esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Compatibility Between ESDM and NetCDF4

Julian Kunkel Luciana Pedro

Work Package:	Work Package 4 Exploitability
Responsible Institution:	University of Reading
Contributing Institutions:	Deutsches Klimarechenzentrum GmbH (DKRZ), Science and Technology Facilities Council (STFC), Centro Euro-Mediterranean sui Cambiamenti Climatici (CMCC), Seagate Systems UK Limited (SEAGATE)
Date of Submission:	30 June 2019

Contents

1 Files From Directory nc_test4

1.1 h5testszip.c

Example illustrates the use of SZIP compression in HDF5.

***PASS

Status: SUCCESS!

1.2 Test cdm_sea_soundings.c

The cdm tests confirm complience with the Common Data Model. This file creates some sample data structures to hold sea soundings.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

1.3 Test t_type.c

This test program is only built if netCDF-4 is disabled. It tests the netCDF-3 version of NC_inq_type().

*** Tests successful!

Status: SUCCESS!

1.4 Test test_szip.c

Example illustrates the use of SZIP compression in netCDF5. Taken from HDF5 example.

ESDM does not support compression!

Remove lines 67-88. Specific to compression.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.5 Test tst_atts.c

Test the netCDF-4 attribute code.

//TODO Testing the permission and reserved words before name/rename.

Remove lines 90-98. Expected error: NC_EPERM

Remove lines 116, 118, 234. Expected error: NC_ENOTVAR

Remove line 117. Expected error: NC_EINVAL

Remove lines 148, 180, 186. Expected error: NC_ENOTINDEFINE

Remove line 207, because the test works with ESDM.

tst_atts: /home/lucy/esiwace/esdm/deps/smd/src/smd-core.c:2619: smd_attr_new_usertype: Assertion 'name != NULL' failed.

//TODO Change the function from assertion to returning something meaninful

Expected error: NC_EBADNAME

Remove lines 235-242. Testing invalid parameters as input.

Remove lines 243-269. Check that the reserved words (NC_GLOBAL) are rejected. ESDM does not have this limitation.

Remove lines 280-317, 327-335, 341-343, 353-355, 365-367, 375. Testing invalid parameters as input.

Remove lines 372, 383, 414, 423. Same problem with smd_attr_copy_value and strings.

tst_atts: /home/lucy/esiwace/esdm/src/esdm-datatypes.c:959: esdmI_dataset_metadata_create: Assertion 'd->dataspace->size != ((void *)0)' failed.

//TODO This error is out of my league.

Remove lines 376-385, 415-424. NetCDF is creating a file with one variable and one attribute. The variable has zero dimensions. ESDM doesn't handle it. So it cannot close and reopen the file. ERROR dimension zero.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.6 Test tst_atts1.c

Test attributes.

// TODO This test deals with the conversion of types. It needs to be tested later.

Status: INCONCLUSIVE!

1.7 Test tst_atts2.c

Test copy of attributes.

ESDM does not support user-defined datatypes from NetCDF!

Remove lines 57-78, 157-212. Specific to compound type.

Remove lines 101-105. Testing invalid parameters as input.

Remove lines 214-259. Specific to enum type.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.8 Test tst_atts3.c

This is a very simple example which writes a netCDF file with Unicode names encoded with UTF-8. It is the NETCDF3 equivalent of tst_unicode.c

Remove lines 2393-2404. Testing invalid parameters as input.

tst_atts3: /home/lucy/esiwace/esdm/src/esdm-datatypes.c:959: esdmI_dataset_metadata_create: Assertion 'd->dataspace->size != ((void *)0)' failed.

//TODO This error is out of my league.

Remove lines 2407-2427. ERROR dimension zero.

//TODO Find file nc_test

Remove lines 2431-2488. NetCDF calls create_file() with no parameters at all. ESDM does not handle it.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.9 Test tst_atts_string_rewrite.c

This test was provided by Jeff Whitaker as an example of a bug, specifically a segfault when re-writing an NC_CHAR attribute as an NC_STRING attribute.

*** Tests successful!

Status: SUCCESS!

1.10 Test tst_attsperf.c

Test the netCDF-4 attribute code.

Too long to run!

Status: INCONCLUSIVE!

1.11 Test tst_bug324.c

No description.

Remove line 71. Expected error: NC_FORMAT_NETCDF4_CLASSIC.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.12 Test tst_camrun.c

This program writes a data file from the CAM model run.

tst_camrun: /home/lucy/esiwace/esdm/src/esdm-datatypes.c:959: esdmI_dataset_metadata_create: Assertion 'd->dataspace->size != ((void *)0)' failed.

//TODO This error is out of my league.

/* rank (number of dimensions) for each variable */

define RANK_P0 0

Remove lines 7390, 7396. ERROR dimension zero.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.13 Test tst_chunks.c

Test netcdf-4 variables.

ESDM does not support compression!

Status: FAILURE!

1.14 Test tst_chunks2.c

Test netcdf-4 chunking.

ESDM does not support compression!

Remove lines 106, 102-121, 137-140, 170-176, 208-214, 246-252, 287-293, 327-333, 369-375, 408-414, 447-453. ESDM does not support compression!.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.15 Test tst_chunks3.c

Runs benchmarks on different chunking sizes.

ESDM does not support compression!

Remove lines 283-296. ESDM does not support compression!

Too long to run!

Status: INCONCLUSIVE!

1.16 Test tst_compounds.c

Test netcdf-4 compound type feature.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

1.17 Test tst_compounds2.c

Test netcdf-4 compound type feature.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

1.18 Test tst_compounds3.c

Test netcdf-4 compound type feature, even more.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

1.19 Test tst_converts.c

Test data conversions and fill value handling.

//TODO once the smd tests are finished

Sorry! Unexpected result, ../../nc_test4/tst_converts.c, line: 147

2 failures

7 errors detected! Sorry!

Status: INCONCLUSIVE!

1.20 Test tst_converts2.c

tst_converts2: /home/lucy/esiwace/esdm/src/hypercube.c:62: esdmI_hypercube_make: Assertion 'offset' failed.

//TODO This error is out of my league.

Remove lines 31-51. ERROR dimension zero.

if (mem_nc_type != type_esdm_to_nc(esdm_dataspace_get_type(space)))

//TODO once the smd tests are finished

Status: INCONCLUSIVE!

1.21 Test tst_coords.c

Test netcdf-4 coordinate variables and dimensions.

//TODO This is a test I would like to make it work.

ESDM does not support groups from NetCDF!

Remove lines 92-93. ESDM does not keep the same order for dimids.

Remove lines 115, 122. ESDM does not support compression!

Remove lines 157, 505. It's not working due to a problem in smd_attr_copy_value and strings.

Remove lines 488, 490, 492, 494. ESDM does not keep the same order for dimids.

Remove lines 537, 539, 781, 785, 786, 790, 792. ESDM does not keep the same order for dimids.

Remove line 583-602. ESDM does not work with groups.

Remove line 606-647. This is actually an interesting problem.

NetCDF inserts two variables with the same name, different type, and that will have the same id.

// if (nc_def_var(ncid, VAR_NAME, NC_USHORT, NDIMS, time_dimids, &time_id) != NC_NOERR) ERR;

// if (nc_def_var(ncid, VAR_NAME, NC_CHAR, NDIMS, time_dimids, &time_id)) ERR;

ESDM does not allow two variables with the same name. This might be a big restriction.

tst_coords: /home/lucy/esiwace/esdm/src/esdm-datatypes.c:382: esdmI_fragment_create: Assertion 'sspace->size[i] > 0' failed.

//TODO This error is out of my league.

Remove lines 711-738. NetCDF is creating a file with one unlimited dimension and one variable. Then a value is assigned to the variable without an update in the dimension (currently zero). ESDM doesn't handle it. ERROR dimension zero.

I was assuming that before using an unlimited dimension its value should be defined. Clearly I was wrong.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.22 Test tst_coords2.c

Test netcdf-4 coordinate variables and dimensions.

ESDM does not support groups from NetCDF!

Remove line 121.

// if (ndims_in != NDIMS) ERR;

//TODO This is a problem that needs to be understood it and fixed.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.23 Test tst_coords3.c

Test netcdf-4 coordinate variables and dimensions with an example from the CF conventions.

Remove lines 136, 138, 140, 144-145, 155-156, 177-178, 186-187, 195-196. ESDM does not keep the same order for dimids.

Remove lines 147, 149, 151, 158, 160, 162, 171, 173, 180, 182, 189, 191, 198, 200. Same problem with smd_attr_copy_value and strings.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.24 Test tst_create_files.c

This program creates a test file.

ESDM does not support groups from NetCDF!

tst_create_files: /home/lucy/esiwace/esdm/src/esdm-scheduler.c:125: backend_thread: Assertion 'status->pending_ops >= 0' failed.

//TODO This error is out of my league.

Too long to run!

Status: INCONCLUSIVE!

1.25 Test tst_dims.c

Test netcdf-4 dimensions.

ESDM does not support groups from NetCDF!

Remove lines 104-106, 115-120, 136-138, 141, 150, 163-164, 190-200, 216-217, 225-227, 225-227, 260-263, 466. Testing invalid parameters as input.

Remove lines 907-909, 947-949, 1106-1108. This is an interesting error related to NC_NAT.

Two float variables were defined.

```
if (nc_def_var(ncid, LAT_NAME, NC_FLOAT, 1, dimids, &lat_varid)) ERR;
```

```
if (nc_def_var(ncid, LON_NAME, NC_FLOAT, 1, dimids, &lon_varid)) ERR;
```

Then, some values with the correct type are attributed.

```
if (nc_put_var_float(ncid, lat_varid, lat)) ERR;
if (nc_put_var_float(ncid, lon_varid, lon)) ERR;
```

After that, the values are retrieved.

When using the specific function nc_get_var_float, it works. When using the general function nc_get_var, this function calls NC_get_var(ncid, varid, ip, NC_NAT) with the NC_NAT parameter.

```
if (nc_get_var(ncid, lat_varid, lat_in)) ERR;
if (nc_get_var_float(ncid, lon_varid, lon_in)) ERR;
```

ESDM cannot understand that the type is correct, because ESDM_get_vars has the following test.

esdm_dispatch.c:1273: if (mem_nc_type != type_esdm_to_nc(esdm_dataspace_get_type(space)))

Eventually this test will change, but even then will not be compatible with NC_NAT. For NetCDF, this is transparent.

tst_dims: /home/lucy/esiwace/esdm/src/esdm-scheduler.c:456:

esdm_scheduler_makeSplitRecommendation: Assertion 'splitFactors[i] >= 1' failed.

//TODO This error is out of my league.

Remove lines 1101-1103. ERROR dimension zero.

```
455     splitFactors[i] = (int64_t)ceil(space->size[i]/targetEdgeLength);
(gdb)
456     eassert(splitFactors[i] >= 1);
(gdb) p splitFactors[i]
$8 = 0
(gdb) p space->size[i]
$9 = 0
(gdb) p i
$10 = 3
(gdb) p targetEdgeLength
$11 = 11.445523142259598
(gdb) p space->dims
$12 = 4
```

NetCDF defines an unlimited dimension and does not fill its value.

/* Write our 4D pressure, elevation, and hp data. But this * should do nothing for pressure and hp, because these are * record vars, and nc_put_var_* doesn't do anything to record * vars, because it doesn't know how big the var is supposed to * be. */

Remove lines 1157-1162. For some reason, the double variable is turning into (int) zero.

```
(gdb) p pres
$1 = {{{{1013.1, 1013.1, 1013.1, 1013.1}, {1013.1, 1013.1, 1013.1, 1013.1},
{1013.1, 1013.1, 1013.1, 1013.1}}, {{1014.1, 1014.1, 1014.1, 1014.1},
{1014.1, 1014.1, 1014.1, 1014.1}, {1014.1, 1014.1, 1014.1, 1014.1}}}}
(gdb) p pres_in
$2 = {{{{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}},
{{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}}}}
```

Remove lines 1165-1168. Conversion from INT64 to UINT64. Eventually it will work.

```
1273   if (mem_nc_type != type_esdm_to_nc(esdm_dataspace_get_type(space))) {
(gdb)
1274     return NC_EBADTYPE;
(gdb) p mem_nc_type
$1 = 11
(gdb) p type_esdm_to_nc(esdm_dataspace_get_type(space))
$2 = 10
```

Remove lines 1171-1175. For some reason, the ushort variable is turning into zero.

```
(gdb) p hp
$1 = {{{100, 101, 102, 103}, {100, 101, 102, 103}}}
(gdb) p hp_in
$2 = {{{0, 0, 0, 0}, {0, 0, 0, 0}}}
```

Remove lines 1257-1258. ESDM does not keep the same order for dimids.

Remove lines 1326-1356. Creates a new file name (file_in) and tries to open it.

```
strcpy(file_in, "");
if (getenv("srcdir"))
{
   strcat(file_in, getenv("srcdir"));
   strcat(file_in, "/");
}
strcat(file_in, REF_FILE_NAME);

if (nc_open(file_in, NC_NOWRITE, &ncid)) ERR;
```

ESDM does not recognize the metadata and cannot open the container.

```
301   status = esdm_container_open(cpath, 0, &e->c);
303   if (status != ESDM_SUCCESS) {
(gdb) p status
$1 = ESDM_ERROR
```

*** Tests successful!

Status: PARTIAL SUCCESS!

1.26 Test tst_dims2.c

Test netcdf-4 dimensions some more.

Remove lines 58-59. ESDM does not keep the same order for dimids.

Remove lines 440-441. Testing invalid parameters as input.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.27 Test tst_dims3.c

Test netcdf-4 dimensions inheritance.

ESDM does not support groups from NetCDF!

Status: FAILURE!

1.28 Test tst_elatefill.c

Test proper elatefill return when fillvalue is assigned outside of the initial define.

line 41 expecting NC_ELATEFILL but got 0

Expected error: NC_ELATEFILL

Status: INCONCLUSIVE!

1.29 Test tst_empty_vlen_unlim.c

This program excersizes HDF5 variable length array code.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

1.30 Test tst_endian_fill.c

Create a test file with fill values for a variable of specified endianness.

ESDM does not support endianness!

Status: FAILURE!

1.31 Test tst_enums.c

Test netcdf-4 enum types.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

1.32 Test tst_files.c

Test netcdf-4 file code.

Remove line 131. Expected error: NC_ERANGE

Remove line 189. Expected error: NC_ENOTINDEFINE

Remove line 279, 303. Expected error: NC_FORMAT_CLASSIC

Remove line 285, 309. Expected error: NC_FORMAT_64BIT_OFFSET

Remove line 291. Expected error: NC_FORMAT_NETCDF4

Remove lines 336-337. ESDM does not support compression!

Remove lines 251-269, 519-527. Expected errors related to the format.

Remove line 534. Expected error: NC_EPERM

Remove lines 479-481, 517-518, 529-530. Testing invalid parameters as input.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.33 Test tst_files3.c

This is a benchmark program which tests file writes with compressed data.

ESDM does not support compression!

Too long to run!

Status: INCONCLUSIVE!

1.34 Test tst_files4.c

Test netcdf-4 file from user-reported error. This code based on an ncgen output.

ESDM does not support groups from NetCDF!

Status: FAILURE!

1.35 Test tst_files5.c

Test netcdf files a bit.

*** Tests successful!

Status: SUCCESS!

1.36 Test tst_files6.c

Test netcdf files a bit.

Remove line 86. Expected error: NC_EHDFERR

*** Tests successful!

Status: PARTIAL SUCCESS!

1.37 Test tst_fill_attr_vanish.c

Based on tst_fillbug.c

Remove lines 91-97. Expected error: NC_ELATEFILL

*** Tests successful!

Status: PARTIAL SUCCESS!

1.38 Test tst_fillbug.c

Test for a bug that Russ found testing fill values.

Remove lines 64-67. Expected error: NC_EINVAL / NC_EBADTYPE

Remove lines 72-74, 83-86. Expected error: NC_FILL_FLOAT

*** Tests successful!

Status: PARTIAL SUCCESS!

1.39 Test tst_fills.c

Create a test file with default fill values for variables of each type.

Remove line 59.

// if (NC_def_var(ncid, STRING_var_NAME, NC_STRING, 0, NULL, &varid)) ERR;

Create file with a 1D string var. Set its fill value to the empty string. I'm not sure this should work with ESDM.

Test using fill values. I don't understand yet.

Status: INCONCLUSIVE!

1.40 Test tst_fills2.c

Create a test file with default fill values for variables of each type.

ESDM does not support NC_STRING from NetCDF!

Status: FAILURE!

1.41 Test tst_filterparser.c

No description.

SUCCESS!!

Status: SUCCESS!

1.42 Test tst_grps.c

Test netcdf-4 group code.

ESDM does not support groups from NetCDF!

Status: FAILURE!

1.43 Test tst_grps2.c

Test netcdf-4 group code some more.

ESDM does not support groups from NetCDF!

Status: FAILURE!

1.44 Test tst_h_refs.c

This program tests fixes for reading netCDF-4 files that contain datasets with reference datatypes. The netCDF-4 library should ignore the datasets and attributes that have reference datatypes and allow the rest of the file to be accessed.

ESDM does not support HDF5 format!

// if (NC_open(FILE_NAME, NC_NOWRITE, &ncid)) ERR;

//TODO Insert a condition in FILE_NAME if the file is .h5

Status: FAILURE!

1.45 Test tst_h_scalar.c

This program tests reading HDF5 files that contain scalar attributes and variables, of both string and numeric datatypes. The netCDF-4 library should allow access to all of these.

ESDM does not support HDF5 format!

Status: FAILURE!

1.46 Test tst_h_strbug.c

This program tests fixes for bugs reported with accessing fixed-length scalar string variables and variable-length scalar string attributes from HDF5 files through the netCDF-4 API.

ESDM does not support HDF5 format!

Status: FAILURE!

1.47 Test tst_h5_endians.c

No description.

ESDM does not support HDF5 format!

ESDM does not support endianness!

Status: FAILURE!

1.48 Test tst_hdf5_file_compat.c

Tests library ability to open files generated by a netcdf instance linked against libhdf5 1.10.0.

ESDM does not support HDF5 format!

Status: FAILURE!

1.49 Test tst_interops.c

Test that HDF5 and NetCDF-4 can read and write the same file.

ESDM does not support HDF5 format!

Status: FAILURE!

1.50 Test tst_interops5.c

Test that HDF5 and NetCDF-4 can read and write the same file.

ESDM does not support HDF5 format!

Status: FAILURE!

1.51 Test tst_interops6.c

Test that HDF5 and NetCDF-4 can read and write the same file.

ESDM does not support HDF5 format!

Status: FAILURE!

1.52 Test tst_large.c

Test netcdf-4 large file fill values.

WARN ESDM_set_fill():503 NOT IMPLEMENTED

Status: INCONCLUSIVE!

1.53 Test tst_large2.c

Test large file problems reported by user.

Too long to run!

Status: INCONCLUSIVE!

1.54 Test tst_mem.c

Test internal netcdf-4 file code.

*** Tests successful!

Status: SUCCESS!

1.55 Test tst_mode.c

Test some illegal mode combinations

ESDM does not support compression!

Remove line 27. Expected error: NC_EINVAL

*** Tests successful!

Status: PARTIAL SUCCESS!

1.56 Test tst_mpi_parallel.c

This just excersizes MPI file I/O to make sure everything's working properly. If this does not work, netcdf/HDF5 parallel I/O also won't work.

*** Tests successful!

Status: SUCCESS!

1.57 Test tst_opaques.c

Test netcdf-4 opaque types.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

1.58 Test tst_parallel.c

This program tests netcdf-4 parallel I/O.

WARN ESDM_var_par_access():520 NOT IMPLEMENTED

Status: INCONCLUSIVE!

1.59 Test tst_put_vars.c

No description.

*** SUCCESS writing example file tst_put_vars.nc!

Status: SUCCESS!

1.60 Test tst_rehash.c

Tests to see if the hashmap is being properly updated.

Tests successful!

Status: SUCCESS!

1.61 Test tst_rename.c

Test renames of vars and dims.

ESDM does not support groups from NetCDF!

Remove lines 482-555. Specific to groups.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.62 Test tst_rename2.c

Test more renames of vars and dims.

Remove lines 110-112. ESDM currently does not work with NC_NAT.

*** Tests successful!

Status: SUCCESS!

1.63 Test tst_simplerw_coll_r.c

This test is for parallel IO and the collective access of metadata with HDF5.

WARN ESDM_var_par_access():520 NOT IMPLEMENTED

Status: INCONCLUSIVE!

1.64 Test tst_strings.c

Test netcdf-4 string types.

ESDM does not support NC_STRING from NetCDF!

Status: FAILURE!

1.65 Test tst_strings2.c

Test netcdf-4 string types.

*** Tests successful!

How is this test successful if ESDM does not support NC_STRING?

Status: SUCCESS!

1.66 Test tst_sync.c

Test netcdf-4 syncs.

Remove line 128. Expected error: NC_EINDEFINE

*** Tests successful!

Status: PARTIAL SUCCESS!

1.67 Test tst_types.c

Test netcdf-4 types.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

1.68 Test tst_udf.c

Test user-defined formats.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

1.69 Test tst_unlim_vars.c

Test netcdf-4 variables with unlimited dimensions.

Sorry! Unexpected result, ../../nc_test4/tst_unlim_vars.c, line: 70

Works, does not work. Who knows...

Status: INCONCLUSIVE!

1.70 Test tst_utf8.c

This is a very simple example which writes a netCDF file with Unicode names encoded with UTF-8. It is the NETCDF3 equivalent of tst_unicode.c

ESDM does not support Unicode names encoded with UTF-8!

//TODO Insert in the function that does this test this message.

Status: FAILURE!

1.71 Test tst_v2.c

Test internal netcdf-4 file code.

*** Tests successful!

Status: SUCCESS!

1.72 Test tst_varms.c

Test netcdf-4 mapped var operations.

*** Tests successful!

Status: SUCCESS!

1.73 Test tst_vars.c

Test netcdf-4 variables.

ESDM does not support groups from NetCDF!

//TODO WARN ESDM_inq_typeid():1549 NOT IMPLEMENTED

/* Open the file and read everything as double. */

Why does this not work?

Great test for the conversions.

Status: INCONCLUSIVE!

1.74 Test tst_vars2.c

Test netcdf-4 variables.

//TODO Understand and implement the fill value.

Remove line 148. Expected error: NC_EPERM

Remove lines 434-439, 522-524, 530-535, 541, 544-551, 638-639, 644-646, 652, 716-725, 744-745, 1060-1062, 1079-1083, 1107-1111, 1232-1259, 1242-1243, 1274-1275, 1291-1318, 1457-1458. Testing invalid parameters as input.

Remove lines 649, 671-672. Conversion.

Remove lines 703, 732-733, 757-758. ESDM does not support endianness!

Remove lines 767-1032, 1145, 1158-1159, 1179-1180. ESDM does not support compression!

Remove line 1270. ESDM does not support user-defined datatypes from NetCDF!

Remove lines 1190-1228. ERROR dimension zero.

Remove lines 1284-1285. ESDM does not support groups from NetCDF!

Remove lines 1286-1287. //TODO Implement function nc_inq_typeids.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.75 Test tst_vars3.c

Test netcdf-4 variables.

Remove lines 115, 117, 119, 123-124, 126-127, 140, 142, 144, 149-150, 152-153, 285. ESDM does not keep the same order for dimids.

Remove lines 159-197. ESDM does not support endianness!

Remove lines 219. ESDM does not support endianness!

Remove lines 229-230, 447. ESDM does not support compression!

Remove lines 269-276. Testing invalid parameters as input.

Remove lines 274-276, 286-288. Expected error: NC_FILL_FLOAT.

/** Default fill value. This is used unless _FillValue attribute * is set. These values are stuffed into newly allocated space as * appropriate. The hope is that one might use these to notice that a * particular datum has not been set. */

Remove lines 311. ESDM does not support compression!

Remove lines 322-363. ESDM does not support groups from NetCDF!

Remove lines 409-558. ESDM does not support szip filter.

*** Tests successful!

Status: PARTIAL SUCCESS!

1.76 Test tst_vars4.c

Test netcdf-4 variables.

ESDM does not support compression!

Remove lines 74-76, 81-82, 96-98, 103-104. ESDM does not support compression!

*** Tests successful!

Status: PARTIAL SUCCESS!

1.77 Test tst_vl.c

Test netcdf-4 variable length code.

ESDM does not support user-defined datatypes from NetCDF!

Status: FAILURE!

File	Not Tested	Not Working Build	Not Working Run	Working
bigmeta.c	✓			
openbigmeta.c	✓			
bm_file.c		✓		
bm_many_atts.c		✓		
bm_many_objs.c		✓		
bm_netcdf4_recs.c		✓		
ref_bzip2.c		✓		
renamegroup.c				✓
test_filter.c		✓		
test_filter_misc.c		✓		
test_szip.c				✓
tst_atts1.c				✓
tst_atts2.c				✓
tst_atts3.c				✓
tst_atts.c				✓
tst_attsperf.c				✓
tst_atts_string_rewrite.c				✓
tst_ar4_3d.c		✓		
tst_ar4_4d.c		✓		
tst_ar4.c		✓		
tst_bug324.c				✓
tst_camrun.c				✓
tst_chunks.c				✓
tst_chunks2.c				✓
tst_chunks3.c				✓
tst_compounds.c			✓	
tst_compounds2.c				✓
tst_compounds3.c				✓
tst_converts.c				✓
tst_converts2.c				✓
tst_coords.c				✓
tst_coords2.c				✓
tst_coords3.c				✓
tst_create_files.c				✓

Table 1.1: List of nc_test4 files – Part I

File	Not Tested	Not Working Build	Not Working Run	Working
tst_dims.c			✓	
tst_dims2.c				✓
tst_dims3.c				✓
tst_elatefill.c				✓
tst_empty_vlen_unlim.c				✓
tst_endian_fill.c				✓
tst_enums.c				✓
tst_files.c				✓
tst_files2.c		✓		
tst_files3.c				✓
tst_files4.c				✓
tst_files5.c				✓
tst_files6.c				✓
tst_fill_attr_vanish.c				✓
tst_fillbug.c				✓
tst_fills.c				✓
tst_fills2.c				✓
tst_filterparser.c				✓
tst_grps.c				✓
tst_grps2.c				✓
tst_h_many_atts.c		✓		
tst_h_refs.c				✓
tst_h_scalar.c				✓
tst_h_strbug.c				✓
tst_h5_endians.c				✓
tst_hdf5_file_compat.c				✓
tst_interops.c				✓
tst_interops4.c			✓	
tst_interops5.c				✓
tst_interops6.c				✓
tst_knmi.c		✓		
tst_large.c				✓
tst_large2.c				✓
tst_mem.c				✓

Table 1.2: List of nc_test4 files – Part II

File	Not Tested	Not Working Build	Not Working Run	Working
tst_mode.c				✓
tst_mpi_parallel.c				✓
tst_nc4perf.c			✓	
tst_opaques.c				✓
tst_parallel.c				✓
tst_parallel3.c			✓	
tst_parallel4.c			✓	
tst_parallel5.c			✓	
tst_put_vars.c				✓
tst_put_vars_two_unlim_dim.c		✓		
tst_rehash.c				✓
tst_rename2.c				✓
tst_rename.c				✓
tst_simplerw_coll_r.c				✓
tst_strings2.c				✓
tst_strings.c				✓
tst_sync.c				✓
tst_types.c				✓
tst_udf.c				✓
tst_unlim_vars.c				✓
tst_utf8.c				✓
tst_utils.c		✓		
tst_v2.c				✓
tst_varms.c				✓
tst_vars2.c				✓
tst_vars3.c				✓
tst_vars4.c				✓
tst_vars.c				✓
tst_vl.c				✓
tst_xplatform2.c			✓	
tst_xplatform.c			✓	
t_type.c				✓

Table 1.3: List of nc_test4 files – Part III

Filename	Success	Partial Success	Inconclusive	Failure	Not Tested Yet
h5testszip.c	✓				
cdm_sea_soundings.c				✓	
t_type.c	✓				
test_szip.c		✓			
tst_atts.c		✓			
tst_atts1.c			✓		
tst_atts2.c		✓			✓
tst_atts3.c		✓			
tst_atts_string_rewrite.c	✓				
tst_attsperf.c					✓
tst_bug324.c		✓			
tst_camrun.c		✓			
tst_chunks.c				✓	
tst_chunks2.c				✓	
tst_chunks3.c				✓	
tst_compounds.c				✓	
tst_compounds2.c				✓	
tst_compounds3.c				✓	
tst_converts.c			✓		
tst_converts2.c			✓		
tst_coords.c		✓			
tst_coords2.c		✓			
tst_coords3.c		✓			
tst_create_files.c			✓		
tst_dims.c		✓			
tst_dims2.c		✓			
tst_dims3.c				✓	
tst_elatefill.c			✓		
tst_empty_vlen_unlim.c				✓	
tst_endian_fill.c				✓	
tst_enums.c				✓	
tst_files.c		✓			
tst_files3.c					✓
tst_files4.c				✓	
tst_files5.c	✓				
tst_files6.c		✓			
tst_fill_attr_vanish.c		✓			
tst_fillbug.c		✓			
tst_fills.c			✓		
tst_fills2.c				✓	
tst_filterparser.c	✓				
tst_grps.c				✓	
tst_grps2.c				✓	

Table 1.4: Status – Part I

Filename	Success	Partial Success	Inconclusive	Failure	Not Tested Yet
tst_h_refs.c				✓	
tst_h_scalar.c				✓	
tst_h_strbug.c				✓	
tst_h5_endians.c				✓	
tst_hdf5_file_compat.c				✓	
tst_interops.c				✓	
tst_interops5.c				✓	
tst_interops6.c				✓	
tst_large.c				✓	
tst_large2.c					✓
tst_mem.c	✓				
tst_mode.c		✓			
tst_mpi_parallel.c	✓				
tst_opaques.c				✓	
tst_parallel.c			✓		
tst_put_vars.c	✓				
tst_rehash.c	✓				
tst_rename.c		✓			
tst_rename2.c	✓				
tst_simplerw_coll_r.c			✓		
tst_strings.c				✓	
tst_strings2.c	✓				
tst_sync.c		✓			
tst_types.c				✓	
tst_udf.c				✓	
tst_unlim_vars.c			✓		
tst_utf8.c				✓	
tst_v2.c	✓				
tst_varms.c	✓				
tst_vars.c			✓		
tst_vars2.c		✓			
tst_vars3.c		✓			
tst_vars4.c		✓			
tst_vl.c				✓	

Table 1.5: Status – Part II

Filename	Output
h5testszip.c	PASS
cdm_sea_soundings.c	ESDM does not support user-defined datatypes from NetCDF!
t_type.c	Tests successful!
test_szip.c	ESDM does not support compression!
tst_atts.c	Tests successful!
tst_atts1.c	
tst_atts2.c	Tests successful!
tst_atts3.c	Tests successful!
tst_atts_string_rewrite.c	Tests successful!
tst_attsperf.c	Too long to run!
tst_bug324.c	Tests successful!
tst_camrun.c	Tests successful!
tst_chunks.c	ESDM does not support compression!
tst_chunks2.c	ESDM does not support compression!
tst_chunks3.c	ESDM does not support compression!
tst_compounds.c	ESDM does not support user-defined datatypes from NetCDF!
tst_compounds2.c	ESDM does not support user-defined datatypes from NetCDF!
tst_compounds3.c	ESDM does not support user-defined datatypes from NetCDF!
tst_converts.c	
tst_converts2.c	
tst_coords.c	Tests successful!
tst_coords2.c	Tests successful!
tst_coords3.c	Tests successful!
tst_create_files.c	Too long to run!
tst_dims.c	Tests successful!
tst_dims2.c	Tests successful!
tst_dims3.c	ESDM does not support groups from NetCDF!
tst_elatefill.c	line 41 expecting NC_ELATEFILL but got 0
tst_empty_vlen_unlim.c	ESDM does not support user-defined datatypes from NetCDF!
tst_endian_fill.c	ESDM does not support endianness!
tst_enums.c	ESDM does not support user-defined datatypes from NetCDF!
tst_files.c	Tests successful!
tst_files3.c	Too long to run!
tst_files4.c	ESDM does not support groups from NetCDF!
tst_files5.c	Tests successful!
tst_files6.c	Tests successful!
tst_fill_attr_vanish.c	Tests successful!
tst_fillbug.c	Tests successful!
tst_fills.c	
tst_fills2.c	ESDM does not support NC_STRING from NetCDF!
tst_filterparser.c	SUCCESS!!
tst_grps.c	ESDM does not support groups from NetCDF!
tst_grps2.c	ESDM does not support groups from NetCDF!

Table 1.6: Output – Part I

Filename	Justification
tst_h_refs.c	ESDM does not support HDF5 format!
tst_h_scalar.c	ESDM does not support HDF5 format!
tst_h_strbug.c	ESDM does not support HDF5 format!
tst_h5_endians.c	ESDM does not support endianness!
tst_hdf5_file_compat.c	ESDM does not support HDF5 format!
tst_interops.c	ESDM does not support HDF5 format!
tst_interops5.c	ESDM does not support HDF5 format!
tst_interops6.c	ESDM does not support HDF5 format!
tst_large.c	
tst_large2.c	Too long to run!
tst_mem.c	Tests successful!
tst_mode.c	Tests successful!
tst_mpi_parallel.c	Tests successful!
tst_opaques.c	ESDM does not support user-defined datatypes from NetCDF!
tst_parallel.c	
tst_put_vars.c	SUCCESS writing example file!
tst_rehash.c	Tests successful!
tst_rename.c	Tests successful!
tst_rename2.c	Tests successful!
tst_simplerw_coll_r.c	
tst_strings.c	ESDM does not support NC_STRING from NetCDF!
tst_strings2.c	Tests successful!
tst_sync.c	Tests successful!
tst_types.c	ESDM does not support user-defined datatypes from NetCDF!
tst_udf.c	ESDM does not support user-defined formats from NetCDF!
tst_unlim_vars.c	
tst_utf8.c	ESDM does not support Unicode names encoded with UTF-8!
tst_v2.c	Tests successful!
tst_varms.c	Tests successful!
tst_vars.c	
tst_vars2.c	Tests successful!
tst_vars3.c	Tests successful!
tst_vars4.c	Tests successful!
tst_vl.c	ESDM does not support user-defined datatypes from NetCDF!

Table 1.7: Output – Part II

NetCDF ERROR	Description
NC_EBADNAME	Attribute or variable name contains illegal characters.
NC_EHDFERR	Error at HDF5 layer.
NC_EINDEFINE	Operation not allowed in define mode.
NC_EINVAL	Invalid Argument.
NC_ELATEFILL	Attempt to define fill value when data already exists.
NC_ENOTINDEFINE	Operation not allowed in data mode.
NC_ENOTVAR	Variable not found.
NC_EPERM	Write to read only.
NC_ERANGE	Math result not representable.
NC_FILL_FLOAT	Default fill value.
NC_FORMAT_64BIT_OFFSET	Format specifier for nc_set_default_format() and returned by nc_in
NC_FORMAT_CLASSIC	Format specifier for nc_set_default_format() and returned by nc_in
NC_FORMAT_NETCDF4	Format specifier for nc_set_default_format() and returned by nc_in
NC_FORMAT_NETCDF4_CLASSIC	Format specifier for nc_set_default_format() and returned by nc_in

Table 1.8: Convertion between ESDM and NetCDF4 Erros.

NetCDF TYPE	Number	ESDM Type	ESDM Representation
NC_BYTE	1	SMD_DTYPE_INT8	int8_t
NC_UBYTE	7	SMD_DTYPE_UINT8	uint8_t
NC_CHAR	2	SMD_DTYPE_CHAR	char
NC_SHORT	3	SMD_DTYPE_INT16	int16_t
NC_USHORT	8	SMD_DTYPE_UINT16	uint16_t
NC_INT	4	SMD_DTYPE_INT32	int32_t
NC_LONG	4	SMD_DTYPE_INT32	int32_t
NC_UINT	9	SMD_DTYPE_UINT32	uint32_t
NC_INT64	10	SMD_DTYPE_INT64	int64_t
NC_UINT64	5	SMD_DTYPE_UINT64	uint64_t
NC_FLOAT	11	SMD_DTYPE_FLOAT	32 bits
NC_DOUBLE	6	SMD_DTYPE_DOUBLE	64 bits

Table 1.9: Convertion between ESDM and NetCDF4 datatypes – Datatypes sorted by size.

NetCDF TYPE	Number	ESDM Type	ESDM Representation
NC_BYTE	1	SMD_DTYPE_INT8	int8_t
NC_CHAR	2	SMD_DTYPE_CHAR	char
NC_SHORT	3	SMD_DTYPE_INT16	int16_t
NC_INT	4	SMD_DTYPE_INT32	int32_t
NC_LONG	4	SMD_DTYPE_INT32	int32_t
NC_UINT64	5	SMD_DTYPE_UINT64	uint64_t
NC_DOUBLE	6	SMD_DTYPE_DOUBLE	64 bits
NC_UBYTE	7	SMD_DTYPE_UINT8	uint8_t
NC_USHORT	8	SMD_DTYPE_UINT16	uint16_t
NC_UINT	9	SMD_DTYPE_UINT32	uint32_t
NC_INT64	10	SMD_DTYPE_INT64	int64_t
NC_FLOAT	11	SMD_DTYPE_FLOAT	32 bits

Table 1.10: Convertion between ESDM and NetCDF4 datatypes – Datatypes sorted by NETCDF4 description.

Functionality	NetCDF Support	ESDM Support
	Datatypes Support	
NC Data Type	NC Description	ESDM Data Type
NC_NAT	NAT = Not A Type (c.f. NaN)	SMD_TYPE_AS_EXPECTED
NC_BYTE	signed 1 byte integer	SMD_DTYPE_INT8
NC_CHAR	ISO/ASCII character	SMD_DTYPE_CHAR
NC_SHORT	signed 2 byte integer	SMD_DTYPE_INT16
NC_INT	signed 4 byte integer	SMD_DTYPE_INT32
NC_LONG	deprecated, but required for backward compatibility	SMD_DTYPE_INT32
NC_FLOAT	single precision floating point number	SMD_DTYPE_FLOAT
NC_DOUBLE	double precision floating point number	SMD_DTYPE_DOUBLE
NC_UBYTE	unsigned 1 byte int	SMD_DTYPE_UINT8
NC_USHORT	unsigned 2-byte int	SMD_DTYPE_UINT16
NC_UINT	unsigned 4-byte int	SMD_DTYPE_UINT32
NC_INT64	signed 8-byte int	SMD_DTYPE_INT64
NC_UINT64	unsigned 8-byte int	SMD_DTYPE_UINT64
NC_STRING	string	SMD_DTYPE_STRING
NC_VLEN	used internally for vlen types	NOT SUPPORTED YET
NC_OPAQUE	used internally for opaque types	NOT SUPPORTED YET
NC_COMPOUND	used internally for compound types	NOT SUPPORTED YET
NC_ENUM	used internally for enum types	NOT SUPPORTED YET

Table 1.11: Functionalities – NetCDF and ESDM – Datatypes Support.

Functionality	NetCDF Support	ESDM Support
	Modes – Creating a file	
NC_CLOBBER	Overwrite existing file	ESDM_CLOBBER
NC_NOCLOBBER	Do not overwrite existing file	ESDM_NOCLOBBER
NC_SHARE	Limit write caching - netcdf classic files only	NOT SUPPORTED
NC_64BIT_OFFSET	Create 64-bit offset file	NOT SUPPORTED
NC_64BIT_DATA	Create CDF-5 file (alias NC_CDF5)	NOT SUPPORTED
NC_NETCDF4	Create netCDF-4/HDF5 file	NOT SUPPORTED
NC_CLASSIC_MODEL	Enforce netCDF classic mode on netCDF-4/HDF5 files	NOT SUPPORTED
NC_DISKLESS	Store data in memory	NOT SUPPORTED
NC_PERSIST	Force the NC_DISKLESS data from memory to a file	NOT SUPPORTED

Table 1.12: Functionalities – NetCDF and ESDM – Modes – Creating a file.

Functionality	NetCDF Support	ESDM Support
	Modes – Opening a file	
NC_NOWRITE	Open the dataset with read-only access	ESDM_NOWRITE
NC_WRITE	Open the dataset with read-write access	ESDM_WRITE
NC_SHARE	Share updates, limit caching	NOT SUPPORTED
NC_DISKLESS	Store data in memory	NOT SUPPORTED
NC_PERSIST	Force the NC_DISKLESS data from memory to a file	NOT SUPPORTED

Table 1.13: Functionalities – NetCDF and ESDM – Modes – Opening a file.

Functionality	NetCDF Support	ESDM Support

Table 1.14: Functionalities – NetCDF and ESDM.

2 Conversion

```
Final Results - INT8 to FLOAT

*****************************************
INT8_MAX = 127
Maximum FLOAT = 126.000000
Maximum INT8 = 126
*****************************************

*****************************************
INT8_MIN = -128
Minimum FLOAT = -127.000000
Minimum INT8 = -127
*****************************************

Final Results - INT16 to FLOAT

*****************************************
INT16_MAX = 32767
Maximum FLOAT = 32766.000000
Maximum INT16 = 32766
*****************************************

*****************************************
INT16_MIN = -32768
Minimum FLOAT = -32767.000000
Minimum INT16 = -32767
*****************************************

Final Results - INT32 to FLOAT

*****************************************
INT32_MAX = 2147483647
Maximum FLOAT = 2147483520.000000
Maximum INT32 = 2147483583
*****************************************

*****************************************
INT32_MIN = -2147483648
Minimum FLOAT = -2147483520.000000
Maximum INT32 = -2147483583
*****************************************
```

```
Final Results - INT64 to FLOAT

*****************************************
INT64_MAX = 9223372036854775807
Maximum FLOAT = 9223371487098961920.000000
Maximum INT64 = 9223371761976865807
*****************************************

*****************************************
INT64_MIN = -9223372036854775808
Minimum FLOAT = -9223371487098961920.000000
Minimum INT64 = -9223371761976865808
*****************************************

Final Results - UINT8 to FLOAT

*****************************************
UINT8_MAX = 255
Maximum FLOAT = 254.000000
Maximum UINT8 = 254
*****************************************

Final Results - UINT16 to FLOAT

*****************************************
UINT16_MAX = 65535
Maximum FLOAT = 65534.000000
Maximum UINT16 = 65534
*****************************************

Final Results - UINT32 to FLOAT

*****************************************
UINT32_MAX = 4294967295
Maximum FLOAT = 4294967040.000000
Maximum UINT32 = 4294967167
*****************************************

Final Results - UINT64 to FLOAT

*****************************************
UINT64_MAX = 18446744073709551615
Maximum FLOAT = 18446742974197923840.000000
Maximum UINT64 = 18446743523953731615
*****************************************

Final Results - INT8 to DOUBLE

*****************************************
INT8_MAX = 127
Maximum DOUBLE = 126.000000
```

```
Maximum INT8 = 126
*****************************************

*****************************************
INT8_MIN = -128
Minimum DOUBLE = -127.000000
Minimum INT8 = -127
*****************************************

Final Results - INT16 to DOUBLE

*****************************************
INT16_MAX = 32767
Maximum DOUBLE = 32766.000000
Maximum INT16 = 32766
*****************************************

*****************************************
INT16_MIN = -32768
Minimum DOUBLE = -32767.000000
Minimum INT16 = -32767
*****************************************

Final Results - INT32 to DOUBLE

*****************************************
INT32_MAX = 2147483647
Maximum DOUBLE = 2147483646.000000
Maximum INT32 = 2147483646
*****************************************

*****************************************
INT32_MIN = -2147483648
Minimum DOUBLE = -2147483647.000000
Minimum INT32 = -2147483647
*****************************************

Final Results - INT64 to DOUBLE

*****************************************
INT64_MAX = 9223372036854775807
Maximum DOUBLE = 9223372036854774784.000000
Maximum INT64 = 9223372036854775295
*****************************************

*****************************************
INT64_MIN = -9223372036854775808
Minimum DOUBLE = -9223372036854774784.000000
Minimum INT64 = -9223372036854775295
*****************************************
```

```
Final Results - UINT8 to DOUBLE

*****************************************
UINT8_MAX = 255
Maximum DOUBLE = 254.000000
Maximum UINT8 = 254
*****************************************

Final Results - UINT16 to DOUBLE

*****************************************
UINT16_MAX = 65535
Maximum DOUBLE = 65534.000000
Maximum UINT16 = 65534
*****************************************

Final Results - UINT32 to DOUBLE

*****************************************
UINT32_MAX = 4294967295
Maximum DOUBLE = 4294967294.000000
Maximum UINT32 = 4294967294
*****************************************

Final Results - UINT64 to DOUBLE

*****************************************
UINT64_MAX = 18446744073709551615
Maximum DOUBLE = 18446744073709549568.000000
Maximum UINT64 = 18446744073709550591
*****************************************
```