

ESMF Python Interface JST call summary

Date: November 13, 2012

Participants:

Charles Doutriaux - UV-CDAT
David Hassell - cf-python
Karl Smith - pyferret
Dave Kindig - TechX
Alex Pletzer - TechX
Arlindo da Silva - NASA

ESMF Core team:

Ryan O'Kuinghtons
Gerhard Theurich
Silverio Vasquez
Bob Oehmke
Peggy Li
Fei liu
Joe Jacobs

Minutes:

Initialize/Finalize:

- singleton class approach seems best (over static methods)
- Why not do the initialize() on import?
- don't have to use the functionality unless the class object is created
- more object oriented, can add other functionality to the class

Logging:

- should have the ability to turn off completely (not create log file) on initialize()
- logging should be off by default, turned on with a flag to initialize()

Get:

- use properties over explicit get() calls, in general
- esmf object must be immutable, so we cannot do something like `abc.local_pet = -1`

Names:

- ESMF nomenclature seems best, better to stick with the brand name

Documentation:

- keep documentation close to the code (doc strings)

Mesh:

- no strong typing in the Mesh creation interface, recast everything to the safest option
- there could be a performance hit, but a warning will be issued if this happens

Enumerated constants:

- these are good how they are (integers better than strings)
- should be able to pass a scalar for an array that is all the same value

Charles and Karl had to leave at this point.

Numpy arrays vs. C buffer pointers

- best solution would be to return multidimensional Numpy arrays with C style indexing
- this is a departure from current ESMF convention, which uses the native Fortran indexing