Fortran	python	Notes
CALL oasis_start_grids_writing (flag)		Implicitly in the pyoasis.Grid class constructor for the first instantiated grid
CALL oasis_write_grid (cgrid, nx_global, ny_global, lon, lat, il_part_id)	grid = pyoasis.Grid(cgrid, nx_global, ny_global, lon, lat[, partition])	Notice that *grid* is a user chosen variable name for each object of the Grid class lon and lat are entered as 2D arrays - for unstructured grids the second dimension has to be 1 - of the same shape (nx_loc, ny_loc) where nx_loc, ny_loc are the number of points in each dimension on the calling process. The optional partition argument is a member of the pyoasis.Partition class.
CALL oasis_write_corner (cgrid, nx_global, ny_global, nc, clon, clat, il_part_id))	grid.set_corners(clo, cla)	clo, cla are 3d arrays: the last dimension is the max. number of corners per cell, the two others are nx_loc, ny_loc.
CALL oasis_write_mask (cgrid, nx_global, ny_global, mask, il_part_id, companion)	grid.set_mask(mask[, companion])	mask is a 2d integer array of size nx_loc, ny_loc. Oasis convention: 1 for masked points, 0 elsewhere. The optional argument companion is the name of a companion grid.
CALL oasis_write_frac (cgrid, nx_global, ny_global, frac, il_part_id, companion)	grid.set_frac(frac[, companion])	frac is a 2d array of size nx_loc, ny_loc. The optional argument companion is the name of a companion grid.
CALL oasis_write_area (cgrid, nx_global, ny_global, area, il_part_id)	grid.set_area(area)	area is a 2d array of size nx_loc, ny_loc.
	grid.write()	Terminates the declarations for the grid *grid*
CALL oasis_terminate_grids_writing ()		Implicitly activated by the grid.write() of the last grid instance to call it
CALL oasis_def_var (var_id, name, il_part_id, [1,1], kinout, var_type, ierror)	var = pyoasis.Var(name, partition, inout)	partition is a pyoasis.Partition object as in previous section. inout is an enumerate parameter, either in the long form `pyoasis.OasisParameters.OASIS_OUT/OASIS_IN` or in the short form (cf Module) `OASIS.OUT/OASIS.IN` Notice that the var_type is left to the dtype of the exchanged field. Once again *var* is a user chosen variable name for an object of the Var class.
CALL oasis_def_var (var_id, name, il_part_id, [1,n], kinout, var_type, ierror)	var = pyoasis.Var(name, partition, inout[, bundle_size=n])	The size of the bundle is set via the optional argument bundle_size
CALL oasis_enddef (ierror)	comp.enddef()	Notice that it is a method of the comp object, comp being the user chosen variable name of the component