

What's New with the I/O API

Carlie J. Coats, Jr., Ph.D.

UNC Institute for the Environment

The Models-3/EDSS Input/Output Applications Programming Interface (I/O API) provides the environmental model developer with an easy-to-learn, easy-to-use programming library for data storage and access, offering *selective direct access* to the data in terms meaningful to the modeler. For example, the following is a direct English translation of a typical `READ3()` call:

```
Read layer 1 of variable 'OZONE' from 'CONCFIL' for 5:00 PM GMT on
July 19, 2018 and put the result into array A.
```

Note also that this allows us to read or write anywhere in the file (variables, dates×, layers) without performance penalty, even if the file is decades long.

There are a number of new capabilities and options available in the Models-3 I/O API, which are of interest and use to the CMAS community of modelers and others:

- **Now Available on GitHub:** go to <https://github.com/cjcoats/ioapi-3.2> for download, <https://cjcoats.github.io/ioapi/AA.html> for documentation. This allows easy update for individual bug-fixes, without having to download and build the entire package. (Access from the CMAS web site is still available, unchanged)
- **Increased maximum numbers of files and variables.** There are two versions:
 - For the **default** release after October 1, parameter `MXVARS3=256`, to support various **CMAQ** and **SMOKE** applications and with chaining filke sequences (see below). *This change should be transparent to the vast majority of applications (others may possibly need re-compiling; CMAQ's `pario` should be the only problem).*
 - **New version I/O API-3.2-Large** for use in **CMAQ DDM** and **ISAM** applications has `MXFILE3=512` and `MXVARS3=16384`; it is not link compatible with the regular version. **I/O API-3.2-Large** needs to be **built separately from the normal version, and kept carefully isolated from it**; it will have slightly lower performance and substantially greater memory requirements, due to these size-increases. It can be downloaded from <https://www.cmascenter.org/ioapi/download/ioapi-3.2-large.tar.gz> at the CMAS web site.
- **PnetCDF Distributed I/O** for **CMAQ**. Note that this is pretty-much limited to **CMAQ** only...
- **Chaining File Sequence** capabilities using `LIST:`, e.g., to handle all the single-day outputs from a study as though they were a single file. Data is read from the first file in the list which contains the requested date&time. For example, the following would do a unified `m3stat` run for all the files of a given type for a 31-day month:

```
setenv NAME_1 <path>
. . .
setenv NAME_31 <path>
setenv F00 "LIST:NAME_31,...,NAME_1"
m3stat F00 DEFAULT
```
- **MODULE M3UTILIO** replaces **INCLUDE**-files (use of which has been deprecated since 2002); has subroutine **INTERFACES** for argument-checking, and generic routines, e.g., **ENVGET()** subsumes the following older specific forms (which are still available):

```
BENVDBLE(LNAME, ...): get DOUBLE PRECISION from environment, with bounds-checking
BENVINT(LNAME, ...): get INTEGER from environment, with bounds-checking
BENVREAL(LNAME, ...) get REAL from environment, with bounds-checking
ENVDBLE(LNAME, ...): get DOUBLE PRECISION from environment
ENVINT(LNAME, ...): get INTEGER from environment
ENVREAL(LNAME, ...): get REAL from environment
ENVVYN(LNAME, ...): get LOGICAL from environment from the environment
```
- **New MODULE MODATTS3** for maintaining extra file-header metadata: https://www.cmascenter.org/ioapi/documentation/all_versions/html/MODATTS3.html
- **CF metadata** generated automatically. This allows import I/O API files into GIS systems such as **ARC-INFO**, many other tools. Turned on by

```
% setenv IOAPI_CFMETA YES
```
- **Grid-transform matrix** metadata: input, output grid descriptions
- **CMAQ** metadata (*note*: automatically supported by **m3tools** programs)

- **New MODULE MODGCTP** for coordinate transform operations. https://www.cmascenter.org/ioapi/documentation/all_versions/html/MODGCTP.html
- **PARAMETERS** for GCTP spheroid names and indices
- Explicit **INTERFACES** for older routines `INITPROJ()`, `...`, `ALB2EQM()`
- New **generic transform routines**:
 - `XY2XY` does map-projection transforms on points and (1-D, 2-D) arrays of points, with optional spheroid argument
 - `GRID2XY` convert grid cell-centers to a different map projection,, with optional spheroid argument
 - `GRID2INDEX`, `PNTS2INDEX`, `INDEXMULT` for computing bilinear interpolation indices and coefficients, and optimized OpenMP-parallel bilinear-matrix multiplication
- **New MODULE MODNCFIO** for "Raw netCDF": https://www.cmascenter.org/ioapi/documentation/all_versions/html/MODNCFIO.html
- Declarations for netCDF and PnetCDF (necessary because PnetCDF **INCLUDE**-files are incomplete and partially-incompatible with netCDF **INCLUDE**-files)
- High Level interface to netCDF scientific data sets:
 - `CREATENC()`: Create a new "raw netCDF" file according to a supplied dataset definition
 - `DESCNCVAR()`: Return the list of variables for a "raw netCDF" file, together with their units, types, and dimension-info.
 - Generic `READNCVAR()`: Read a (1-D, 2-D, 3-D; `REAL(4,8)`, `INTEGER(1,2,4,8)`) variable, or a timestep of a variable, from a "raw netCDF" file.
 - Generic `WRITENCVAR()`: Write a variable, or a timestep of a variable, to a "raw netCDF" file.
- **MPAS support:** MPAS is a potentially-global unstructured-grid weather model that currently is being adapted for atmospheric chemistry, land-surface modeling, and other tasks. *[MPAS uses unstructured (non-rectangular) grids; the MPAS file format is quite complicated, with (among other things) 57 variables used to describe how an MPAS unstructured grid fit together. Just declaring the variables in a file-header takes more than 300 lines of code. Doing MPAS-format I/O correctly is most exceedingly tedious.]* See <https://mpas-dev.github.io/files/documents/MPAS-MeshSpec.pdf> for the MPAS grid and netCDF-file specifications.
- **MODULE MPASFIO** provides MPAS-format-netCDF I/O, with-grid descriptions; grid related utility routines (e.g., `FINDCELL()` for finding the cell that contains `<lat,lon>`, which is a PITA to do "by hand", as is the unstructured-grid interpolation):see https://www.cmascenter.org/ioapi/documentation/all_versions/html/MODMPASFIO.html
- MPAS related **m3tools** programs are `mpas2m3`, `mpasdiff`, `mpasstat`
- **New MODULE MODWRFIO:** has high level routines and **INTERFACES** for reading and writing WRF netCDF files:

```
OPENWRF()
Generic READWRF()
Generic WRITEWRF()
CLOSEWRF().
```
- There are also related **m3tools**-programs `wrfgriddesc`, `wrftom3`

- **New M3TOOLS programs** and capabilities: See https://www.cmascenter.org/ioapi/documentation/all_versions/html/AA.html#tools
- **OpenMP parallelism** for many **m3tools** programs
- New **date&time manipulation programs for scripting**. No extraneous output; support **YESTERDAY**, **TODAY**, **TOMORROW** as arguments:

```
greg2jul: Convert a Gregorian-calendar date YYYYMMDD to Julian YYYYDDD and echo the result.
Jul2greg: Convert Julian YYYYDDD to Gregorian YYYYMMDD, and echo...
juldifff: Echo difference (days) between two Julian dates
julshift: Shift Julian YYYYDDD by specified number of days
timeshift: Shift a Julian date&time YYYYDDD.HHMMSS by a specified time step HHMMSS and echo...
```
- e.g.: `% JDATE = `julshift $JDATE -1``
- MPAS tools: `mpas2m3`, `mpasdiff`, `mpasstat`, `mpaswtest`
- WRF tools: `wrfgriddesc`, `wrftom3`
- More/Enhanced programs: `dayagg`, `findwndw`, `gridprobe`, `insertgrid`, `m3mask`, `m3merge`, `m3probe`, `vertimeproc`, `vertintegral`; enhanced `latlon` and `m3stat` (now with threshold statistics)
- **New Build Options:**
- 32-bit and 64-bit **MS-Windows** under CygWin
- 360-day and 365-day **climatology-year** modes
- 64-bit "**medium memory model**" builds for *Linux/x86_64* supporting >2GB arrays and stack (which the compiler-default builds do not). *Use it for CMAQ-DDM, CMAQ-ISAM*
- PnetCDF Distributed I/O for **CMAQ** (above)
- **Miscellaneous**
- **Multi-decade/century support** (coding to avoid integer-overflow while processing long time periods).
- New **F90-generic subroutines** and functions using **MODULE M3UTILIO**
- **Standard-year**, standard-month, standard-week, standard-day support
- **INTEGER*1**, **INTEGER*2**, **INTEGER*8** support (including search and sort routines)
- New **map projections**: Albers, Lambert, and Sinusoidal Equal Area map projections.
- Routines **FILCHK3**, **GRDCHK3**, **IOPARMS3** for numerically-robust file-consistency checking
- **64-bit record** support for huge grids and/or huge variable-sets.
- Modifications for **netCDF-4**.
- "**Snoop mode**"—re-try read-operations until they become available, for use in model coupling
- Substantial use of other Fortran-90 features, such as "free" (`.f90`) source code format, auto and allocatable arrays.
- Hacks to support **gfortran**, which fails to follow industry standards (not even compatibly from version to version).