The Met.Office

UNIFIED MODEL DOCUMENTATION PAPER
C6: OASIS COUPLING
Technical overview

JC Thil (jcthil@meto.gov.uk)

VERSION 1

February 05, 1998

MODEL VERSION 4.4

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

Modification Record		
Document version	Author	Description....................
Draft 1.0	J-C Thil	First version. 05/02/98

Contents

1 Introduction

1.1 Background

For historical reasons, atmosphere, ocean, ice models have been developed as separate entities and often by different groups. Whenever the need to merge these models occurred, mainly for climate prediction, or medium term forecast, those groups were left with no alternative but to effectively tackle the merge of the models at the level of their implementation ; a lengthy and costly operation especially with regard of the complexity of such codes. The OASIS (Ocean Atmosphere Sea Ice Soil) coupler developed at CERFACS by Terray et al. aims at easing this task : instead of merging the codes of the models; the models are kept mostly unchanged and interact with another program called the coupler. This note describes the implementation of the changes made to the UM so that it can interact with the OASIS coupler; It does not describe the coupler itself and one shall refer to the original documentation of OASIS for its description.

2 Code structure

2.1 General layout

Apart from some minor changes, the code of OASIS provided by CERFACS remains unchanged from its original version, and those changes which were introduced into the original software were passed on to CERFACS in order to be included into future releases of OASIS. On the other hand, the UM code received a few additions in order to be able to communicate with OASIS; those modifications are discussed here.

2.2 Location of the OASIS libraries

Similarly to GCOM, the OASIS libraries are kept outside the UM code maintenance system. The reasons for this are as follows :

OASIS is an external product and there would be implications as far as the copyright is concerned whenever the UM is distributed to tier organizations.

Maintainability : OASIS is still being developed at CERFACS, and new versions may be imported en bloc in the future.

Integrating the entire file system of OASIS into the UM libraries would require a sensible amount of work and prevent future enhancements of the system to be implemented easily.

2.3 Fortran code : algorithm

For the sake of simplicity, the changes required for the use of OASIS were kept to a minimum of places. The plug routine to OASIS occurs into the the main loop of the model at the top level of the UM so that in pseudo-code the main loop of the UM with OASIS looks like :

```
UM_SHELL(...){
   /*
    *  Pseudo-tree of the UM below the general level
    *  of UM_SHELL which is not relevant to this material.
    *  This holds for the Atmosphere internal model,
    *  for the Ocean, formally replace the call to
    *  ATM_STEP with OCE_STEP
    */
   loop on all time-steps {
         /*
          * Increment time according to the size of
```

```
          * the time step of the model currently running
          */
         call INSCRTIME(...)

         /*
          * connect to the external OASIS coupler if some
          * fields need to be exchanged. (see below)
          *
          */
         call OASIS_STEP(...)

         /*
          * call one time-step of the model
          */
         call ATM_STEP(...)

   } end of the loop on time-steps.
   /*
    * Miscellaneous closure instructions.
    */
} UM_SHELL
```

The first occurrence of the OASIS routines behaves slightly differently from subsequent ones since it also handles the creation of various description files for grids and masks of the coupling fields. It also creates UNIX named pipes for each of the coupling fields. After the first call to OASIS_STEP has occurred, the rest of the calls simply handle the gathering/scattering of the fields across the array of processors as well as some minor computations like units conversions, and synchronize their import/export to OASIS.

```
OASIS_STEP{
   /*
    * Pseudo-tree of the OASIS routines in the UM
    * ie : call tree of the OASIS_Step routine.
    */

   /*
    * Whenever OASIS_STEP is called for the first time :
    * -Initialize the named pipes and create description files
    *  of the coupling fields to communicate with the OASIS
    *  external coupler.
    * -Synchronize the first timestep with OASIS.
    * After init_oasis has been called for the first time,
    * it won't be called anymore.
    */
   call init_oasis(...)

   Loop on the list of coupling fields {

      If the current field is to be imported at this timestep
      then {
         wait for a signal on the named pipe
         read the field on PE 0
         scatter the field across all the PEs
      } else if the field is to be exported at this timestep
      then {
         gather the field from all PEs to PE 0
         send a signal on the named pipe dedicated to the field
         write the field on a file, ready to read by the coupler
      }
```

```
    } End Loop

} OASIS_STEP
```

2.4 Scripts

The OASIS functionality is nested into the web of scripts which constitutes the UM. As far as OASIS is concerned, the scripts can mainly be used into two modes, the MASTER one and the SLAVE one. In the former, the instance of the UM handles all the system aspects of OASIS, like compilation, launch of the submodels, submission to the queueing system, while in the latter, the model behaves basically like a standard UM job would. This arrangement allows the flexible use of either the UM with OASIS and UM submodels or external submodel models with a UM component.

The modified pseudo-tree of the qsmaster script to cater for the use of OASIS follows :

```
qsmaster
  |
  |--> qsoasis
  |
  |--> if [ OASIS = 'true' ]
  |     then
  |        # start all : coupler and submodels.
  |        LAUNCH OASISEXEC external_model qsmain
  |     else
  |        qsmain    # usual non OASIS way.
  |     fi
  |
  |--> qsfinal
  |         |
  |         |-- if [[ OASIS = 'true' ] && [ OMASTER = 'true' ]]
  |         |    then
  |         |       qsresubmit
  |         |    else
            |       nothing # a non-master job does not resubmit itself.
            |    fi
```

qsoasis is the script dedicated to handling OASIS itself.

2.4.1 Compilation

Makefile files handle the compilation of the OASIS libraries; they are called from within the oasis script system whenever the user requests a new compilation of OASIS, however under normal circumstances (standard number of fields with usual resolution) the standard built libraries should satisfy a normal user. If it were not the case the output file of the OASIS coupler ('cplout') will ask for more memory.

2.5 Time-stepping

The time-stepping strategy is entirely flexible when using OASIS; the models can run parallely or in sequence (although for reasons of time-slicing, the latter not advised on the t3e) and fields may themselves be exchanged at different frequencies within the same model. This kind of feature may for example be useful when coupling physical phenomena of various timescales. Information describing the time-stepping strategy must be specified through the 'namoasis_atm' (respectively 'namoasis_oce') input file for the atmosphere (respectively ocean) component. See in appendix A for a description of those files.

Of course, the system provided being entirely flexible, it requires the user to ensure consistency between each of the components.

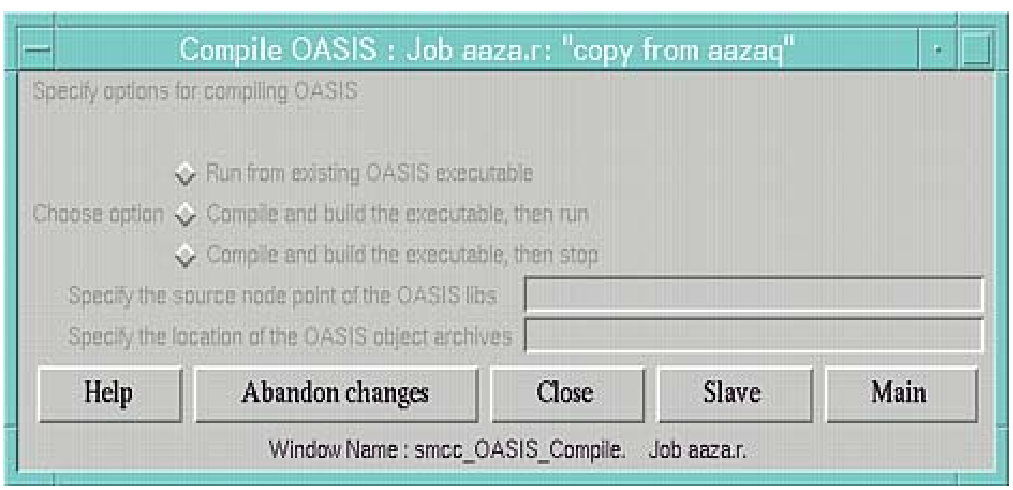
2.6 Umui panels

Three panels are provided to pass on information to the coupler; they can be accessed from 'Sub-Model Configuration and Coupling' followed by 'OASIS coupling'. They are :

- the *main panel* selects whether the current job will be the master job or the slave job. It is the only panel that should be accessed when setting up the slave job. Two further panels have to be accessed for the master job as it handles the build of OASIS and effectively starts the executables of the models and the OASIS executable. In effect, a slave job is very close to a standard UM job.

- the *slave panel* is only accessible from a master job *(!)* and sets up the script entry point of the slave model ; for example, in the case of the UM, this is the *umuisubmit* script file generated in the job library on the t3e when a job is submitted. Note that because of the particular nature of the submission system of the UM, you will need to first submit the slave job in order to effectively know the location of the script. Other external models might not have this 'feature'.

- the *compile panel* is only accessible from the master job and is used to set various modes of compilation of the OASIS coupler itself.



- An extra panel allows the user to ask the C99_1A section dedicated to the UM plug-in routines of OASIS to be included into the UM. It may be accessed from 'Submodel Configurations and Coupling' followed by 'Coupling pre-compiled coupling sections'.

2.7 MPP specific code

The sequential and MPP versions of the code only differ in the gathering/scattering events added in the MPP version to handle each of the fields; no explicit use is made of the parallel decomposition capabilities of the OASIS coupler : this allows to make use of the most thoroughly tested version of OASIS : the unix named-pipes version. In a nutshell fields are gathered/scattered across the PEs within each of the models towards a dedicated PE which serves as front-end for the communication with the OASIS coupler. OASIS itself runs on a single PE while the models each run on a distinct batch of PEs.

2.8 Memory requirements

More memory is required to handle the coupling fields when OASIS is used by comparison to a stand alone submodel, but the increase is limited. On the other hand, the coupler requires a significant amount of memory since for each coupling field the coordinate of each point, the surface of the gridbox, the mask, etc... are permanently stored, to the point that the OASIS executable requires the usage of a large memory PE of the T3E machine currently installed in the premises of the Met Office. However, if one compares a OASIS coupled experiment with both UM ocean and atmosphere submodels against a standard UM coupled experiment of otherwise equivalent configuration, the OASIS one should be substantially smaller because the standard coupling on MPP platforms keeps both copies of the ocean and the atmosphere on top of the current running submodel. On a non-MPP platform, this is not the case and OASIS should increase the memory requirements against a similar model. When setting up a job coupled with OASIS, one should require a memory allocation equivalent to the sum of the memory requirement for each of the submodels plus some extra-memory for OASIS itself.

3 Usage

3.1 UM-submodel vs UM-submodel

The user should set up two UM jobs, one being the MASTER job will handle OASIS and submit itself to the supercomputer, while the other one should be the SLAVE model. For reasons of file-naming conventions, the SLAVE job should be processed first.

Both the models and OASIS, should run into the same directory, therefore the user is advised not to use the usual default of running the UM into the job directory, but rather into the experiment directory or alternatively in another unrelated directory.

3.2 UM-submodel vs external-model

Only one UM job should be set up, and It is advised to make the UM the MASTER job, so that it takes on the burden of building OASIS (if required), launching the submodels, and restarting itself.

The files required by OASIS which describe the layout of the grids *grids*, their masks *masks*, and the surfaces of the gridbox horizontal domains *areas* should be produced by the external model and written into those files during the initialization process of the external model. Care should also be taken to generate the appropriate UNIX named pipes for each of the fields exchanged with OASIS.

4 Future developement

4.1 Communication Libraries

For the time being, only one communication paradigm has been implemented : the UNIX named pipes. Another communication system already existed at the time the project was carried out : the PVM ie: Parallel Virtual Machine one, but it has not much been used with OASIS up to now, and possible conflicts could have arisen with the rest of the

MPP version of the UM. The coming vn2.2 release of OASIS should make use of the shared memory IPC communication protocol to exchange and synchronize the fields. This feature is interesting in some respect; since the exchange of data occurs directly in memory, it is more efficient (ie : faster) than the pipes which route the data through the file-system. Unfortunately, according to early tests, this method cannot run onto the T3E, since some of the operating system aspects it refers to are not implemented into UNICOS/mk, however, it currently runs on the VPP 700 machine of ECMWF. One can hope that this new feature will be able to run onto fully UNIX system V compliant machines.

When using the MPP version of the UM, the fields are gathered/scattered internally in the UM towards a single PE which then communicates with the OASIS process through UNIX named pipes. This constitutes an acceptable solution since, the mechanisms to gather/scatter the fields are well in place inside the UM, and whatever the solution envisaged, the fields have to find themselves on a single PE at some point to be handled by the single process OASIS.

5 Conclusion

OASIS provides a flexible solution to import and test externally developed sub-models against the Unified Model; this document describes the implementation of OASIS as conceived in the Unified Model to the exclusion of the possible other routes which could have been developed.

A Files

The following files are input to either OASIS or the UM and should be provided by the user.

- *namcouple* is the standard input file of OASIS; it entirely describes the simulation from the point of view of OASIS. See the original documentation of OASIS for a full description of the file. One should note that the first part of the file is generated by the master UM job of the umui, the second part, beginning with the keyword $STRINGS has to be generated with the help of a text editor and subsequently passed on to the umui in the 'interpolation description file' field of the main OASIS umui panel.
- *namoasis_atm* : input file for the UM-atmosphere provided through the umui into the field 'description file for this model' of the main OASIS panel; the first line specifies the number of coupling fields from the viewpoint of the atmosphere, the remaining line describe each of the coupling fields. The items describing the coupling fields are :
  - Stash code of the field
  - The type of the grid : T or U
  - The file is exported (E) or imported (I) from/into the UM
  - The frequency in timesteps of the UM of the coupling for the field
  - The first coupling event in timestep for the field ; the rest of the coupling events will occur at the specified frequency, offset by the first coupling event value.

  The sample file below corresponds to the standard list of coupling fields as used in ocean-atmosphere UM integrations for the atmosphere sub-model.

```
17                   ! number of coupling fields.
03219 U E 48 49      ! tau x
03220 U E 48 49      ! tau y
03224 T E 48 49      ! windmixing power
01204 T E 48 49      ! penetrative SW radiation
03228 T E 48 49      ! non penetrative surface heatfluxes
04203 T E 48 49      ! precipitation minus evaporation
08205 T E 48 49      ! runoff
04204 T E 48 49      ! snowfall
03231 T E 48 49      ! sublimation
03201 T E 48 49      ! sea-ice diffusive heatfluxes
03235 T E 48 49      ! sea-ice topmelt heatfluxes
00031 T I 48 49      ! sea-ice fraction
00024 T I 48 49      ! sst
00032 T I 48 49      ! ice-depth
00023 T I 48 49      ! snowdepth
00028 U I 48 49      ! u surface current
00029 U I 48 49      ! v surface current
```

- *namoasis_oce* : same as for the atmosphere, but for the UM-ocean submodel. It's location must also be input through the umui of its UM job; The sample file below is the counterpart of the above one for the ocean part of an atmosphere-ocean coupling experiment. Note that imported fields in one model become exported and the other, also, coupling frequencies of fields vary accordingly of the different time-step duration of the models in order to insure synchronization of the models.

```
17                  ! number of coupling fields.
00141 T E 24 25     ! snowdepth
00146 T E 24 25     ! sea ice fraction
00101 T E 24 25     ! sst
00147 T E 24 25     ! ice depth
00121 U E 24 25     ! u surface current
00122 U E 24 25     ! v surface current
00150 U I 24 25     ! tau x
```

```
00151 U I 24 25   ! tau y
00152 T I 24 25   ! windmixing power
00161 T I 24 25   ! penetrative SW radiation
00162 T I 24 25   ! non penetrative heat fluxes
00165 T I 24 25   ! precipitation minus evaporation
00166 T I 24 25   ! runoff
00171 T I 24 25   ! snowfall
00172 T I 24 25   ! sublimation
00191 T I 24 25   ! sea-ice diffusive heatflux
00190 T I 24 25   ! sea-ice topmelt heatflux
```

The files below are generated at runtime into the $OASISRUNDIR directory by either OASIS or the models but it can be useful to know about them :

- *cplout* : verbose output file of the coupler ; it delivers information on the progress of the simulation from the point of view of OASIS. It is useful to browse this file; especially when things go wrong.
- *masks* : contains the land-sea masks for each field and for each grid used. This file is generated by the UM and subsequently input into OASIS. When using a model other than the UM, care must be taken to generate this file. The format of the file is described in the OASIS documentation.
- *grids* : contains the latitudes and longitudes for each of the points of each grid for each of the fields. This file is generated by the UM and subsequently input into OASIS. When using a model other than the UM, care must be taken to generate this file.
- *areas* : contains the surface area for each gridbox of each of the coupling fields. This file is generated by the UM and subsequently input into OASIS. When using a model other than the UM, care must be taken to generate this file.
- *mweights*, *gweights* : contain weights associated with each of the grid-points ; those files are generated and subsequently used by OASIS at runtime.
- UNIX named pipes for initial synchronization of the models : they are Preadm0x, Pwritem0x, where 'x' is the number of the submodel from the viewpoint of OASIS. Preadm0x serves for the UM to send initial information to the coupler, Pwritem0x is used by OASIS to send initial data to the UM-submodel.
- UNIX named pipes for initial synchronization of each of the exchanged fields. Their names are exactly the stash codes of the exchanged fields; ie : a 5 number digit.
- files for exchanging each of the fields ; their names are : 'UMxxxxx' where xxxxx is the stash code of the field exchanged between the UM-submodel and OASIS.
- *.assign_oasis* contains the assign commands issued at run-time.

B Environment variables

This section describes the korn shell environment variables which are provided through the umui. It must be stressed that this is an important aspect of the UM which relies heavily on UNIX's shell environment system.

- $OASISRUNDIR : directory in which the OASIS executable runs ; set by default at $DATAW. Note that with the pipe version models should also reside into this same directory (ie the running directory of the models must be OASISRUNDIR as well).
- $OMASTER : 'true' or 'false' ; tells whether the job is a master job (is it handles the set up of the OASIS executable and launch of submodels) or not.
- $STEPC : Steps for OASIS compilation ; 4 means no compilation, 1, means full re-compilation of the coupler.
- $SRCCPL : source node point of the OASIS libraries.

- $LIBCPL : OASIS object archives files.
- $FIELDSTRINGS : last part of the 'namcouple' OASIS input file describing the coupling fields and the interpolation methods.
- $NFIELDS : total number of coupling fields (exported and imported).
- $NMODELS : number of models coupled by OASIS ; currently limited to 2.
- $MESSAGEPASSING : message passing paradigm to exchange the fields, currently, only the 'PIPE' version is implemented.
- $TIMESTRATEGY : tell whether the integration of the models is carried out in parallel (1), or sequentially (2).
- $MASTERNAME : name of the executable of the 'master' model
- $SUBMASTERNAME : name of executable of the model which is not the master one.
- $SUBLOADMODULE : entry point of the scripts of the model which is not the master.
- $LIBPVM : location of the PVM object archive libraries.
- $namoasis : input description file of the coupling fields for the model
- $localnamoasis : currently hard-coded to 'namoasis_atm' for the UM atmosphere, and 'namoasis_oce' the UM ocean, this is the name of the input description file of the coupling fields for the model as they stand into the $OASISUNDIR directory.
- $FILENV : this variable points to the '.assign_oasis' file (see above) and is used by UNICOS/mk prior accessing any file.

C Executables

- Both sub-models.
- OASISEXEC : the OASIS executable.
- LAUNCHER : a program whose role is to start the above programs in related manner in the sense of UNIX. Ie : whenever one of the processes terminates abnormally, the others do as well, this in order to prevent some processes to hang, waiting for others while the process they are waiting for is actually dead.

D Coupling Fields

In the UM, it is usually best to refer to physical quantities with their stash codes, as they probably offer the only unequivocal way to describe diagnostic or prognostic variables. The following two tables then give the correspondence between the UM-ocean and UM-atmosphere coupling fields as they are used for standard ocean-atmosphere coupling. It also gives the names of the internal pointer to the stash array as used into the code. Whenever the physical quantity is derived from a combination of others, the formula is given and the stash code we should refer to for the whole is emphasized in **bold**.

A plain text description of the physical quantities involved is appended to the tables.

Coupling fields Atmosphere towards ocean :

Field	*Atmosphere model*	*Ocean model*
taux	03219 ja_taux	00150 jo_taux
tauy	03220 ja_tauy	00151 jo_tauy
windmix	03224 ja_windmix	00152 jo_windmix
non penetrative surface heatfluxes	$ja\_solar(01203) - ja\_blue(\mathbf{01204}) + ja\_longwave(02203) - ja\_sensible(03228) + LC * ja\_evap(03232)$	00162 jo_heatflux
precipitation minus evaporation	If SEAICE undefined $ja\_lssnow(04204) + ja\_cvconv(05206) + ja\_lsrain(\mathbf{04203}) + ja\_cvrain(05206) - ja\_evap(03232)$	00165 jo_pminuse
	If SEAICE defined $(ja\_lssnow(04204) + ja\_cvsnow(05206)) * (1.0 - ja\_aice(00031)) + ja\_lsrain(\mathbf{04203}) + ja\_cvrain(05206) - ja\_evap(03232))$	
runoff	$ja\_slowrunoff(\mathbf{08205}) + ja\_fastrunoff(08204)$ (also converted from daily to instantaneous)	00166 jo_riverout
snowfall	$ja\_lssnow(\mathbf{04204}) + ja\_cvsnow(05206)$	00171 jo_snowfall
sublimation	03231 ja_sublimation (also converted from daily to instantaneous)	00172 jo_sublim
sea-ice diffusive heat flux (botmelt)	03201 ja_botmelt	00191 jo_botmelt
sea-ice topmelt heatflux	03235 ja_topmelt	00190 jo_topmelt

Ocean towards atmosphere :

Field	*Ocean model*	*Atmosphere model*
sea ice fraction	00146 jo_icecon	00031 ja_aice
ice depth	00147 jo_icepth (remove ice depth at boxes with less than the minimum ice fraction, set ice depth to zero at ice free boxes)	00032 ja_icedepth
snowdepth	00141 jo_snowdepth (multiply by the density of snow RHOWSNOW in kg/m^3)	00023 ja_snowdepth
sst	00101 joc_tracer(1,2) (also unpacked and converted from C till K)	00024 ja_tstar
u surface current	00121 joc_u(2) (also unpacked and rescaled from cm/s to m/s)	00028 ja_ucurr
v surface current	00122 joc_v(2) (also unpacked and rescaled from cm/s to m/s)	00029 ja_vcurr

Excerpt of the STASH table for the quoted codes:

```
03219   61 X-COMP OF SURF & BL WIND STRESS N/M2
00150  721 TAUX: X_WINDSTRESS           N/M2  A
03220   62 Y-COMP OF SURF & BL WIND STRESS N/M2
00151  722 TAUY: Y_WINDSTRESS           N/M2  A
```

```
03224  182 WIND MIXING EN'GY FLUX INTO SEA W/M2
00152  627 WME: WIND MIXING ENERGY FLUX W/M2  A
01203  186 NET DOWN SW RAD FLUX: OPEN SEA
01204  186 NET DOWN SURFACE SW FLUX BELOW 690NM
02203  187 NET DOWN LW RAD FLUX: OPEN SEA
03228  178 SURFACE SH FLUX FROM SEA (GBM)  W/M2
03232  285 EVAPORATION FROM SEA (GBM)   KG/M2/S
00162  626 HTN:NONPEN.HT.FLX*LF INTO OCN W/M2 A
04204  118 LARGE SCALE SNOWFALL RATE    KG/M2/S
05206  119 CONVECTIVE SNOWFALL RATE     KG/M2/S
00031   37 SEA ICE FRACTION AFTER TIMESTEP
04203   99 LARGE SCALE RAINFALL RATE    KG/M2/S
05206  119 CONVECTIVE SNOWFALL RATE     KG/M2/S
03232  285 EVAPORATION FROM SEA (GBM)   KG/M2/S
08205  112 SUB-SURFACE RUNOFF AMOUNT   KG/M2/TS
08204  111 SURFACE RUNOFF AMOUNT       KG/M2/TS
00166  631 RIVER OUTFLOW INTO OCEAN  KG/M2/S  A
04204  118 LARGE SCALE SNOWFALL RATE    KG/M2/S
05206  119 CONVECTIVE SNOWFALL RATE     KG/M2/S
00171  623 SNOWFALL INTO OCN/ONTO ICE KG/M2/S A
03231  107 SUBLIM. FROM SURFACE (GBM)  KG/M2/TS
00172  624 SUBLIMATION FROM SEAICE    KG/M2/S A
03201  261 HEAT FLUX THROUGH SEA ICE (GBM) W/M2
00191  682 GBM SEAICE BOTMELT HEAT FLUX W/M2  A
03235  260 SEAICE TOP MELTING LH FLUX(GBM) W/M2
00190  681 GBM SEAICE TOPMELT HEAT FLUX W/M2  A

00146  683 AICE : ICE CONCENTRATION
00031   37 SEA ICE FRACTION AFTER TIMESTEP
00147  687 HICE: MEAN ICE DEPTH OVER GRIDBOX  M
00032  687 SEA ICE DEPTH (MEAN OVER ICE)      M
00141  688 SNOW DEPTH (OCEAN)                 M
00023   93 SNOW AMOUNT AFTER TIMESTEP     KG/M2
00024   16 SURFACE TEMPERATURE AFTER TIMESTEP
00028  701 SURFACE ZONAL CURRENT AFTER TIMESTEP
00029  702 SURFACE MERID CURRENT AFTER TIMESTEP
00101  601 POTENTIAL TEMPERATURE (OCEAN)  DEG.C
00121  701 BAROCLINIC U_VELOCITY (OCEAN)   CM/S
00122  702 BAROCLINIC V_VELOCITY (OCEAN)   CM/S
```

Note the inconsistency between the above list and the actual definition of the fields 08205, 08204, 03231 ; they are actually daily accumulation and not timestep accumulation as the list would suggest.