14. Number of 'passive' tracers to be advected excluding moisture
15. Number of different field types in dump
16.
 - vn 4.5 - vn 5.4 : Not used
 - vn 5.5 - : No of timesteps since last river routing
17.
 - vn 4.5 vn 5.1 : Not used
 - vn 5.2 : Algorithm used to generate height fields.
 - 1 Original - defined by 3 regimes where height levels transit from terrain following to constant height.
 - 2 Smooth - quadratic transition of height levels from terrain following to constant height.
18. Number of radiation variables
19.
 - vn 5.0 - vn 5.4 : Not used
 - vn 5.5 - : River routing - row length
20.
 - vn 5.0 - vn 5.4 : Not used
 - vn 5.5 - : River routing - no of rows
21. Value of integer missing data indicator (-32768)
22.
 - vn 4.4- : Calling period (days) for TRIFFID vegetation model otherwise not used.

- 23. • vn 4.4- : Number of atmosphere timesteps since last call to TRIFFID vegetation model otherwise not used.
- 24. • vn 5.0- : First rho level at which height is constant.
- 25. Number of land points
- 26. Number of ozone levels
- 27. Number of levels at which tracers advected
- 28. Atmos Dumps (vn4.1-) and Atmos Fieldfiles (vn5.2-): Number of soil hydrology (moisture) levels. (SM.LEVELS) Atmos Obs files : Number of observations in observation file.
- 29. Number of data values in observation file.
- 30. Time window (minutes) of observations before observation file time.
- 31. Time window (minutes) of observations after observation file time.
- 32. Number of AC Observation Types in observation file.

33-46 : vn 5.0 onwards

- 33. Not used.
- 34. Number of convective levels.
- 35. Radiation timestep number.
- 36. Flag for AMIP run.
- 37. First AMIP year.
- 38. First AMIP month.
- 39. Current AMIP day.
- 40. Current ozone month
- 41. Flag for SH_zonal_quadratics.
- 42. SH_zonal_begin.
- 43. SH_zonal_period.
- 44. SuHe level weight.
- 45. SuHe level cutoff.
- 46. Frictional timescale

For **atmosphere dumps** : Dimension is 46. Words 29-33 are not used.

For **atmosphere fieldfiles** : Dimension is 46

For **ancillary data sets** : Dimension is 15 and the following words are used :-

- 3. Number of different times for which data is present in data set. This is used to determine the periodicity of the data. e.g. 12 for twelve monthly values.
- 6. Number of points E-W (x direction or 1st dimension)

- 7. Number of points N-S (y direction or 2nd dimension)
- 8. Number of vertical levels in model providing or receiving data.
- 15. Number of different field types present in data.

For **atmosphere boundary data sets** : Dimension is 46 and the following words are used :-

- 3. Number of different times for which data is present in data set.
- 6-9. As in full list. The values refer to the model to receive the data.
- 15. Number of different field types present in data.
- 17. As in full list.
- 24. As in full list.

For **atmosphere observation files** : Dimension is 32.

- 1-5. Not used.
- 6-9. As in full list. These values refer to the model being used to assimilate the observations.
- 10-27 Not used.
- 28.32 Used as in full list above.

3.3 REAL_CONSTANTS

Location and dimensions defined by words 105-106 of FIXED_LENGTH_HEADER

Constants marked (*) are 'passive' - see introduction.

Full list of words reserved for Atmosphere use

1. EW (x) grid spacing in degrees (RMDI for variable resolution grids)
2. NS (y) grid spacing in degrees (RMDI for variable resolution grids)
3. Latitude of first PTR row in degrees (latitudes in range 90 to -90)
 - (RMDI for variable resolution grids)
4. Longitude of first PTR point on row in degrees(longitudes in range 0-360) (RMDI for variable resolution grids)
5. Real latitude of 'pseudo' N pole in degrees
6. Real longitude of 'pseudo' N pole in degrees
7. Not used
8. ATMOS YEAR copied to FIXED_LENGTH_HEADER(28)
9. ATMOS DAY copied to FIXED_LENGTH_HEADER(34)
10. ATMOS HOUR copied to FIXED_LENGTH_HEADER(31)

11. ATMOS_MINUTE copied to FIXED_LENGTH_HEADER(32)
 12. ATMOS_SECOND more precise value to be copied to FIXED_LENGTH_HEADER(33)
 13. 13-15 Not used
 16. vn 5.0- : Height at top theta level
 17.
 - vn 5.0 : Pressure at top theta level
 - vn 5.1- : Not used.
 18. Global mean diabatic flux
 19. MASS
 20. ENERGY
 21. ENERGY_DRIFT
- 22-38 : Pre vn 4.0
22. (*)LATITUDE_BAND
 23. (*)VERTICAL_DIFFUSION
 24. (*)SNOW_MASKING_DEPTH
 25. (*)CO2
 26. (*)C
 27. (*)KAY_GWAVE
 28. (*)NU_BASIC
 29. VALUE OF REAL MISSING DATA INDICATOR
 - Version 3.2 onwards : -2**(-30)
 - Pre-Version 3.2 : -32768.0
 30. ATMOS_STEP (converted from history file integer)
 31. (*)FILTERING SAFETY FACTOR
 32. (*)CO2START
 33. (*)CO2RATE
 34. (*)CO2END
 35. (*)CLOUD_WATER_THRESHOLD_OVER_LAND_FOR_CONVERSION_TO_PRECIPITATION
 36. (*)CLOUD_WATER_THRESHOLD_OVER_SEA_FOR_CONVERSION_TO_PRECIPITATION
 37. (*)WIND_LIMIT for half timestep dynamics
 38. (*)DIV_LIMIT Divergence limit for half timestep dynamics
 - 22-38 : vn4.0 onwards
 - 22-28. Not used

- 29. As pre vn 4.0
- 30-38. Not used.

In atmosphere dumps : Dimension is 38 for Version 3.3 (36 for Versions 2.8 - 3.2, 34 for pre-version 2.8) dumps. Words 1-38 (1-36, 1-34) in the full list are used.

In atmosphere fieldsfiles :

Dimension is 38 (6 for pre vn5.0) and the first 38(6) values in the full list above are used.

In atmosphere boundary data sets, the dimension is 38 (6, pre vn 5.0).

Values 1-6 and 16 in the full list above are used.

In atmosphere ancillary data sets :

Dimension is 6 and the first 6 values in the full list above are used.

In atmosphere observation files : Dimension is 34 and words 1-6 and 29 are used as above.

3.4 LEVEL_DEPENDENT_CONSTANTS

Location and dimensions defined by words 110-112 of FIXED_LENGTH_HEADER

For atmosphere data sets except observation files :

The first dimension is the number of model levels while the second dimension refers to :-

See Document Version No 27 for pre vn 4.0

vn 4.0 5.1 See doc vn 40 or earlier AK

vn 5.2 onwards All REAL values

1. Eta values at theta levels
2. Eta values at rho levels
3. RHCRIT Critical RH for cloud amount calculation.
4. Soil Thickness (for ST_LEVELS only)
5. Zsea values for theta levels (#)
6. C values for theta levels (#)
7. Zsea values for rho levels (#)
8. C values for rho levels.(#)

Values 5-8 are set only if using the new smooth height generation method.(Integer Constants Word 17 = 2) otherwise set to RMDI.

(# - See Appendix A)

For atmosphere dumps the second dimension is

8 vn5.2 onwards

4 vn5.0 - vn5.1

6 vn4.0 - vn4.5

corresponding to the lists above.

For atmosphere fieldsfiles the second dimension is

8 vn5.2 onwards

4 vn5.0 - vn5.1

corresponding to the lists above. These refer to the levels of the model providing the results, not the output levels.

For ancillary and boundary data sets the second dimension is 4 and they refer to the model levels for which the data are present (model level of field in LOOKUP(33) for ancillary data sets). These constants are not included for ancillary and boundary data sets containing single level data.

For atmosphere observation files this array stores the levels of the observation data for each observation type.

The first dimension will be the observation type with the maximum number of level information plus two and the data are arranged as follows :-

1. Observation level type

- 0 Single level, no level in data
- 1 Single level, level included in data.
- 2 Model levels.
- 3 Pressure levels.
- 4 Pressure levels corresponding to layer boundaries.
- 10 A-value of hybrid levels.
- 11 B-value of hybrid levels.

2. Number of observation levels or layers.

3. Observation levels. Any levels not used are padded with the real missing data indicator. Note that for N layers there will be N+1 pressure levels corresponding to the layer boundaries.

The second dimension is the number of observation types plus two for the A and B values of the hybrid levels if included.

3.5 ROW_DEPENDENT_CONSTANTS

Location and dimensions defined by words 115-117 of FIXED_LENGTH_HEADER

First dimension is over rows, for atmosphere, second dimension refers to :

- 1. FILTER_WAVE_NUMBER_P_ROWS
- 2. FILTER_WAVE_NUMBER_U_ROWS
- 3. Number of E-W sweeps per row used in tracer advection stability calculations.

Used in atmosphere dumps (pre vn5.2).

From vn7.0 onward, for atmosphere, first dimension is :

-32768 Uniform resolution dump.

Rows Variable resolution dump.

And the second dimension is :

0 Uniform resolution dump.

2 Variable resolution dump (for variable grids phi_p and phi_v).

3.6 COLUMN_DEPENDENT_CONSTANTS

Location and dimensions defined by words 120-122 of FIXED_LENGTH_HEADER

First dimension is over columns

No specific atmosphere use pre vn6.6.

From vn7.0 onward, for atmosphere, first dimension is :

-32768 Uniform resolution dump.

Columns Variable resolution dump.

And the second dimension is :

0 Uniform resolution dump.

2 Variable resolution dump (for variable grids lambda_p and lambda_u).

3.7 FIELDS_OF_CONSTANTS

Location and dimensions defined by words 125-127 of FIXED_LENGTH_HEADER

First dimension is over fields

No specific atmosphere use at present

3.8 EXTRA_CONSTANTS

Location and dimensions defined by words 130-131 of FIXED_LENGTH_HEADER

No specific atmosphere use at present

3.9 TEMP_HISTORYFILE

Location and dimension defined by words 135-136 of FIXED_LENGTH_HEADER. Copy of temporary history common block. Not used from UM vn 3.4.

3.10 COMPRESSED_FIELD_INDEX

Locations and dimensions defined starting at word 140 of FIXED_LENGTH_HEADER

NOTE: The complexities of this indexing system for allowing this option of data compression, mean that it is not used for atmosphere, where possibly only a single land sea compression mask might be required. (This scheme is only viable where a separate mask is required for each level). If only simple 0/1 bitmasks are required, these are held in DATA as integer strings of 0/1. The lookup contains an indicator for compressed fields and a pointer to that part of the model DATA area containing the integer string required for the expansion. Further details in section 4.

Chapter 4

LOOKUP table and PP headers

Location and dimensions defined by words 150-152 of FIXED_LENGTH_HEADER. Full details given in section 4. Also, see section 3 Fixed Length Header, Word 152.

4.1 DATA

Array for all data, pointed to by words 160-161 of FIXED_LENGTH_HEADER and decoded LOOKUP.

From vn 5.0, the data rows have been ordered as South to North for atmosphere dumps, fieldsfiles and ancillary files. Prior to vn 5.0, the data rows run from North to South for all atmosphere UM files.

For observation files word 162 contains the maximum length of observational data which will be the observation type with the highest number of data values. The number of observations varies from file to file and this value enables the correct amount of workspace to be allocated in the UM I/O routines.

NOTE: For atmosphere use : the format of the dump is sufficiently flexible to include any field in any order (although coding consideration imply that multilevel fields (and all soil fields and all vegetation fields) are contiguous). A multitude of diagnostic fields of both 1 and 2 dimensions can be included if desired, but strictly a dump should only contain those fields necessary for restarting.

4.2 The LOOKUP table: PP header content

For Dumps, fieldsfiles, Boundary data sets and ancillary data sets, each entry in the lookup table is a 64 word PP header. For Observation files a 128 word header is used see below section 5 for details.

Each word is detailed by: 'Word No.' 'Name' 'Contents'

(Words 1-45 INTEGERS)

- | | | |
|-----------------------|---|---|
| 1. LBYR Year | } | Validity time for instantaneous field.
For Accumulations/Mean: start of A/M period.
Start time of partial timeseries if
data record is a timeseries. |
| 2. LBMON Month | | |
| 3. LBDAT Day of Month | | |
| 4. LBHR Hour | | |
| 5. LBMIN Minute | | |
| 6. LBDAY Day number | } | if LBREL \leq 2 |
| 6. LBSEC Seconds | | if LBREL \geq 3 |

7. LBYRD Year	}	Data time for instantaneous field.
8. LBMOND Month		For Accumulations/Means: end of A/M period.
9. LBDATD Day of month		End time of partial timeseries if
10. LBHRD Hour		data record is a timeseries.
11. LBMIND Minute		
12. LBDAYD Day number		
12. LBSECD Seconds		if LBREL \leq 2
		if LBREL \geq 3

- If using FIELDCOS with OPER=T in namelist, Words 1-6 become the validity time and Words 7-12 become the data time for Accumulations and Means in the converted file.

13 LBTIM Time indicator.

- This indicates what the times in words 1-12 represent. Referring to the times represented by words 1-6 and 7-12 as T1 and T2, LBTIM is coded as (100*IA + 10*IB + IC) where:
 - IA
 - for time aggregated fields IA is the time interval in hours between the individual fields from which the mean was computed
 - IA may be left as zero for time-means to indicate that the time interval is unspecified
 - IB
 - = 0 if only the validity time (T1) is valid.
 - = 1 if the field is a forecast from T2 valid at T1.
 - = 2 if the field is a time mean between T1 and T2, or represents a sequence of times between T1 and T2.
 - = 3 if the field is a time mean from T1 to T2 for each year from LBYR to LBYRD.
 - = 4 if the field is a difference between fields valid at T1 and T2 (in sense T2-T1).
 - = 5 if the field is a mean daily cycle between T2 and T1
 - IC
 - = 0 if 'model time' is used for T1 and T2 (i.e. only day number, hour and minute are set).
 - = 1 if the Gregorian calendar is used for T1 and T2.
 - = 2 if the '360-day' calendar (i.e. 12 30-day months) is used for T1 and T2.
 - = 3 if 'model time' is used for T1 and T2 (i.e. only day number, hour and minute are valid; year, month and day in month are to be ignored if set).
 - * If 'IC' is 1 or 2, coding of the 'day numbers' (words 6 and 12) is optional: code as 0 if not used.

14 LBFT Forecast period (hours) : The difference between validity time and data time. For accumulation/mean fields: the reference time is derived by the lbft offset from T2 (the end of the A/M period).

15 LBLREC Length of data record (including any extra data)

- **Dumps** : LBLREC is the original number of data points.
- **fieldsfiles** : LBLREC is the number of words needed to hold the information.
- **Timeseries** : LBLREC is the length of the timeseries. Note that number of individual points sampled is LBLREC/LBUSER(3).
- **Boundary datasets (vn5.2-)** : LBLREC is the number of data points. See Words 17, 18, 19 and 41 for details on grid for boundary data. Data for all multi-level variables are stored as one field.

- For fieldsfiles LBLREC may be machine dependent. For unpacked data it makes no difference, but for packed data it does. On any 32 bit word machine the value of LBLREC will be twice the value it has on a 64 bit word machine, because it takes twice as many 32 bit words to hold the same amount of data as 64 bit words.

16 LBCODE Grid code. This indicates the type of grid and is coded as:

- 1 Regular latitude/longitude grid.
 - 2 Regular lat/lon grid boxes (grid points are box centres).
 - 3 Polar stereographic grid.
 - 4 Spectral coefficients.
 - 7 Mercator grid.
 - 8 Plane polar grid.
 - 9 Plane Cartesian grid.
- For grids with non-standard polar axis, add 100 to the above numbers.
 - For cross sections, code as (10000 + 100*IX + IY) where IX and IY are codes for the x- and y-axes from the list below. Cross section fields indicated in this way must contain x- and y-coordinate vectors in the extra data.
 - When LBCODE is coded as 30000 + 100*IX + IY, with IX and IY from the same list, the axes are given the same interpretation as for a normal cross section, but coordinate vectors need not be supplied. In this case, however, the field is not regarded as a cross section by PP cross-section routines.
 - Axis codes are as follows:
 - 0 Sigma (or eta, for hybrid coordinate data).
 - 1 Pressure (mb).
 - 2 Height above sea level (km).
 - 3 Eta (U.M. hybrid coordinates) only.
 - 4 Depth below sea level (m).
 - 5 Model level.
 - 6 Theta.
 - 7 Sigma only.
 - 8 Sigma-theta.
 - 10 Latitude (degrees N).
 - 11 Longitude (degrees E).
 - 12 (Horizontal) distance (km).
 - 13 Site number (set of parallel rows or columns e.g. Time series)
 - 20 Time (days). (Gregorian calendar (not 360 day year) if distinction applicable)
 - 21 Time (months).
 - 22 Time (years).
 - 23 Time (model days with 360 day model calendar)
 - 31 Logarithm to base 10 of pressure in mb.
 - 40 Pseudolevel
 - 99 Other.
 - Known values of LBCODE in use :-
 - 1 Unrotated regular lat/long grid
 - 101 Rotated regular lat/long grid
 - 11110 Variable grid - pp method

31320 Gregorian calendar timeseries

31323 Timeseries in other calendar

17 LBHEM Hemisphere indicator. For geographical grids this is coded as

0 Global field (i.e. covering the WHOLE globe).

1 Northern hemisphere polar stereographic grid, or other geographic grid covering the WHOLE northern hemisphere.

2 Southern hemisphere polar stereographic grid, or other geographic grid covering the WHOLE southern hemisphere.

3 Limited area of globe without 'wrap-around' (i.e. NOT covering the full 360-degree longitude range).

4 Limited area of globe with 'wrap-around' (i.e. covering the full 360-degree longitude range).

- For SPECTRAL COEFFICIENTS, code 0, 1 or 2 as appropriate.
- For CROSS-SECTION FIELDS and TIME SERIES/TRAJECTORIES, a value of 3 should be coded.
- For GRAPH FIELDS, a value of 5 should be coded.
- Boundary datasets:
 - 99 (pre vn5.0) Set to 99 for a 'rim' data set as used for lateral boundary data sets.
 - 100+n (vn5.2-) n is the number of levels for the variable

18 LBROW Number of rows in grid

- Boundary datasets. (pre vn5.0) LBROW is set to the width of the boundary zone for lateral boundary data sets. (vn5.2-) LBROW is the no of rows excluding any halo rows.
- Time series - set to -99
- Set to zero for land packed fields

19 LBNPT Number of points per row in grid

- Set to zero for land packed fields

20 LBEXT Length of extra data.

- Extra data is arranged as one or more data vectors. These consist of an integer descriptor code followed by the data values. See Appendix 3 for more information.

21 LBPACK Packing method indicator vn 2.8 onwards.

- The LBPACK code describes packing and compression of fields in memory and on disk. The integer value is the combination of a series of codes as follows :-
 - LBPACK = N5N4N3N2N1
 - (eg LBPACK=1210; N5=0, N4=1, N3=2, N2=1, N1=0)
 - N1 Packing
 - * (reduces storage requirements by removing land points from fields within ocean fieldsfiles - a sequence of missing data values is mapped into a single missing data value followed by a value indicating the length of the run of missing data values).
 - 0 data not packed.
 - 1 data packed using WGDOS archive method. (*) see Appendix 2 for details on WGDOS packing.
 - 2 data packed using CRAY 32 bit method.

- 3 data compressed using the GRIB method.
- 4 data compressed using Run Length Encoding
- N2 Data Compression
 - 0 data not compressed.
 - 1 Data compressed using the N3rd group of compressed field index arrays in the dump.
 - 2 Data compressed with the N3rd bit mask
- N3 Compression
 - * If N2=1, then
 - N3 is the number of the group of compressed field index arrays used
 - * If N2=2, then
 - N3 = 1 use land mask
 - N3 = 2 use sea mask
- N4 Number Format
 - 0 Native (to the target machine) format
 - 1 IBM format
 - 2 CRAY format
 - 3 IEEE format
 - 4 GRIB format
 - 5 Standard VAX format
- N5 Reserved for future use (=0)
- Note that the WGDOS packing method would not usually be adopted because it would not allow results to be replicated on a restart
- **Change for 4.5:**
 - * N4 : This has been reset from 2 to 3 for **fieldsfiles**.

22 LBREL Header release number From **vn 3.3**

- Set to 2 for UM versions up to 8.0.
- set to 3 for UM version 8.1 onwards
 - usage of LOOKUP items 6&12 changed at UM vn8.1

23 LBFC Field code. This indicates what the data in the field represents. A list of field codes is available online at http://www-hc/umdoc/pp_package_ibm_docs/@fcodes.

24 LBCFC Second field code (=0 in UM)

25 LBPROC Processing code. This indicates what processing has been done to the basic field. It should be 0 if no processing has been done, otherwise add together the relevant numbers from the list below :

- 1 Difference from another experiment.
- 2 Difference from zonal (or other spatial) mean.
- 4 Difference from time mean.
- 8 X-derivative (d/dx)
- 16 Y-derivative (d/dy)
- 32 Time derivative (d/dt)
- 64 Zonal mean field
- 128 Time mean field
- 256 Product of two fields

- 512 Square root of a field
- 1024 Difference between fields at levels BLEV and BRLEV
- 2048 Mean over layer between levels BLEV and BRLEV
- 4096 Minimum value of field during time period
- 8192 Maximum value of field during time period
- 16384 Magnitude of a vector, not specifically wind speed
- 32768 Log10 of a field
- 65536 Variance of a field
- 131072 Mean over an ensemble of parallel runs

(NOTE: The UM does not use any values of LBPROC above 8192).

- 26 LBVC Vertical co-ordinate type. The vertical co-ordinate is the one which has the same value at every grid point; e.g. pressure for a 500mb height field or longitude for a zonal mean cross section. The coordinate of field codes as for word 23. Numbers 126-139 are used for special levels. The most common vertical co-ordinate types are :

- 1 Height (m)
- 8 Pressure (mb)
- 9 Hybrid co-ordinates ($A \cdot \sigma + B \cdot \text{pressure}$)
- 10 Sigma (p/p^*)
- 65 Hybrid height co-ordinates (vn5.2-) See Appendix 1
- 128 Mean sea level
- 129 Surface
- 130 Tropopause level
- 131 Maximum wind level
- 132 Freezing level
- 142 Upper hybrid level
- 143 Lower hybrid level
- 176 Latitude (deg)
- 177 Longitude (deg)

- 27 LBRVC Vertical co-ordinate type for reference level. Always zero.

- 28 LBEXP Experiment number (optional parameter for users reference only).

- In OPERATIONAL use this is set to the ITAB number:

- 1 Global main run data
- 2 Global update run data
- 3 Limited-area main run data
- 4 ECMWF data
- 5 Washington data
- 6 Limited-area update run data
- 7 Global wave data
- 8 European wave data
- 9 UK Mesoscale model main runs
- 10 Paris data
- 21 Bosnia Mesoscale model runs

- 22 Gulf Mesoscale model runs
 - 23 Stratospheric model runs
 - 24 Short term Mesoscale models
 - 25 FOAM model runs
 - 26 SURG fieldsfile
 - 27 'Non-critical' suite (from Dec. 1998)
 - 28 Middle East Mesoscale model
 - 29 SW Asia model (Formerly Indonesian Mesoscale model)
 - 30 NW European Baroclinic ShelfSeas model (Formerly Baroclinic Shelf Seas Model)
 - 31-40 will not be used by operational models
 - 41 Extended Middle East model (Formerly Oman Mesoscale model)
 - 42 Gulf Baroclinic ShelfSeas model
 - 43 Indian ocean FOAM model
 - 44 European Model
 - 61 4km model
- CLIMATE MODEL: This word contains the 5-character experiment and job name in a form encoded by the UM subroutine EXPT_ENC. Encoding can also be done by the PV-WAVE routine encode_expt and decoding by decode_expt.

29 LBEGIN Disk address/Start Record

- Files with well-formed records : Disk address (Word Number) relative to start of file. From vn 4.3
- Boundary datasets with well-formed records : (Pre vn 5.0) Only the start of the data for each validity time is aligned with the start of a sector. All the data for the same validity time is contiguous. For example, in a file with 5 variables for each validity time, only the disk addresses for fields 1, 6, 11, 16 etc. are aligned with a sector boundary. (vn5.2-) All data fields are aligned with a sector boundary.
- Fieldsfile direct access data sets : Start record
- For all other files including those not set up with well-formed records : set to zero

30 LBNREC Disk length/No of records.

- Dumps and Ancillary files with well-formed records : Length on disk which is number of words (after packing for packed fields). The length includes rounding up to an exact number of sectors. Sector length currently in use is 2048 words.
- Boundary datasets with well-formed records. (Pre vn5.0) As dumps except that the length is the exact length and is not rounded to an exact number of sectors.(vn 5.2-) All data fields are rounded to an exact number of sectors.
- Fieldsfile direct access data sets : No of records
- For all other files including those not set up with well-formed records : set to zero

31 LBPROJ fieldsfile projection number; depends on grid resolution (applies to global grids only).

- Number of EW points = 192, LBPROJ is 802
- Number of EW points = 188, LBPROJ is 800
- Number of EW points = 96, LBPROJ is 870
- Number of EW points = 432, LBPROJ is 800
- Otherwise, LBPROJ is 900

32 LBTyp fieldsfile field type; equal to stash master entry PPF (see UMDP C4)

33 LBLEV fieldsfile level code

- Set to 7777 for a multi-level field, as occurs in lateral boundary data sets.
- Set to 8888 for single level fields which are not at the surface e.g. boundary layer depth after timestep.
- Set to 9999 for Surface fields, and for zeroth level of items on rho levels.
- Set to value of CFL from STASHmaster if considered to be at special level (e.g. levelT == 5 in STASHmaster).
- Otherwise, equal to level number

34 LBRSVD(1) Reserved for future PP-package use

35 LBRSVD(2) Reserved for future PP-package use

36 LBRSVD(3) Reserved for future PP-package use

Words 34-36 are reserved for PP-package use and should not be used by programmers. Word 37 was previously reserved but from vn7.1 is in use.

37 LBRSVD(4) Ensemble member number ; 0 for single model (From vn7.1)

38 LBSRCE

- Pre vn 5.3 LBSRCE=YYYY
- vn 5.3 - LBSRCE=XXXXYYYY
- YYYY is the model identifier, 1111 for Unified Model
- If YYYY=1111, Words 39-45 = LBUSER area for UM use.
- XXXX is the UM Version Number. ie. For 5.3, XXXX=0503

39 LBUSER(1) DATA_TYPE Data Type vn 2.8 onwards

- 1 Real field
- 2 Integer Field
- 3 Logical Field

- Change for 4.5 For fieldsfiles, the data type for the Land-Sea mask has been reset from 1 to 3. Note that CUMF will not compare a 4.4 and 4.5 mask.

40 LBUSER(2) NADDR

- Start address in DATA. Not to be confused with disk address - Word 29.
- Relative to start of DATA, i.e. First field NADDR=1
- For fieldsfiles relative to start of Fixed Length Header
- Set to -99 if LBEGIN not known

41 LBUSER(3)

- Boundary datasets (vn5.2-) : stores the rimwidth and halo sizes for each variable as R1R0Y1Y0X1X0 where R1R0 is the rimwidth, Y1Y0 is the y-halo size and X1X0 is the x-halo size. Example :- 040505 : Rimwidth=4, y-halo=5, x-halo=5.

42 LBUUSER(4) ITEM_CODE

- Stash Section number and Item number
- Held in the form : Section Number*1000 + Item Number

43 LBUUSER(5) LBPLEV

- Stash Pseudo Dimension
- Used to specify level eg. Radiation Band or Assimilation Group.

44 LBUUSER(6) Free space for users.

- Used by external program FIELDCOS to denote length of unpacked field size.
- Contains number of times FIELDOP has been applied to a dump of fieldsfile. From vn 4.3.

45 LBUUSER(7) MODEL_CODE: Internal Model Number. vn 4.0 onwards.

- 1 Atmosphere
- 5 Ice (vn 5.5 onwards)
- 10 Diagnostics derived by the FieldCalc program

Words 46-64 are REAL values

46 BULEV Upper layer boundary or BRSVD(1)

47 BHULEV Upper layer boundary or BRSVD(2)

- For hybrid levels:
 - BULEV is B-value at half-level above.
 - BHULEV is A-value at half-level above.
- For hybrid height levels (vn5.2-, Smooth heights) (#)
 - BULEV is Zsea of upper layer boundary
 - * If rho level: Zsea for theta level above
 - * If theta level: Zsea for rho level above
 - BHULEV is C of upper layer boundary
 - * If rho level: C for theta level above
 - * If theta level: C for rho level above

48 BRSVD(3) Reserved for future PP-package use

49 BRSVD(4) Reserved for future PP-package use

Words 48-49 are reserved for PP-package use and should not be used by programmers. Words 46-47 were reserved and have been used in the UM since vn 4.0.

50 BDATUM Datum value constant subtracted from each field value.

- Usually zero ; would be 273.15 for a temperature field in deg C.

51 BACC (Packed fields) Packing accuracy

52 BLEV Level (the value of LBVC, word 26)

53 BRLEV Lower Layer Boundary

54 BHLEV Level

55 BHRLEV Lower layer boundary

- For hybrid levels:
 - BLEV is B-value of level.
 - BRLEV is B-value of half-level below.
 - BHLEV is A-value of level.
 - BHRLEV is A-value of half-level below.
- For hybrid height levels (vn5.2-, Smooth heights) (#)
 - BLEV is Zsea of rho/theta level
 - BRLEV is Zsea of lower layer boundary, i.e. theta/rho level below
 - BHLEV is C for rho/theta level
 - BHRLEV is C for theta/rho level below
- For single or special levels (e.g. LBVC \geq 126 .AND. LBVC \leq 139):
 - BLEV, BRLEV, BHLEV, BHRLEV all 0.0
- Real levels (eg. pressure levels):
 - BLEV = Real level (usually related to value of LBVRC, word 33 but could be -1 if single level not covered above)
 - BRLEV, BHLEV, BHRLEV all 0.0

(#) For notes on vertical representation of height levels, see Appendix A.

56 BPLAT Real latitude of 'pseudo' N pole (equal to fixhd(5))

- Code as 90.0 for grid with normal polar axis.

57 BPLON Real longitude of 'pseudo' N pole (equal to fixhd(6))

- Code as 0.0 for grid with normal polar axis.

- | | | |
|--|---|--|
| 58. BGOR
59. BZY
60. BDY
61. BZX
62. BDX | } | These parameters define the grid for the field.
The coding depends on what type of grid the field is on.
Details for various types follow. All are zero for time series. |
|--|---|--|

(i) Latitude/longitude grids.

- BGOR Not used - set to zero.
- BZY Latitude of 'zeroth' row (i.e. an imaginary row one grid length before the first row) in degrees.
 - Pre-vn5.0: row at or nearest N pole
 - vn5.0- : row at or nearest S pole
 - Variable resolution grids: code as RMDI
- BDY Latitude interval between rows in degrees (negative if rows are north to south).
 - Variable resolution grids: code as RMDI

- BZX Longitude of 'zeroth' point in row (i.e. an imaginary point one grid length before the first point) in degrees (east positive).
 - Variable resolution grids: code as RMDI
- BDX Longitude spacing of points in each row in degrees (negative if points run from east to west).
 - Variable resolution grids: code as RMDI

(ii) Polar stereographic grids.

- BGOR Grid orientation. This is the longitude in degrees of the meridian which would be vertical with north at the top on a chart drawn for the grid.
- BZY Reference latitude in degrees (see BDY).
- BDY Grid length in metres at reference latitude.
- BZX BDX x, y-coordinates (In grid lengths, can be fractional, need not be within grid area; bottom left grid point of a chart counts as the point (1.0, 1.0))

(iii) Mercator grids.

- Coded as in (i) except that BZY and BDY are values of $(\text{earth's radius}) * \log((1+\sin(\text{latitude}))/\cos(\text{latitude}))$ instead of latitude.
- (Earth's radius is in metres and 'log' is the natural logarithm.)

(iv) Cross sections.

- Coded as in (i) above except that BZY and BDY are values of y co-ordinate and BZX and BDX are values of 'x' co-ordinate.
- (Note that for zonal mean fields or cross sections along a meridian, BZX and BZY are longitudes.)
- If the cross section is on irregularly spaced levels, BZY and BDY should both be zero.

(v) Graph fields.

- Coding is as in (i) above except that BZX and BDX are values of the 'x' co-ordinate, and BDY and BZY are both coded as 1.0.
- If points on the graphs are not evenly spaced in the 'x' co-ordinate, BZX and BDX should both be coded as zero.

(vi) Spectral coefficients.

- BGOR Not used - coded as zero.
- BZY Not used - coded as zero.
- BDY Not used - coded as zero.
- BZX Reference longitude - usually 0.0. This is the meridian which is treated as the origin of longitude for the coefficients.
- BDX Not used - coded as zero.

63 BMDI Value used in the field to indicate missing data points (always RMDI, even for integer fields). If the field contains no missing data, code a value of -1.0E30.

64 BMKS M.K.S scaling factor, always 1.0.

Chapter 5

Atmosphere Observation Files

An observation file has the following components :-

Fixed Length Header, Integer Constants, Real Constants, Level dependent constants, Lookup table and a Data section.

The lookup table for an observation file is different from the other data sets as its first length is 128 compared to 64 in the other data sets. The first 64 words is the PP Header as described in section 4 and Words 65-128 contains extra information on the observation data.

Word 65-75 is reserved for values specific to each observation type. Those in use are

65 AC Observation Type Number

66 Number of Observations

67 Number of Data Values per observation

68-75 reserved for future use

76-128 reserved for the data descriptors to describe the data contents and is used by the ocean observation files and maybe in the atmosphere observation files in future.

The total number of data values for an observation type is Word 66 multiplied by Word 67. The total number of observations and data values in an observation file are given by Word 28 and 29 respectively of the Integer Constants array.

Observation Data

The observation data is contiguous by observation type and is ordered according to the headers in the lookup table. For each observation type, the data is contiguous by data value. The data is stored in 32 bits on disk and 64 bits in memory.

5.1 Observation files for VAR and OPS

Details on the file formats for the new schemes can be found at :-

<http://www-nwp/opsrc/OPS/view/dev/doc/OTDP16.html>

and

<http://www-nwp/opsrc/OPS/view/dev/doc/OTDP17.html>

Appendix A

Model level Definition

From UM Version 5.2, the method of defining the model levels in PP headers was revised.

At vn5.0 and 5.1, eta values were used in the pp headers to specify the levels of model data, which was of limited use when plotting data on model levels.

From 5.2, the pp headers were redefined to give information on the height of the level. Given a 2D orography field, the height field for a given level can then be derived.

The height co-ordinates for pp-output are defined as :-

$$Z(i, j, k) = Z_{sea}(k) + C(k) * orography(i, j)$$

Where $Z_{sea}(k)$ and $C(k)$ are height based hybrid coefficients.

$$Z_{sea}(k) = eta_value(k) * Height_at_top_of_model$$

$C(k) = [1 - eta_value(k)/eta_value(first_constant_rho_level)] * *2$ For levels less than or equal to (first constant rho level)

$C(k) = 0.0$ For levels greater than (first constant rho level)

Where $eta_value(k)$ is the eta value for theta or rho level k. The eta value is a terrain-following height coordinate; full details are given in UMDP15, Appendix B.

The pp headers store Zsea and C as follows :-

46=bulev Zsea of upper layer boundary

52=blev Zsea of level

53=brlev Zsea of lower layer boundary

47=bhulev C of upper layer boundary

54=bhlelev C of level

55=bhrlev C of lower layer boundary

If the level is a theta/rho level then the upper and lower layer boundaries will be the rho/theta levels above and below.

For the top theta level, the upper layer boundary is a rho level above which is calculated to be the same distance above as the rho level below.

If the top rho level is above the top theta level, the upper layer boundary is set the same as the top rho level.

In atmosphere dumps and fieldfiles, the Zsea and C values are also stored in the level dependent constants header. See section 3.4.

Appendix B

WGDOS packing in the UM

See also appendix 3 of UMDP C4 on STASH gives information about packing codes and how they are interpreted.

B.1 COMPRESSION AND EXPANSION ROUTINES

These subroutines have been written as general purpose data compression routines to be used to pack data for transfer from the supercomputer to its front end, and also to unpack the packed data.

The programs have as far as possible been written in standard FORTRAN and are compatible with existing IBM assembler routines COEX described in Operational Suite Documentation paper 9.1 which may be used to unpack data on the IBM. N.B. IBM assembler routines cannot process data containing a missing data indicator.

```
SUBROUTINE COEX(FIELD,M,ICOMP,N,IX,IY,NUM,ISC,OCO,AMDI,LWORD)
```

```
!                                OCO=.TRUE.                | OCO=.FALSE.
REAL FIELD          ! (ARRAY) = FIELD FOR COMPRESSING    | RETURNED EXPANDED DATA
INTEGER M           ! = SIZE OF FIELD FOR COMPRESSING    | SIZE OF RETURNED FIELD
INTEGER ICOMP       ! (ARRAY) = RETURNED COMPRESSED DATA | FIELD FOR EXPANSION
INTEGER N           ! = SIZE OF COMP                      | -----
INTEGER IX          ! = X DIMENSION OF FIELD             | RETURNED X DIMENSION
INTEGER IY          ! = Y DIMENSION OF FIELD             | RETURNED Y DIMENSION
INTEGER NUM         ! = NO. OF COMPRESSED WORDS RETURNED | -----
INTEGER ISC        ! = ACCURACY IN POWER OF 2
LOGICAL OCO        ! = .TRUE. FOR COMPRESSION .FALSE. FOR EXPANSION
INTEGER LWORD      ! = LENGTH IN BITS OF FULLWORD VARIABLES
```

Subroutine to COMPRESS called in COEX

```
SUBROUTINE CMPS(IX,FIELD,ICOMP,NOP,APREC,IBIT,BASE,AMDI)
```

```
INTEGER IX          ! = LENGTH OF FIELD
REAL FIELD         ! (ARRAY) = FIELD FOR COMPRESSING
INTEGER ICOMP       ! (ARRAY) = RETURNED COMPRESSED DATA
INTEGER NOP        ! = NUMBER OF WORDS OF COMP FILLED BY THIS CALL
REAL APREC         ! = PRECISION AS DECIMAL FRACTION
INTEGER IBIT       ! = NUMBER OF BITS INTO WHICH DATA PACKED
```

```

REAL BASE      ! = REFERENCE (MIN) VALUE FOR PACKED DATA
REAL AMDI     ! = MISSING DATA INDICATOR

```

Subroutine to EXPAND called in COEX

```

SUBROUTINE XPND(IX,ICOMP,FIELD,APREC,IBIT,BASE,NOP,AMDI)

```

```

INTEGER IX      ! = LENGTH OF FIELD
INTEGER ICOMP   ! (ARRAY) = DATA FOR EXPANSION
REAL FIELD     ! (ARRAY) = FIELD OF EXPANDED DATA
REAL APREC     ! = PRECISION AS A DECIMAL FRACTION
INTEGER IBIT   ! = NUMBER OF BITS INTO WHICH DATA PACKED
REAL BASE     ! = REFERENCE (MIN) VALUE FOR PACKED DATA
INTEGER NOP    ! = SIZE OF COMP
REAL AMDI     ! = MISSING DATA INDICATOR

```

B.2 Format of Compressed data

Word	Bit
	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F

	FIELD HEADER

1	TOTAL LENGTH OF PACKED FIELD (32 bit words) = 32 bit integer

2	PRECISION (power of 2) = 32 bit integer

3	No. POINTS IN ROW = 16 bit INT No. ROWS IN FIELD = 16 bit INT

	ROW HEADER

4	BASE VALUE (Minimum of Field) = 32 bit IBM real

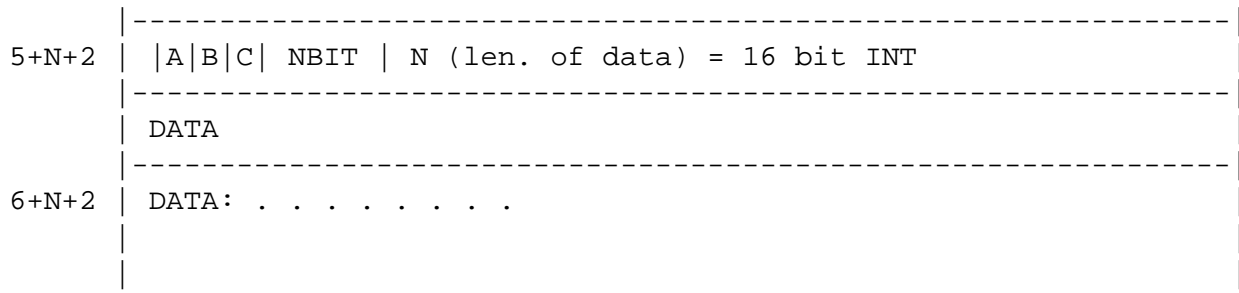
5	C B A NBIT N (len. of data) = 16 bit INT

	DATA

6	DATA: if A=1 bit map for MDI, if B=1 bit map for minimums, if C=1 bitmap for zeros. Followed by packed data NBIT bits per data point. NBIT = 5 bit positive integer.

	ROW HEADER

4+N+2	BASE VALUE (Minimum of Field) = 32 bit IBM real



Appendix C

FORMAT OF DATA RECORD FOR PP-FIELDS

PP files are created from UM fieldsfiles by the convpp utility. A PP file comprises pairs of header records and data records. Each header record provides a description of the contents of the corresponding data record. This is normally a two-dimensional array containing values of a particular field (eg. values of a model field at a given level). It is also possible to store cross-sections, but this entails a set of 'extra data' which is stored at the end of the data record (see below).

PPFiles: LBLREC is the number of words needed to hold the information. This defines a maximum size for a field, including both data and any 'extra data'.

PPFiles are 32bit so this limit in bytes is:

$$4 \times (2^{31} - 1) = 8589934584 \approx 8.5GB$$

However PP files provide a record control word, which defines the length of the field in Bytes, this is also a 32bit integer. Applications, including MOOSE and TIDL, make use of this record control word for file reading. This imposes a smaller limit on a PP field which can be read by TIDL and MOOSE:

$$2^{31} - 1 = 2147483647 \approx 2.1GB$$

The maximum size of a field is taken to be the lower of these two limits, 2.1GB ¹

C.1 Order of grid points

Data is normally in the same order as model grid points. The values are stored row-wise i.e. data for a complete row is followed by data from the next row. A row is normally along a line of constant latitude. The order is indicated by the sign of header variables BDX and BDY (so a negative value of BDY and a positive BDX indicate rows ordered north to south and points within a row ordered west to east). Before data is plotted it is re-oriented as appropriate (so that, for a lat-long chart the northern-most row is at the top).

Cross-section (and similar) type fields are always stored such that the first row stored will be at the bottom of the cross-section as plotted. Within a row the data is stored left-to-right. This is also true for fields on a polar-stereographic grid.

¹When the limitation of field size in pp fields was first encountered it was in an example that affected only one, fairly specialised, application. The intention was to move away from the pp file format. It is hoped that the retirement of pp can be achieved before more Applications are affected by the large file limitation. If you find you are affected by the large field size limitation then please contact the UM System Team.

For spectral harmonic representation, the spectral coefficient values are stored in a similar manner to grid-point values. These are complex numbers, so they are stored as pairs comprising the real and imaginary parts. They are stored in the following order:

$$(m,n) = (0,0), (0,1), (1,1), (0,2), (1,2), (2,2), \dots, (N,N)$$

where N is the truncation level for n.

Normally each row of a PP-field is the same length, but rows of different length are also allowed. Currently this facility is only used for spectral data.

C.2 Extra data

The data record may include 'extra data' in addition to the usual grid-point values. Cross-section fields must always be followed by extra data giving the x- and y-coordinate values of each column and row of the grid. (This is to allow irregular grid). It is also possible to use extra data in conjunction with other types of field, eg. a character string description of the field.

The header variable LBEXT shows the length of extra data associated with a field. Thus the total length (in words) of the data record (LBLREC) equals the number of grid-point values plus LBEXT.

Extra data is arranged in vectors, each of which is made up of an integer code followed by data values. The integer code is $(1000 \times IA + IB)$, where IA is the length of the vector in words (i.e. usually the number of data values) and IB is a standard code:

1. vector gives LBNPT x-coordinate values
2. vector gives LBROW y-coordinate values
3. vector gives lower y_limit of domain (real)
4. vector gives lower x-limit of domain (real)
5. vector gives upper y_limit of domain (real)
6. vector gives upper x_limit of domain (real)
7. vector gives lower z-limit of domain (real)
8. vector gives upper z-limit of domain (real)
9. character title of field (see TITLFD)
10. character title of domain (character - ASCII)
11. vector gives lower x-boundary (real)
12. vector gives upper x-boundary (real)
13. vector gives lower y-boundary (real)
14. vector gives upper y-boundary (real)

A zero integer code indicates no (more) extra data.

For cross-sections the extra data MUST start with an x-vector followed by a y-vector; optionally other extra data may follow.

Appendix D

Towards Portable PP files

When the Met Office migrated from HP Unix to Linux running on an Intel based architecture a decision was made to keep the real number format of pp files as 32 bit big endian. This decision did not propagate to all platforms where pp files were stored, and it was little advertised within the External (then Portable) UM community. It was, however, adopted on the desktop and server architecture and used as the 'canonical' format in MASS.

This appendix attempts to document some of the decisions made about the physical format of pp files. It also highlights some areas where, as far as we know, no decisions have been made and so there may still be platform dependent implementations of the pp format. This documentation has been put together after the fact, and may not be complete. Please contact the document owner if you can fill in any gaps or think there is an inaccuracy in what follows.

D.1 Records

PP files use HP FORTRAN record control words around both the header and data sections. Each header and each data section are an HP FORTRAN record. Each record (both header and data) begin and end with a record control word. The record control word is identical for the start and end of a record. The record control word is a big endian integer giving the number of bytes in the record (so 256 for the 64 word header). Ideally (though this may not always be the case) the length of the records as indicated by the record control words for the data should be equal to LBLREC.

D.2 Data types

Within the header, data, and extra data sections real numbers are represented as 32 bit IEEE big endian reals. The one exception to this is in WGDOS records where the row offset is stored as an IBM 32 bit real.

With the exception of the variable length integers in WGDOS packed records all integers are 32 bit signed big endian.

The representation of logical and string data within the data and extra data sections is undefined and so can be application and platform dependent.