

**UNIFIED MODEL DOCUMENTATION PAPER NO C4**

**STORAGE HANDLING AND DIAGNOSTIC SYSTEM (STASH)**

by

K.Rogers

Version 14

2 March 1999

Model Version 4.5

Climate Research  
Meteorological Office  
London Road  
BRACKNELL  
Berkshire  
RG12 2SY  
United Kingdom

(c) Crown Copyright 1999

This document has not been published. Permission to quote from it must be obtained from the Head of Climate Prediction at the above address.

Modification Record to version 10

Version 9 STASHmaster and preSTASHmaster section (through to the end of the paper) updated for model version 2.8. Version number and date information incorporated on front page and this modification record. Section 5 removed.

Version 10 STASHmaster and preSTASHmaster section (through to the end of the paper) updated for model version 10.

<b>Modification Record from version 11</b>		
<b>Document version</b>	<b>Author</b>	<b>Description.....</b>
11	M.Carter.	Add changes to preSTASHmaster definition to allow user prognostics for UM version 3.3.
12	S.J.Swarbrick	Interim update for UM vn4.0.
12.1	S.J.Swarbrick	Section 6 added.
13	S.J.Swarbrick	Document rewritten for UM vn4.1.
14	D. Robinson	Document updated to UM Vn 4.5. Appendix 2, new entries 29-33. Appendix 3, new Space, PseudT, PseudL and Option codes. Appendix 5 added.

## **1. Introduction and Background**

STASH is an acronym for "Spatial and Temporal Averaging and Storage Handling". This document outlines the basic structure of the STASH system as used in version 4.1 of the Unified Model. The system developed at vn4.0 was in certain respects very different from that which existed up to vn3.4. The changes introduced at that stage were intimately associated with the development of submodel partitioning as a structural feature of the UM. Further significant changes have been introduced at vn4.1.

In view of the complexity of the STASH system, it is appropriate to give some indication of the motivation which lay behind its original development. A fundamental aspect of the UM structure is the "plug-compatibility" barrier. The plug-compatibility rules, which the "physics" routines conform to, require that routines make no assumptions about the format of permanent storage, that they do not call any routines which do so, and that they communicate with other routines only via their argument lists. Dynamics routines also obey these rules as far as possible, given that they have to do horizontal differencing.

It is desirable that any experimental diagnostic can be introduced into a plug-compatible routine (PCR) at short notice without the developer having to know anything about the storage or averaging needed. In order to facilitate this, a very general user-friendly storage and averaging system is required.

The system must cope with a variety of non-meteorological processing such as averaging, but physical diagnostic calculations do not come within its remit - it only deals with their results. However it must be able to apply several possible treatments in time (eg. replace, accumulate) and a number of options for spatial treatment (eg. full fields, zonal and meridional means, limited-area means, single-point data). A particular quantity may be diagnosed in more than one of these ways at once.

Code of the complexity that is required for STASH processing is best not duplicated in different control routines, and also for clarity it is better if it is distinct from other control functions. To reduce calling overheads, the various STASH processing functions should be grouped in a single set of modular routines.

Up to UM version 3.4, the processing of STASH requests submitted via the user interface, and the computation of space requirements and addresses for UM data fields, was performed by the user interface processing code. The control data generated by this was written to the STASH control (STASHC) file in the job library. At vn4.0, this processing functionality was transferred into the UM at the control level, and at the same time was modified in accordance with submodel partitioning requirements. Other improvements were also introduced, in particular a more flexible system for user-defined prognostics and diagnostics. This system is described in more detail in the next section.

The STASH-request processing and addressing code referred to above (known as STASH\_PROC) is called by the UM top-level control routine UMSHELL. The STASH routines themselves form a generic system which is called by each internal model section. The input to STASH comes either from the main data array D1 (either a prognostic or a diagnostic previously stored there by STASH) or from STASH's own work array STASHwork. STASHwork is dynamically allocated and reused between sections; its length is obtained from

from space requirements for each diagnostic computed by STASH\_PROC. Diagnostic output produced by a PCR is either stored in a local work array and later copied to STASHwork, or stored directly in STASHwork by the PCR, i.e., STASHwork is passed to the PCR as an argument. When STASHwork is used directly by the PCR (an "intercepted" diagnostic) the full vertical domain is assumed and no compression is allowed. For diagnostics copied from temporary workspace to STASHwork, vertical compression is allowed. Vertical compression may be done by the copying routine or within the PCR.

For climate runs, all diagnostics which cannot be derived from other quantities must be held in the dumps (produced at specified intervals), where the automatic meaning programs can access them. For forecast use, however, there is a requirement to produce post-processing ("pp") files, at more frequent intervals, for diagnostics which will not be in the dumps. It is most efficient to produce pp files for these quantities directly from the STASH routine. The STASH routine is therefore called not only after the PCRs which perform the integration of the model governing equations, but also after PCRs dedicated entirely to producing diagnostics.

The above account concentrates on diagnostics since they are arbitrarily variable and so more difficult to deal with than the regular quantities that must be passed between PCRs for the model code to run. In general the term "diagnostic" is used to refer to all output from a PCR. This may even include the primary model variables if they are to be processed by STASH. However the updating of primary fields will be hardwired into each PCR and will not be part of the STASH system.

## **2. UM Data Field Addressing and STASH Request Processing System**

### **2.1 Submodels and Data Partitions**

Version 4.1 of the Unified Model contains four "internal models". These are the atmosphere, ocean, slab ocean and ocean wave models. A "submodel" is a made up of one or more internal models which share a single data partition. To say that two internal models share a single data partition implies that the D1 array contains data fields belonging to both internal models at the same time. To illustrate this, suppose that internal models IM1 and IM2 share a data partition. The arrangement of fields in the D1 array is then as follows:

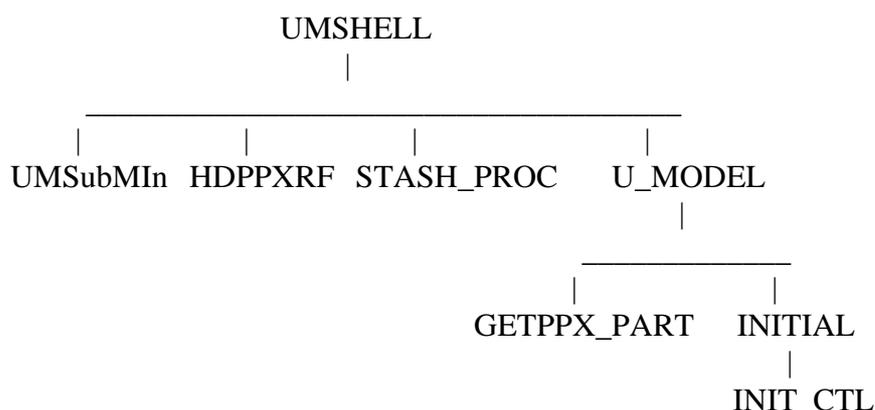
{ IM1 primary, IM2 primary, IM1 diag, IM2 diag, IM1 sec, IM2 sec }

where "primary" refers to prognostic and ancillary fields, "diag" to diagnostic fields, and "sec" to secondary fields. The lengths and start addresses of these fields are computed by the STASH addressing system during UM run time as part of the initial control stage of the run. The primary fields are the same as those in the reconfigured input dump (the reconfiguration program uses the same data addressing routines as the UM). The diagnostic and secondary fields depend on the STASH requests which the user sets up through the UMUI. The secondary fields are diagnostics which are not requested as output but which are to be passed from one UM section to another.

At UM vn4.1, there is only one case in which a submodel contains two internal models, namely the atmosphere/slab ocean case. In an atmos/slab model, the atmosphere and slab fields are arranged as in the above diagram. In an atmosphere/ocean coupled run, the

atmosphere and ocean fields are arranged as two separate data partitions, i.e. at any particular stage of the run, the D1 array contains either atmosphere or ocean fields but not both. At vn4.1, the wave model can only be run on its own, as facilities to couple it to other models do not yet exist.

## 2.2 Top Level Structure of STASH Addressing System



The top level control routine for the STASH request processing and addressing system is STASH\_PROC. The diagram above shows where STASH\_PROC fits into the UM system. Only the subroutines which are directly relevant to STASH addressing are shown. The UMSubModel\_Init routine sets up the arrays which define the submodel partitioning arrangements, i.e. they specify how many submodels there are in the run, and which internal models each submodel contains. HDPPXRF performs certain preliminary functions related to the STASHmasters (see below), then STASH\_PROC carries out the addressing of data fields and processing of STASH requests. The STASH control and addressing information generated by STASH\_PROC is held in a number of arrays. These arrays are passed to routine INITIAL (via COMMON blocks), and INITIAL calls INIT\_CTL, which transfers the STASH control and addressing information to arrays which are passed to the STASH system. The function of GETPPX\_PART will be described below.

## 2.3 STASHmaster files

Each internal model has its own "STASHmaster" file, which defines the characteristics of all the data fields associated with that internal model (Appendix 3). The STASHmasters are fixed for a particular UM release. The user may add extra STASHmaster records, or overwrite records in the STASHmasters, by defining his own "user STASHmasters", for a particular UM experiment. This facility enables the user to define additional prognostics and diagnostics, for example.

Each internal model is divided into processing sections, which can be regarded as master PCR's, each followed by a call to STASH. The STASHmasters are divided into corresponding sections. Within each section, the fields are identified by item numbers. Thus each STASHmaster field record is identified uniquely by the three numbers (internal\_model\_id, section\_no, item\_no). The following table defines the sections for each internal model.

Internal model	STASH section	Master PCR
Atmosphere	0	Primary fields (i.e. dump fields)
	1	Shortwave radiation
	2	Longwave radiation
	3	Boudary layer and surface
	4	Large-scale precipitation
	5	Convection
	6	Gravity wave drag
	7	Vertical diffusion
	8	Hydrology
	9	Cloud scheme
	10	Adjustment
	11	Tracer advection
	12	Primary field advection
	13	Diffusion and filtering
	14	Energy adjustments
	15	Extra dynamics related diagnostics
	16	Extra physics related diagnostics
	17	Sulphur cycle chemistry
	18	Atmospheric data assimilation
	19	Vegetation Distribution
	21-24	Atmospheric climate mean diagnostics, periods 1-4
Ocean	0	Primary fields
	30	General ocean diagnostics
	31	Vorticity related diagnostics
	32	Sea ice related diagnostics
	35	Oceanic assimilation
	41-44	Ocean climate mean diagnostics, periods 1-4
Slab ocean	0	Primary fields
	21-24	Slab climate mean diagnostics, periods 1-4
	40	Slab model diagnostics
Ocean Wave	0	Primary fields
	1	Advection
	2	Input source term fields
	3	Energy increments after nonlinear transfer
	4	Energy increments after dissipation
	5	Energy increments after bottom friction
	6	Diagnostic output
	7	Wave Assimilation

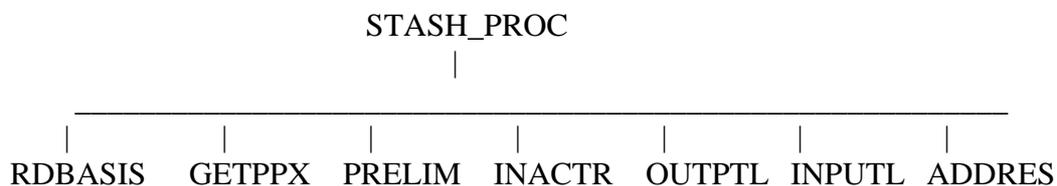
Routine HDPPXRF opens the required STASHmasters and determines how many fields each contains. It also checks how many user-defined STASHmaster records there are. The total number of STASHmaster fields is used to dynamically allocate the arrays into which the STASHmaster files are read.

STASH\_PROC calls routine GETPPX which reads the required STASHmasters into the dynamically allocated arrays. GETPPX also reads any user-defined STASH records from the appropriate job library files and inserts them into the STASHmaster arrays in the correct positions relative to the system STASHmaster records. This may include overwriting system records if required. A generic function EXPPX\_ is provided to extract individual elements of STASHmaster data from the STASHmaster arrays.

The STASHmaster data is used by the STASH\_PROC system, along with data from the job library files, to generate the addressing and STASH control data. When this is complete, it is then known which subset of the STASHmaster records for each internal model are actually required for the particular UM experiment. The total number of these records (including any user-defined records) is passed to U\_MODEL and used to dynamically re-allocate the STASHmaster arrays. Routine GETPPX\_PART is called by U\_MODEL to perform another read of the appropriate STASHmaster files into these arrays, including the user-defined records. These STASHmaster records are then available to the STASH routines etc., which use the EXPPX\_ function to extract data elements as required.

(Historical note: Up to UM vn4.0, there was only one STASHmaster, which held the records for all internal models. The UM did not read directly from this. Instead, the STASHmaster was used to generate the "ppxref file", which was just a direct-access equivalent of the STASHmaster. At vn4.1, the ppxref file no longer exists. However, references to "ppxref" remain in names of subroutines, functions, arrays, etc., although they are no longer appropriate. It is intended that these will be altered in future releases.)

The following diagram shows some of the main subroutines called by STASH\_PROC, which 'are referred to in this outline account.



#### 2.4 Primary Addressing and Reconfiguration

The computation of data lengths and start addresses for primary fields is carried out by routine ADDRES. This part of the STASH\_PROC system is used by the reconfiguration program to compute primary addressing for the reconfigured dump, and also by the UM itself for the D1 array.

Each STASHmaster record includes a "version mask" and an "option code". The version mask is a 20-digit binary integer string that defines which UM releases this STASH item is available to. The option code is a 20-digit decimal integer string which defines the availability of the STASH item within the internal model section to which it belongs (see appendix 3 for precise details). In general, a particular STASH item in a particular section will be available for certain versions of that section and not for others. A routine called TSTMSK is called by ADDRES to perform the checking of version masks and option codes, and so determine the list of primary items required for the particular UM experiment. (The same system is used to check availability of diagnostics). The list of section versions which TSTMSK checks the

option codes against is read in from the SIZES file in the job library.

As well as generating the list of primary items for the reconfigured dump, ADDRES also computes the data length and start address in D1 for each primary field, taking into account horizontal domains, vertical levels and pseudo-levels.

In general, the output dump produced by a UM run may contain non-primary fields in addition to the primary ones. This will be so if any of the diagnostic requests had the dump specified as their output destination, for example. However, none of these fields are available for input to reconfiguration or to another UM experiment. If the dump is reconfigured, for example, the dump output by the reconfiguration system would not contain any of the non-primary fields from the original dump.

## 2.5 STASH Request Processing

The processing of STASH requests and the associated addressing computations are rather complicated and will only be described in bare outline here.

The overall function of the STASH request processing system is to convert the raw STASH requests in the job library into STASH control data arrays, which are passed to the STASH system called by each UM section. This control data tells the STASH system how to process the output for each diagnostic in time and space, and where to send the output (dump, pp files, etc.). The principal STASH control array is the "STASHlist" (see appendix 2).

The STASH requests set up through the UMUI are written to the STASHC file in the job library (see appendix 1). In addition to the individual diagnostics selected from the list which the UMUI presents to the user, this may include user-defined diagnostics and STASHmacro requests. A STASHmacro is a standard set of diagnostics which the user may activate via choices in appropriate UMUI windows. Certain macros are associated with coupling of internal models. Each STASH request has associated time, domain, and usage profiles which are also in the STASHC file. Routine RDBASIS reads the STASHC file information into various arrays, and also reads additional information, such as the section/version specifications and settings of various switches, from the SIZES file. Routine GETPPX reads in the STASHmaster information (as explained above).

Initial processing of the STASH requests is performed by routine PRELIM, which generates the preliminary STASHlist array. Each STASH request gives rise to at least one STASHlist record. For some types of time processing, eg. meaning, there may be more than one STASHlist record (one for the partial sums and one for the final result), depending on what the processing periods are and what the output destination is. After the preliminary STASHlist has been set up, other routines are called to tidy it up in various ways.

The diagnostics which the user has selected through the UMUI, including any user-defined diagnostics, and together with any diagnostics arising from STASHmacros, constitute the "activated diagnostics". As the STASHlist is processed, other diagnostics may be added in; there are two categories of these, known as "inactive" and "implied" diagnostics. Inactive diagnostics are certain diagnostics for which space may need to be allocated, even if they are not required for output. Additionally, there are certain diagnostics which cannot be calculated unless certain other diagnostics are also calculated; if one of these has been requested, then

the diagnostics which are "implied" by it must also be added to the STASHlist. Routine INACTR checks the STASHmaster file for inactive and implied diagnostics, and adds them to the STASHlist if they are required.

The output and input data lengths for each diagnostic are calculated next. The output length is generally given by (no. of horizontal data points x no. of levels x no. of pseudo levels). (The only exceptions to this are in the case of a time series diagnostic request or if MPP processing is being used). Output lengths are calculated by routine OUTPTL. The input length is in many cases the same as the output length, but there are a number of cases where the two will differ. For example, the input to some diagnostics (the non-intercepted ones - see section 1) is allowed to be on a subset of the possible levels rather than on all the levels. Suppose that such a diagnostic is requested a number of times, each with different (and possibly overlapping) sets of output levels. In such a case, the input levels list would be the superset of all the output levels, with any overlap levels being counted once only. Each of these diagnostics would have an input length determined by the number of levels in the superset, and an output length determined by the number of levels in its particular output list. The same situation can occur for input and output pseudo levels. Routine INPUTL constructs the supersets and computes the input lengths. INPUTL also deals with diagnostics for which the input must be on all available pseudo levels; in such a case, the range of pseudo levels is supplied as a list in the same array that contains the supersets of pseudo levels.

After the diagnostic data lengths have been computed, routine ADDRES performs the addressing computations for the diagnostic space in D1 and the STASHwork array, including any secondary space which may be required. This is basically just a matter of adding up the lengths in the correct sequence to obtain the start address for each data field. Certain diagnostics are copies of section 0 prognostics; in such a case, the appropriate slot in the primary section of D1 is used for the diagnostic.

## **APPENDIX 1 STASH Control File (STASHC)**

The STASHC file is generated by the user interface and forms part of the job library. It contains the information which the user has provided through the UMUI STASH panels. The file is in namelist format. There is a block of namelists for each internal model included in the UM experiment, and a block for each STASH macro which is activated. The namelists in each block specify the details of the STASH diagnostic requests for a particular internal model or macro, together with their associated time, domain and useage profiles. the following is a definition of the namelists and their contents in the order in which they appear in each block.

**NAMELIST/STASHNUM/** Appears once in each block

NUM\_REQ No. of STASH requests in this block  
NUM\_DOM No. of domain profiles in this block  
NUM\_TIM No. of time profiles in this block  
NUM\_USE No. of useage profiles in this block

**NAMELIST/STREQ/** Each STREQ defines a STASH diagnostic request

IMOD Internal model identifier  
ISEC Section number  
ITEM Item number  
IDOM Domain profile number  
ITIM Time profile number  
IUSE Useage profile number

The domain, time and useage profile numbers are sequence numbers for this block, not for the whole file.

**NAMELIST/TIME/** Each TIME defines a time profile.

ITYP Time processing code. Specifies how the diagnostic is to be processed. Has the following values:

- 0 Not required by STASH, but space required.
- 1 Replace
- 2 Accumulate
- 3 Time mean
- 4 Append timeseries
- 5 Maximum
- 6 Minimum
- 7 Trajectories

Option 0 is used to turn off diagnostics which are not required for a particular run but for which work space is required within a PCR.

Option 2 accumulates the output from a diagnostic over model timesteps. Zeroing takes place at times defined by entries 4&5.

Options 4&7 increase the size of a field with time.

ISAM Sampling frequency for ITYP=2-7 options.  
UNT2 Unit for sampling frequency (DA days, H hours,  
DU dump periods, T time steps).

INTV Processing period for options 2-7 (-1 for indefinite).  
UNT1 Unit for processing period.

IOPT Switch for output times option:  
1 Regular output times. Specify ISTR,IEND,IFRE -  
start time, end time, frequency.  
2 List of output times at irregular intervals. Specify  
ITIMES=no. of output times, ISER=list of times.

UNT3 Unit for output times.

**NAMELIST/DOMAIN/** Each DOMAIN specifies a domain profile.

IOPL Level type code:

1 Model full levels	6 Deep soil levels
2 Model half levels	7 Theta levels
3 Pressure levels	8 Potential vorticity levels
4 Geometric height levels	9 Cloud threshold levels
5 Single level	10 Wave model direction "levels"

For IOPL=1,2,6,10 ("model" levels) specify ILEVS:

ILEVS=1 Contiguous range of model levels defined by  
LEVB,LEVT - first & last level.

ILEVS=2 Integer list (LEVLST) of specified levels.

IOPA Horizontal domain type.

1 Global	6 0-30 N
2 N hemisphere	7 0-30 S
3 S hemisphere	8 30S-30N
4 30-90 N	9 Area specified in whole degrees
5 30-90 S	10 Area specified in gridpoints

For IOPA=9,10 specify area limits INTH,ISTH,UEST,IWST

IMSK Gridpoint option

- 1 All points
- 2 Land points only
- 3 Sea points only

IMN Spatial meaning option

0 None	3 Meridional
1 Vertical	4 Horizontal area
2 Zonal	

IWT Weighting option

0 None	2 Volume
1 Horizontal	3 Mass

- PLT Pseudo level type
- 0 None
  - 1 SW radiation bands
  - 2 LW radiation bands
  - 3 Atmospheric assimilation groups
  - 4 Ocean assimilation groups
  - 5 Ocean tracer transport (MEAD) diagnostics
  - 6 Wave model frequency
  - 7 Wave model wave train number
  - 8 HadCM2 Sulphate Loading Pattern Index
  - 9 Land and Vegetation Surface Types

Pseudo levels are specified by an integer pseudo-levels list PSLIST. There is as yet no "contiguous range" facility allowing them to be specified by first & last level only.

TS Switch for time series option (y/n). If TS=Y, the time series domains are defined by the following arrays:

TSNUM	No. of time series domains
TNLIM,TSLIM,TELM,TWLIM	Horizontal domain limits
TBLIM ,TTLIM	Vertical limits (model levels)
TBLIMR,TTLIMR	Vertical limits (real levels)

**NAMELIST/USE/** Each USE specifies a useage profile.

LOCN Output destination code

- 1 Dump store with user specified tag
- 2 Dump store with climate mean tag
- 3 PP file
- 5 Mean diagnostic direct to mean PP file
- 6 Secondary dump store with user tag

IUNT PP unit number (for LOCN=3)

Tag (for LOCN=1,2,6)

The STASH request blocks are terminated by a STASHNUM namelist with NUM\_REQ=-1.

**NAMELIST/USTSNUM/** appears after all the STASH request blocks. It specifies user-defined STASHmaster files.

N_USTASH	No. of user STASHmaster files
NRECS_USTASH	Total number of user STASH records
USTSFILS	Names of user STASH files

Via the UMUI, the user may specify any number of user STASHmaster files for each internal model. For each internal model, the umui concatenates the STASHmasters into a single file - PRESM\_A, PRESM\_O, PRESM\_S, PRESM\_W for atmos, ocean, slab, & wave. Note that the PRESM\_ file always appears in the job library for each internal model included in the experiment, even if it is empty. (The name PRESM denotes "pre-STASHmaster" - this name is a residual from the pre-vn4.0 UM, and is no longer meaningful. It should be changed in a future release.)

## **APPENDIX 2 STASH list array and other STASH arrays**

**STASH list array LIST\_S** is generated by the STASH\_PROC system and holds the processed STASH requests. All requests which are either active or require workspace are held. Some are duplicated if they are to be processed in more than one way. Each row of LIST\_S is a single processed STASH request and contains 28 entries. Each STASH request is identified uniquely by entries 1,2, & 28 (item,section,internal model id).

The entries in each row of LIST\_S are as follows:

LIST\_S entry

1	Item number
2	Section number
3	Processing code
4	Frequency
5	Start timestep
6	End timestep
7	Period
8	Gridpoint code
9	Weighting code
10	Input bottom level
11	Input top level
12	Northern row
13	Southern row
14	Western column
15	Eastern column
16	Input code
17	Input length
18	Output code
19	Output length in D1
20	Output address in D1
21	Output first level
22	Output last level
23	Position of PP header for field in D1
24	Pointer for time series
25	STASHlist tag
26	Input pseudo level list pointer
27	Output pseudo level list pointer
28	Internal model identifier
29	Position of item in D1 array for relevant sub model
30	Offset for sampling
31	Output address in dump (MPP jobs only)
32	Output length in dump (MPP jobs only)
33	Output length of a single level on dump (MPP jobs only)

An explanation of the above entries now follows.

### **ENTRY 3 Processing code**

Specifies how the diagnostic is to be processed.

- 0 Not required by STASH, but space required.
- 1 Replace
- 2 Accumulate
- 3 Time mean
- 4 Append timeseries
- 5 Maximum
- 6 Minimum
- 7 Trajectories

**ENTRY 4** Frequency (input and output)

Specifies the frequency at which processing occurs, in timesteps. The umui allows this to be specified in a variety of units (hours, dump periods, etc.), and STASH\_PROC converts it to timesteps. Note that a diagnostic can be processed only at those timesteps at which the appropriate PCR is called, and that processing will occur only when the calling and processing timesteps coincide. Eg. if the LW radiation PCR is called every 12 timesteps and the processing frequency is set to 8 time steps (both starting at 0 timesteps), then no processing will occur until 24 timesteps. The UMUI ensures that processing of diagnostics from a section can only be requested at those timesteps at which the section is called.

A value of -n in entry 4 indicates that processing is required at the timesteps specified in the STASH output times table STASHtimes(time,n). Entries 5-7 are then ignored.

**ENTRIES 5 & 6** Start & End Timesteps

Specifies the timesteps at which processing starts/ends. The UMUI allows convenient time units to be used. STASH\_PROC converts to timesteps.

**ENTRY 7** Period

Specifies the number of timesteps in the period for the accumulating and meaning options, and for the recycling of the appending options, referred to in entry 3. May be specified as in entries 5&6. For the appending options this entry defines the length of the timeseries or trajectory dataset allocated.

**ENTRY 8** Gridpoint code

Specifies the gridpoints at which processing will occur.

- 1 All points
- 2 Land points (MDI for sea)
- 3 Sea points (MDI for land)
- 1m Vertical mean over domain specified by entries 12-15,21-22 (m=1,2,3)
- 2m Zonal mean over levels specified by entries 21,22
- 3m Meridional mean over levels specified by entries 21,22
- 4m Field mean over domain specified by entries 12-15,21-22

Thus, options 1,2,3 allow the simple extraction of a sub-field of the diagnostic, as defined by entries 12-15, while entries 1m-4m allow that sub field to be meaned in various ways. Note that these fields will be output over the full area and padded with missing data indicators if the data is not available for the full model domain. Zonal and meridional means will always use the full model area.

**ENTRY 9** Weighting code

- |                             |                    |
|-----------------------------|--------------------|
| 0 None                      | 2 Volume weighting |
| 1 Horizontal area weighting | 3 Mass weighting   |

**ENTRY 10** Input first level

Specifies the type of input level selection.

100 Indicates a single special level

-N Points to levels list N

M Input on range of model levels starting at level M

**ENTRY 11** Input last level

Definition depends on entry 10.

If entry 10 negative then:

ENTRY 11 Levels list contains

- 1 model levels
- 2 pressure levels
- 3 heights
- 4 theta levels
- 5 PV levels
- 6 Cloud threshold levels

If entry 10 = 100 then ENTRY 11 contains vertical level code LBVC

If entry 10 = M then ENTRY 11 indicates last input model level

**ENTRIES 12-15** N/S rows, W/E columns

Define the horizontal domain over which the diagnostic is processed. The UMUI presents a number of preset area options (eg. 30N to 90N) and also allows the area to be specified in degrees lat/lon or row/column numbers. For the ELF grid the lat/lon refers to the transformed coordinate system. If the lat/lon specified does not coincide with model points then the smallest area enclosing that specified will be computed.

**ENTRY 16** Input code

Specifies the source of the input to STASH.

0 Use primary field at D1(SI(item,section,model))

1 Use space at index SI(item,section,model)

-j Use diagnostic at D1(LIST\_S(20,j))

The STASHmaster item number within each section references output diagnostics in the PCR CALL argument. If the diagnostic is a primary field then D1(SI(item,section,model)) will be passed across in the PCR CALL. If the diagnostic is transitory and calculated within workspace then STASHWORK(SI(item,section,model)) will be passed across. SI(i,s,m) will generally point to workspace. If no workspace is required at all then the SI(i,s,m) will point to STASHWORK(1) to avoid allocating workspace that is not used. Option 0 allows a primary variable to be processed by STASH. Option -j allows a diagnostic that has already been processed into D1(LIST\_S(20,j)) to be reprocessed by STASH.

**ENTRY 17** Input length

The length of the diagnostic before STASH processing. This is computed by STASH\_PROC.

**ENTRY 18** Output code

Defines the output destination for the diagnostic. Has the following values:

- 1 Dump store
- 2 Secondary dump store
- nn Output to fortran unit nn

**ENTRY 19** Output length

The length of the diagnostic after STASH processing. Computed by STASH\_PROC. Output length will differ from input length if (for example) the levels on which output is requested are fewer than the levels on which the diagnostic is input. For MPP jobs, it will be the length of the diagnostic for on that PE.

**ENTRY 20** Output address

Start location of the diagnostic field in D1 after STASH processing. Used by STASH to locate diagnostics in the dump which require reprocessing. In MPP jobs, it will be the start location in the D1 array containing the data for that PE.

**ENTRY 21** Output first level

Specifies type of level selection required. Has the following values:

- N Points to levels list N
- M Output on a range of model levels starting at level M

**ENTRY 22** Output last level

Definition depends on entry 21.

If entry 21 negative then:

- Entry 22 =1 Model levels
- =2 Pressure levels
- =3 Heights
- =4 Theta levels
- =5 PV levels
- =6 Cloud threshold levels

If entry 21 =M then

Entry 22 indicates last output model level

**ENTRY 24** Pointer for time series

References an entry in STASH\_SERIES\_INDEX which in turn references a set of subdomain records in STASH\_SERIES.

**ENTRY 25** STASHlist tag

Given by  $T_g + 1000 * T_m$  where  $T_g$ =user tag,  $T_m$ =time mean tag.  $T_m$  is defined by  $T_m = M1 + 2 * M2 + 4 * M3 + 8 * M4$  where  $Mn=0$  if the diagnostic is not required on a climate mean PP

file. Mn=1 if the climate mean is required for period n.

**ENTRY 26** Input pseudo dimension list

Points to the list of pseudo dimension indices defining the pseudo levels on which the field will be passed to STASH. This will be the superset of all the output pseudo levels for that item.

**ENTRY 27** Output pseudo dimension list

Points to the list of pseudo dimension indices requested for output from the diagnostic.

**ENTRY 28** Internal model id.

Atmosphere 1, Ocean 2 , Slab Ocean 3, Wave 4

**ENTRY 29** Position of item in D1 array for relevant sub model.

The D1\_ADDR array contains information relating to all items in the D1 array dimensioned by submodel, whereas the STASH list includes information about diagnostics only, and does not dimension by submodel. Entry 29 is a pointer to the records relating to the STASH item in the D1\_ADDR array. For example, D1\_ADDR(d1\_address,STLIST(29,K)) gives the address in D1 for STASH item K which, in MPP runs, differ from the dump address.

Appendix 5 contains information on the D1\_ADDR array in the UM.

**ENTRY 30** Offset for sampling.

When the sampling frequency is more than one time unit, it is possible to specify an offset. For example, if sampling every 2 timesteps, then with a zero offset, timesteps 1,3,5,7... are sampled. With an offset of 1, timesteps 2,4,6,8.. are sampled.

**ENTRY 31** Output address in dump (in MPP jobs only)

Start location of the diagnostic field in the model dump.

**ENTRY 32** Output length in dump (in MPP jobs only)

The length of the diagnostic in the dump. For multi-level diagnostics, this will be the total length of the diagnostic. Note that for MPP jobs, this length represents the full model grid and not the grid for the individual PE (see Entry 19)

**ENTRY 33** Output Length of a single level in dump (in MPP jobs only)

See Entry 32. This is the length for a single level of the diagnostic. For single level diagnostics, entries 32 and 33 will be the same.

The model uses the STASHlist entries to produce a "pp-header" for each diagnostic which is processed to the dump or pp files. If the diagnostic exists at multiple levels then after processing it will be stored as a number of fields, each with its own pp header.

## Other Arrays Generated by STASH\_PROC

(a) STASHtimes(time\_index,table\_index)

The tables of processing times (input & output) for diagnostics which have a negative value in entry 4 of the STASHlist record.

(b) STASH\_LEVELS(NUM\_STASH\_LEVELS+1,NUM\_LEVELS\_LIST).

The levels lists are held in the 2-d STASH\_LEVELS array. The first element in each column is the no. of levels in that list. The array is of type integer. Real values are multiplied by 1000 so allowing for an accuracy of 3 decimal places.

(c) STASH\_PSEUDO\_LEVELS(levels\_index,lists\_index)

These are the lists of indices for pseudo-levels, which may be requested for any STASH item which has a pseudo-dimension (eg. radiation bands).

(d) STASH\_SERIES\_INDEX(2,NSTASH\_SERIES\_BLOCK)

STASH\_SERIES(TIME\_SERIES\_RECORD\_LEN,NSTASH\_SERIES\_RECORDS)

An index to, and table of, subdomain records defining the sampling regions used for averaging when extracting time series data from a field. Sampling regions may have up to 3 dimensions.

(e) STASHindex(2,item,section,internal\_model\_index)

Gives the starting index and number of indices for each STASH item. It is a pointer to the STASHlist.

### APPENDIX 3. STASHmaster FILES

There is one STASHmaster file for each internal model. Each STASHmaster consists of a series of records, one for each primary or diagnostic field, ordered according to section and item number. Each record gives a complete definition of the characteristics of the field to which it refers. The STASHmasters are used by the umui, reconfiguration, and the UM itself. The template for STASHmaster records is as follows:

```
#|Model |Sectn | Item |Name |
#|Space |Point | Time | Grid |LevelT|LevelF|LevelL|PseudT|PseudF|PseudL|LevCom|
#| Option Codes | Version Mask |
#|DataT |DumpP | PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PCA |
#|Rotate| PPFC | USER | LBVC | BLEV | TLEV |RBLEVV| CFLL | CFFF |
```

**Model, Section, Item** identify the record.

Model=1, atmosphere; 2, ocean; 3, slab ocean; 4, wave.

Section numbers are in the range 0-44. Section 0 is for primary fields.

Item numbers are in the range 1-512.

**Name** is a 36-character description of the STASH item.

**Space** The space code.

Specifies the space requirements of a STASH item. If a diagnostic is calculated only when required by STASH (i.e., under a STASH flag), or is only copied to STASHwork under a STASH flag, then the space code is 0. If a section uses STASHwork space directly without STASHflag protection, then 0 is not appropriate - a space code of 1 is more likely. The possible values of space code are:

- 0 Diagnostic field for which space is required only when the diagnostic is active.
- 1 Non-prognostic field for which space is always required.
- 2 Section 0 items only: primary field available to STASH
- 3 Section 0 items only: primary field unavailable to STASH which is addressed in the dump and D1.
- 4 Section 0 items only: primary field which is addressed in D1 only.
- 5 Section 0 items only: primary field which is addressed in the dump only.
- 6 Diagnostic field for which space is required when active or implied by other diagnostics.
- 7 Non-primary field which points back to a section 0 field.
- 8 Dual time level ocean prognostic
- 9 Extra Items at the end of D1 for internal fields such as PEXNER
- 10 Field not held in D1 or the dump

**Point** Section zero point-back.

This is used for non-section 0 items with space code 7. Any such field is a copy of a primary field. The value of "point" is the section 0 item number which it is a copy of.

**Time** Time availability code.

Specifies at which time steps the diagnostic is available.

- 1 Every time step.
- 2 Every long wave radiation timestep.
- 3 Every short wave radiation time step.
- 4 Every coupling period.

- 5 Every call of atmospheric climate meaning period 1.
- 6 Every call of atmospheric climate meaning period 2.
- 7 Every call of atmospheric climate meaning period 3.
- 8 Every call of atmospheric climate meaning period 4.
- 9 Every call of oceanic climate meaning period 1.
- 10 Every call of oceanic climate meaning period 2.
- 11 Every call of oceanic climate meaning period 3.
- 12 Every call of oceanic climate meaning period 4.
- 13 Every convection timestep.
- 14 Every leaf phenology timestep
- 15 Every vegetation competition timestep.

**Grid** Grid type code

Specifies the type of grid on which the field is defined.

- 1 Data on atmospheric theta points.
- 2 Data on atmospheric theta points, values over land only.
- 3 Data on atmospheric theta points, values over sea only.
- 4 Data on atmospheric zonal theta points.
- 5 Data on atmospheric meridional theta points.
- 11 Data on atmospheric uv points.
- 12 Data on atmospheric uv points, values over land only.
- 13 Data on atmospheric uv points, values over sea only.
- 14 Data on atmospheric zonal uv points.
- 15 Data on atmospheric meridional uv points.
- 17 Atmospheric scalar.
- 18 Data on atmospheric u points on the 'c' grid.
- 19 Data on atmospheric v points on the 'c' grid.
- 21 Data compressed to atmospheric land points.
- 22 Data on the ozone grid.
- 25 Data on the atmospheric LBC grid.
- 31 Data on 3D compressed ocean mass points.
- 32 Data on 3D compressed ocean velocity points.
- 36 Data on ocean mass points with optional cyclic columns.
- 37 Data on ocean velocity points with optional cyclic columns.
- 38 Data on oceanic u points on the 'c' grid with optional cyclic columns.
- 39 Data on oceanic v points on the 'c' grid with optional cyclic columns.
- 41 Data on ocean mass points.
- 42 Data on ocean velocity points.
- 43 Data on ocean zonal non-cyclic mass points.
- 44 Data on ocean zonal non-cyclic velocity points.
- 45 Data on ocean meridional non-cyclic mass points.
- 46 Data on ocean meridional non-cyclic velocity points.
- 47 Data on ocean scalar points.
- 51 Data on the ocean LBC grid.
- 60 Data on wave model full grid
- 62 Data on wave model grid, sea points only
- 65 Data on wave model LBC grid

**LevelT.** The level type code.

Specifies the type of levels on which a field is defined.

- 0 Data on many levels, but stored as one array (eg. lateral boundary conditions)
- 1 Data on model full levels.
- 2 Data on model half levels.
- 3 Data on pressure levels.
- 4 Data on geometric height levels.
- 5 Single level data.
- 6 Data on deep soil levels.
- 7 Data on theta levels.
- 8 Data on potential vorticity levels (ascending order).
- 9 Data on cloud threshold levels (octars).
- 10 Data on wave model directions.

**LevelF** and **LevelL**. The first and last level codes.

Specify the first and last levels on which a field is defined.

- 1 Unset, this code must be used for single level data.
- 1 First atmospheric level.
- 2 Top atmospheric level.
- 3 Top wet level.
- 4 Top atmospheric level-1.
- 5 1st level in boundary layer.
- 6 Last level in boundary layer.
- 7 1st level above boundary layer.
- 8 1st soil level.
- 9 Last soil level.
- 10 1st tracer level.
- 11 Last tracer level.
- 12 Last gravity wave drag level +1.
- 13 1st gravity wave drag level.
- 14 Last gravity wave drag level.
- 15 1st level in vertical diffusion routine.
- 16 Last level in vertical diffusion routine-1.
- 17 Last level in vertical diffusion routine.
- 18 Last level in boundary layer -1.
- 19 Top level of atmosphere +1.
- 20 First soil level+1.
- 21 First ocean level (top).
- 22 Last ocean level (bottom).
- 23 Top ozone level.
- 24 Number of atmos levels \* sw bands.
- 25 (number of atmos levels+1) \* sw bands.
- 26 (number of wet levels)\* sw bands.
- 27 Number of atmos levels \* lw bands.
- 28 (number of atmos levels+1) \* lw bands.
- 29 (number of wet levels)\* lw bands.
- 30 First ocean level +1.
- 31 Number of ocean assimilation levels.
- 32 Number of SW radiation bands.
- 33 Number of LW radiation bands.
- 34 Number of soil hydrology levels.
- 35 Number of cloudy levels.

- 36 First wave model direction.
- 37 Last wave model direction.

Note that most derived levels (such as X\* sw bands) are redundant - pseudo levels may be used instead.

**PseudT.** Pseudo level type.

This code defines the type of pseudo level that the diagnostic is defined on.

- 0 None
- 1 SW radiation bands.
- 2 LW radiation bands.
- 3 Atmospheric assimilation groups.
- 4 Oceanic assimilation groups.
- 5 Ocean tracer transport (MEAD) diagnostics.
- 6 Wave model frequencies.
- 7 Wave model wave-train number
- 8 HadCM2 Sulphate Loading Pattern Index
- 9 Land and Vegetation Surface Types

**PseudF.** First pseudo level code.

Defines the lower bound of the possible values of pseudo level for a field.

- 1 Dimension starts at 1.
- 20+n Dimension starts at the first group for the atmospheric assimilation set n.  
Sets are:
  - 1 P\* observation type.
  - 2 Theta observation types.
  - 3 U,V observation types.
  - 4 Relative Humidity observation types.
  - 5 Rainfall rate observation types.
- 40+n Dimension starts at the first group for the oceanic assimilation set n.

**PseudL.** Last pseudo level code.

Defines the upper bound of possible values of the pseudo level for a field.

- 1 The number of short wave bands.
- 2 The number of long wave bands.
- 3 (Number of mead\_types)\*(Number of ocean basins).
- 4 Highest wave frequency (wave model).
- 5 Highest wave train number (wave model).
- 6 The number of sulphate loading patterns
- 7 The total number of surface types.
- 8 The number of surface types that are vegetation.
- 9 The number of surface types that are not ice.
- 20+n The last group for the atmospheric assimilation set n
- 40+n The last group for the oceanic assimilation set n.

The values of the upper bounds can be found in the UM deck PSLIMS1.

**LevCom.** The level compression flag.

This code defines which of the available levels are passed as input to STASH.

- 0 STASH receives input on all available levels and

pseudo-levels.

- 1 STASH receives the diagnostic input on the STASH input levels list. Compression is performed in either the routine or the interfacing. A flag of 1 must be used for multi-level items not on model levels.

### **Option Code. The sectional option code.**

The option code is a 20-digit decimal code which defines under what conditions an item is unavailable to STASH. Each STASH section has its own definitions for the option codes. The digits are labelled n20,n19,...,n1. The rules for interpretation of the option codes are as follows.

#### **Atmosphere & Slab section 0.**

- n2n1 The tracer number.
  - Item included only if the n2n1 tracer switch is set.
- n3 Hydrology indicator.
  - 0 Item not dependent on hydrology scheme.
  - 1 Item included only for single-layer hydrology.
  - 2 Item included only for multi-layer hydrology. (Pre Vn 4.5)
  - 3 Item included only for moses hydrology.
  - 4 Item excluded from single-layer hydrology.
- n4 Stratosphere indicator.
  - 0 Item included in the stratosphere model.
  - 1 Item excluded from the stratosphere model.
- n5 Orographic roughness indicator.
  - 0 Item not dependent on orographic roughness.
  - 1 Item excluded when orographic roughness used.
  - 2 Item included only when orographic roughness used.
- n6 Interface indicator.
  - 0 Item not dependent on interface.
  - 1 Item included only in limited area models.
  - 2 Item included only for models with lower boundary.
- n7 Coupling indicator.
  - 0 Item not dependent on coupling.
  - 1 Item included only if slab model included.
  - 2 Item included only if ocean model included.
  - 3 Item included only for slab calibration runs.
  - 4 Item included only for slab non-calibration runs.
  - 5 Item included only for slab runs with Ice Dynamics
- n8 Extra fields indicator.
  - 0 Item not dependent on extra fields.
  - 1 Item included only for SST anomaly runs.
  - 4 Item included only for total aerosol emission runs.
  - 5 Item included only for total aerosol emission runs.
  - 6 Item included only if snow albedo scheme used.
  - 7 Item included only if HadCM2 approximate radiative effects of sulphate used
  - 8 Item included only if Energy Adjustment Scheme (section 14) used.
- n9 Orographic gradient component indicator.
  - 0 Item not dependent on orographic gradient.
  - 1 Item included only if orographic gradient used (GWD).

- n10 Sulphur cycle indicator. (From UM Vn 4.1)
  - 0 Item not dependent on sulphur cycle scheme.
  - 1 Item included only for SO<sub>2</sub> sulphur cycle.
  - 2 Item included only for SO<sub>2</sub> with surface emissions.
  - 3 Item included only for SO<sub>2</sub> with high level emissions.
  - 4 Item included only for SO<sub>2</sub> with natural emissions.
  - 5 Item included only for SO<sub>2</sub> with DMS cycle.
  - 6 Item included only for SO<sub>2</sub> with DMS cycle and emissions.
  - 7 Item included for SO<sub>2</sub> with O<sub>3</sub> oxidation included.
  - 8 Item included for SO<sub>2</sub> with O<sub>3</sub> oxidation and NH<sub>3</sub> included.
  - 9 Item included only for SO<sub>2</sub> with O<sub>3</sub> oxidation and NH<sub>3</sub> emissions.
- n11 Vegetation parametrization indicator. (From UM Vn 4.4)
  - 0 Item not dependent on vegetation parametrization.
  - 1 Item included only if direct vegetation parametrization not used.
  - 2 Item included only for direct vegetation parametrization with or without competition
  - 3 Item included only for direct vegetation parametrization with competition.
- n12 Mixed phase precipitation (From UM Vn 4.4)
  - 0 Item not dependent on Mixed phase Precipitation.
  - 1 Item dependent on Mixed phase precipitation.
- n13 2D or 3D convective cloud amount indicator. (From UM Vn 4.4)
  - 0 Item not dependent on convective cloud.
  - 1 Item included only if CCA is 2D (anvil code OFF)
  - 2 Item included only if CCA is 3D (anvil code ON)
- n14 Soot scheme indicator (From UM Vn 4.5)
  - 0 Item not dependent on soot scheme.
  - 1 Item included for soot scheme only.
  - 2 Item included for soot scheme with surface emissions.
  - 3 Item included for soot scheme with high level emissions.
- n15 RHcrit paramterization indicator. (From UM Vn 4.5)
  - 0 Item not dependent on RHcrit parametrization.
  - 1 Item available only if RHcrit parametrization used.
- n16 Carbon cycle indicator. (From UM Vn 4.5)
  - 0 Item not dependent on carbon cycle scheme.
  - 1 Item included only for carbon cycle.

Atmosphere section 1 - Shortwave Radiation.

- n1 Global model indicator.
  - 0 Item not dependent on global/LAM option
  - 1 Item available only for global atmosphere.
- n2 Cloud Microphysics indicator.
  - 0 Item not dependent on cloud microphysics.
  - 1 Item available only if cloud microphysics used.
- n3 HadCM2 approximate radiative effects of sulphate indicator
  - 0 Item not dependent on HadCM2 approximate radiative effects of sulphate
  - 1 Item available only if HadCM2 approximate radiative effects of sulphate used
- n4 Indirect effect of sulphate aerosol (SA) in SW Radiation
  - 0 Item not dependent on Indirect effect of SA in SW radiation
  - 1 Item available only if Indirect effect of SA in SW radiation requested

Atmosphere section 3 - Boundary Layer.

- n1 Orographic roughness indicator.
  - 0 Item not dependent on orographic roughness.
  - 1 Item available only if orographic roughness is used.
- n2 SO2 sulphur cycle/NH3/Soot indicator.
  - 0 Item not dependent on sulphur cycle or soot scheme
  - 1 Item available only if SO2 sulphur cycle used.
  - 2 Item available only if NH3 scheme used
  - 3 Item available only if soot scheme used
- n3 Carbon cycle indicator.
  - 0 Item not dependent on carbon cycle.
  - 1 Item included only for carbon cycle used.

Atmosphere section 4 - Large scale precipitation.

- n2 SO2 sulphur cycle/NH3/Soot indicator.
  - 0 Item not dependent on sulphur cycle or soot scheme
  - 1 Item available only if SO2 sulphur cycle used.
  - 2 Item available only if NH3 scheme used
  - 3 Item available only if soot scheme used

Atmosphere section 5 - Convection.

- n2 SO2 sulphur cycle/NH3/Soot indicator.
  - 0 Item not dependent on sulphur cycle or soot scheme
  - 1 Item available only if SO2 sulphur cycle used.
  - 2 Item available only if NH3 scheme used
  - 3 Item available only if soot scheme used
- n3 2D or 3D convective cloud amount indicator
  - 0 Item not dependent on convective cloud.
  - 1 Item included only if CCA is 2D (anvil code OFF)
  - 2 Item included only if CCA is 3D (anvil code ON)

Atmosphere section 9 - Cloud Parametrization

- n2 Cloud area parametrization indicator.
  - 0 Item not dependent on cloud area parametrization.
  - 1 Item available only if cloud area parametrization used.
- n3 RHcrit parametrization indicator.
  - 0 Item not dependent on RHcrit parametrization.
  - 1 Item available only if RHcrit parametrization used.

Atmosphere section 16 - Extra physics

- n2n1 The tracer number.
  - Item available only when the n2n1 tracer switch is set.

Atmosphere section 17 - Chemistry section.

- n1 Sulphur cycle or soot scheme indicator.
  - 0 Item not dependent on sulphur cycle or soot scheme.
  - 1 Item included only for sulphur cycle.
  - 2 Item included only for soot scheme.
- n2 Sulphur cycle options indicator
  - 0 Item always included in sulphur cycle.

- 1 Item included only if DMS included.
- 2 Item included only if O3 oxidation included.

#### Atmosphere section 18 - Data Assimilation

- n1 Assimilation set indicator.  
Item available only if set n1 activated.
- n2 Aerosol indicator.  
0 Item not dependent on aerosol emissions.  
1 Item available only if aerosol emissions included.

#### Atmosphere sections 21-24 - Climate mean diagnostics

Any item in section 20+n is available only if the nth. climate mean period is activated. Otherwise items in these sections are selected as in section 0.

#### Ocean section 0.

- n2n1 The tracer number.  
Item included only if the n2n1 tracer switch is set.
- n3 Barotropic indicator.  
0 Item not dependent on barotropic option.  
1 Item included for the standard rigid-lid barotropic solution.  
2 Item included for the free-surface barotropic solution.  
3 Not used.  
4 Item included if conjugate-gradient solution activated.
- n4 Ice model indicator.  
0 Item not dependent on ice model.  
1 Item included for the prognostic sea-ice model.  
2 Item included for the prognostic sea-ice model driven by surface air temperature and solar radiation only.  
3 Item included for the prognostic sea-ice model driven by fluxes as from an atmosphere model (coupled or ancillary).  
4 Item included for the prognostic sea-ice model with ice advected by the ocean surface currents.  
5 Item included for the prognostic sea-ice model with ice velocity calculated using the free-drift scheme.  
6 Item included for the prognostic sea-ice model with some kind of sea-ice dynamics.
- n5 Mixing and diffusion indicator.  
0 Item not dependent on mixing or diffusion.  
1 Item included for the Kraus-Turner mixed layer.  
2 Item included for the Kraus-Turner mixed layer with Richardson-number vertical diffusion.  
3 Item included for the Kraus-Turner mixed layer with basic vertical diffusion.  
4 Item included for the Gent-McWilliams scheme.  
5 Item included for the Gent-McWilliams scheme with the Visbeck scheme.  
6 Item included for the Richardson-number vertical diffusion scheme.  
7 Item included for the quadratic Large scheme.
- n6 Constraint on climatology.  
0 Item not dependent on climatology constraint.

- 1 Item included only if pseudo-ice is included or Haney forcing is applied.
- 2 Item included only if flux correction of heat is applied.
- 3 Item included only if full Haney forcing applied (including forcing of sea-ice depth)
- 4 Item included only if flux correction of heat is applied.
- n7 Lateral boundary conditions indicator.
  - 0 Item not dependent on lateral bc's.
  - 1 Item included only for LAM model with cox 1,2,3 or 4.
- n8 River runoff indicator.
  - 0 Item not dependent on river runoff.
  - 1 Item included only if river runoff activated.
- n9 General diagnostic indicator.
  - 0 Item not dependent on general diagnostics.
  - 1 Item included only if MEAD diagnostic on.
- n10 Carbon cycle indicator.
  - 0 Item not dependent on carbon cycle model.
  - 1 Item included only if carbon cycle model activated.

Ocean sections 30-32 - Ocean diagnostics.  
Set as for ocean section 0.

Ocean section 35 - Ocean assimilation

- n1 Group indicator.
  - 0 Item not dependent on assimilation group.
  - N Item available only when assimilating group N.
- n2 As for ocean section 0. ??????
- n3 As for ocean section 0.

Ocean sections 41-44 - Ocean climate mean diagnostics.

Any item in section 40+n is available only if the nth. climate mean period is activated. Otherwise items in these sections are selected as in ocean section 0.

Slab section 40 - Slab diagnostics.

- n1 Slab calibration indicator.
  - 0 Item not dependent on slab calibration.
  - 1 Item only available for slab calibration runs.
  - 2 Item only available for slab non-calibration runs.
- n2 Slab ice model indicator.
  - 0 Item not dependent on ice model.
  - 1 Item available only for ice advection or cavitating fluid model
  - 2 Item available only for ice advection model.
  - 3 Item available only for cavitating fluid model.
- n3 Ice thermodynamics indicator.
  - 0 Item not dependent on ice thermodynamics.
  - 1 Item available only for ice thermodynamics model.

### Version Mask.

A 20 digit binary code. Each STASH section can have up to 20 versions, and each version uses some subset of the item numbers in that section. Version 0 of a section is the null

version, i.e., that section is not activated, so none of the STASH items in that section would be available to the run. The UMUI panels define which versions of each section are available. The version mask digits are numbered from right to left; a 1 in position N implies that this item is available to version N of that section. A 0 implies that it is not.

**DataT** Data type code: 1 Real, 2 Integer, 3 Logical.

**DumpP** PP LBPACK code for data in the dump. See UM document F3.

**PCn** Packing accuracy codes.

This code is used for packing to the PP files. 10 codes are available to describe 10 packing profiles for different output requirements.

Pn=i Fields are packed to an accuracy  $2^{**i}$ .

Pn=-99 Fields are not packed.

**Rotate** Rotation code.

This is for wind diagnostics. It indicates whether STASH receives the wind components relative to the model grid or the regular lat-long grid (for ELF models).

0 Data is either non-vectorial or is relative to the model's grid.

1 Data is passed to STASH relative to the lat-long grid.

**PPFC** PP field code - defined in PP package document  
(can be browsed on the web under UM documentation).

**USER** Not used.

**LBVC** Defined in PP package document.

**BLEV, TLEV** Base/top level reference code (not used).

**RBLEV** Defined in PP package document.

**CFLL, CFFF** CF level/field codes - defined in CF documentation.

### Rules for the Level Compression Flag

The value of the level compression flag determines whether the input to STASH from a field is passed in on all available levels and pseudo levels (flag=0) or on a list of specified levels (flag=1). The only restrictions on the flag value are:

- (1) Any item with a non-zero space code must have a levels flag of zero.
- (2) Any item with levels type 3,4,7,8, or 9 (non-model levels) must have a levels flag of 1.
- (3) If the levels type is 5 (single level) then the flag is zero if the pseudo-type is zero, and 1 if the pseudo-type is non-zero.

If the levels type is 1,2 or 10 (model levels) then the flag may be 0 or 1 (0 for an intercepted diagnostic - see section 1 above).

## **APPENDIX 4 User-Defined STASHmaster Records**

Users may define their own STASHmaster records which can either supplement the records in the system STASHmasters or overwrite existing STASHmaster records. By this means it is possible to obtain new diagnostics, include new prognostics and ancillary fields, and correct existing STASHmaster records.

The user must also provide appropriate mods to the source code to calculate any new field he defines in this way and interface it to the STASH routines.

There can be any number of user STASH files. Each file should contain STASH records for one internal model only. The records can be in any order but pairs of identical records are not allowed. Extreme care should be taken when constructing the records as small format errors can cause serious problems which may not become apparent until the STASH routines are executed.

Users should keep their own copies of user STASHmasters and point to these. Changes made to user STASHmasters after a job has been processed will not be effective. User STASHmasters should not be edited during a UMUI session. If user STASHmasters are to be shared it may be advisable to put them in a special directory and keep all users informed of any changes.

If any user STASHmaster contains prognostics the user must specify initialisation choices in the appropriate UMUI panel.

The format for a user STASHmaster is shown below. The header section H1, H2 and the format record are optional. Any line starting with a # is ignored by the UMUI and the UM.

Note the following:

- (1) The end of file record is essential.
- (2) Integers do not have leading zeros. The option code and version mask are strings of single integers and do need leading zeros.
- (3) The exact spacing of the entries is important.

When a job is processed, the UMUI concatenates all user STASHmasters corresponding to a particular internal model into a single file. These files appear in the job library as PRESM\_A, PRESM\_O, PRESM\_S, PRESM\_W, i.e., one for each internal model. These files will be present in the job library even if the user has not defined any user STASHmaster records.



## APPENDIX 5. D1 Addressing Array

The D1\_ADDR array holds a list of information about all objects in the D1 array. A single or a multilevel field counts as one object. Note that when running in MPP mode, each processing element (PE) has its own copy of D1 and, therefore, its own D1\_ADDR array. The number of objects in D1 for each submodel is stored in a 1D array NO\_OBJ\_D1.

Information in the D1\_ADDR is used primarily for gathering and scattering fields between PE0 and all other PEs in MPP jobs. These operations occur mainly in the climate meaning, dumping, and PP output routines.

D1\_ADDR includes pointers to the equivalent objects in the STLIST and LOOKUP arrays, and similarly STLIST information includes a pointer to D1\_ADDR (see the D1\_ADDR d1\_stlist\_no and d1\_lookup\_ptr elements, and the STLIST st\_position\_in\_d1 element). Note that D1\_ADDR is dimensioned by submodel whereas STLIST is not. Therefore, when looping through the STASH list, check that the internal model identities compare. The following code example shows how the cross-referencing between arrays can be used. The example loops through the STLIST items and selects only those in the atmosphere submodel (which can include items in the SLAB internal model):

```
*CALL TYPD1 ! Contains comdeck D1_ADDR

! Get pointer to the partition of D1_ADDR relating to atmosphere
  MODNUM=SUBMODEL_FOR_SM(atmos_im)
  DO IE=1,TOTITEMS ! Loop over STLIST entries
! Get pointer to element in D1_ADDR array
  PTD1=STLIST(st_d1pos,IE)
! Compare internal model ids - ocean items would not match.
  IF(STLIST(s_modl,IE).eq.D1_ADDR(d1_imodl,PTD1,MODNUM))THEN
! This STASH item is part of the atmosphere submodel
```

### Accessing information in D1\_ADDR

The D1\_ADDR array is held within the same super array as D1 and is carried, with D1, in ARGD1 and ARTD1.

The D1\_ADDR array has three dimensions:

```
INTEGER D1_ADDR(D1_LIST_LEN,NO_OBJ_D1_MAX,N_SUBMODEL_PARTITION)
```

D1\_LIST\_LEN : Number of items of information held about each object.

NO\_OBJ\_D1\_MAX : Number of objects in biggest submodel

N\_SUBMODEL\_PARTITION : Number of submodel partitions

The partition number for the submodel im\_ident can be obtained from SUBMODEL\_FOR\_SM(sm\_ident) held in the CSUBMODL comdeck. Note that the partition number is not identical to the submodel identity number. For example, in an ocean-only job, the submodel identity is 2 but because there is only one submodel in use, the partition number is 1.

Some of the information from the D1\_ADDR array is output as a table "Addressing of D1

array".

The information held about each object is as follows:

1. d1\_object\_type
2. d1\_imodl
3. d1\_section
4. d1\_item
5. d1\_address
6. d1\_length
7. d1\_grid\_type
8. d1\_no\_levels
9. d1\_stlist\_no
10. d1\_lookup\_ptr
11. d1\_north\_code
12. d1\_south\_code
13. d1\_east\_code
14. d1\_west\_code
15. d1\_gridpoint\_code
16. d1\_proc\_no\_code

This list is held in the D1\_ADDR comdeck.

1. d1\_object\_type: can be set to prognostic=0, diagnostic=1, secondary=2 for secondary diagnostics and other=3. "other" means the space code 9 items such as P\_EXNER or the dual time ocean diagnostics.
2. d1\_imodl: internal model identity number. eg 3 for SLAB
3. d1\_section: section number.
4. d1\_item: item number.
5. d1\_address: address in D1. When running on MPP machines this refers to the addressing of the local copy of D1.
6. d1\_length: length of the local record.
7. d1\_grid\_type: grid type; from the STASHmaster.
8. d1\_no\_levels: number of levels including pseudolevels.
9. d1\_stlist\_no: a pointer to the STASH list array. Set to -1 for prognostics.
10. d1\_lookup\_ptr: a pointer to the dump header lookup table.
- 11-14. Addresses for the northern, southern, eastern and western rows; identical to the STASHlist records.
15. d1\_gridpoint\_code: From the STASHmaster.
16. d1\_proc\_no\_code: From the STASHmaster.

Unused elements of the array are initialised to -1. These unused elements would include STASH related information for prognostic items (prognostic items have no STASH entries). Additionally, in coupled runs, the two submodels are likely to have differing numbers of D1 objects, so the shorter submodel will include some empty records.