



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

*The information and views set out in this report are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.*

# Contents

<b>1</b>	<b>Files From Directory nc_test4</b>	<b>4</b>
1.1	h5testszip.c . . . . .	4
1.2	Test cdm_sea_soundings.c . . . . .	4
1.3	Test t_type.c . . . . .	4
1.4	Test test_szip.c . . . . .	4
1.5	Test tst_atts.c . . . . .	5
1.6	Test tst_atts1.c . . . . .	5
1.7	Test tst_atts2.c . . . . .	6
1.8	Test tst_atts3.c . . . . .	6
1.9	Test tst_atts_string_rewrite.c . . . . .	6
1.10	Test tst_attsperf.c . . . . .	7
1.11	Test tst_bug324.c . . . . .	7
1.12	Test tst_camrun.c . . . . .	7
1.13	Test tst_chunks.c . . . . .	7
1.14	Test tst_chunks2.c . . . . .	8
1.15	Test tst_chunks3.c . . . . .	8
1.16	Test tst_compounds.c . . . . .	8
1.17	Test tst_compounds2.c . . . . .	8
1.18	Test tst_compounds3.c . . . . .	9
1.19	Test tst_converts.c . . . . .	9
1.20	Test tst_converts2.c . . . . .	9
1.21	Test tst_coords.c . . . . .	9
1.22	Test tst_coords2.c . . . . .	10
1.23	Test tst_coords3.c . . . . .	11
1.24	Test tst_create_files.c . . . . .	11
1.25	Test tst_dims.c . . . . .	11
1.26	Test tst_dims2.c . . . . .	14
1.27	Test tst_dims3.c . . . . .	14
1.28	Test tst_ellatefill.c . . . . .	14
1.29	Test tst_empty_vlen_unlim.c . . . . .	14
1.30	Test tst_endian_fill.c . . . . .	14
1.31	Test tst_enums.c . . . . .	15
1.32	Test tst_files.c . . . . .	15
1.33	Test tst_files3.c . . . . .	15
1.34	Test tst_files4.c . . . . .	15
1.35	Test tst_files5.c . . . . .	16
1.36	Test tst_files6.c . . . . .	16
1.37	Test tst_fill_attr_vanish.c . . . . .	16
1.38	Test tst_fillbug.c . . . . .	16
1.39	Test tst_fills.c . . . . .	16
1.40	Test tst_fills2.c . . . . .	17
1.41	Test tst_filterparser.c . . . . .	17
1.42	Test tst_grps.c . . . . .	17
1.43	Test tst_grps2.c . . . . .	17
1.44	Test tst_h_refs.c . . . . .	18

---

1.45	Test <code>tst_h_scalar.c</code> . . . . .	18
1.46	Test <code>tst_h_strbug.c</code> . . . . .	18
1.47	Test <code>tst_h5_endians.c</code> . . . . .	18
1.48	Test <code>tst_interops5.c</code> . . . . .	19
1.49	Test <code>tst_interops6.c</code> . . . . .	19
1.50	Test <code>tst_interops.c</code> . . . . .	19
1.51	Test <code>tst_large.c</code> . . . . .	19
1.52	Test <code>tst_large2.c</code> . . . . .	19
1.53	Test <code>tst_mem.c</code> . . . . .	20
1.54	Test <code>tst_mode.c</code> . . . . .	20
1.55	Test <code>tst_mpi_parallel.c</code> . . . . .	20
1.56	Test <code>tst_opaques.c</code> . . . . .	20
1.57	Test <code>tst_parallel.c</code> . . . . .	20
1.58	Test <code>tst_put_vars.c</code> . . . . .	21
1.59	Test <code>tst_rehash.c</code> . . . . .	21
1.60	Test <code>tst_rename.c</code> . . . . .	21
1.61	Test <code>tst_rename2.c</code> . . . . .	21
1.62	Test <code>tst_simplerw_coll_r.c</code> . . . . .	21
1.63	Test <code>tst_strings.c</code> . . . . .	22
1.64	Test <code>tst_strings2.c</code> . . . . .	22
1.65	Test <code>tst_sync.c</code> . . . . .	22
1.66	Test <code>tst_types.c</code> . . . . .	22
1.67	Test <code>tst_udf.c</code> . . . . .	22
1.68	Test <code>tst_unlim_vars.c</code> . . . . .	23
1.69	Test <code>tst_utf8.c</code> . . . . .	23
1.70	Test <code>tst_v2.c</code> . . . . .	23
1.71	Test <code>tst_varms.c</code> . . . . .	23
1.72	Test <code>tst_vars.c</code> . . . . .	23
1.73	Test <code>tst_vars2.c</code> . . . . .	24
1.74	Test <code>tst_vars3.c</code> . . . . .	24
1.75	Test <code>tst_vars4.c</code> . . . . .	25
1.76	Test <code>tst_vl.c</code> . . . . .	25

# 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 compliance 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 meaningful
```

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}, {1013.1, 1013.1, 1013.1, 1013.1}}}
(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}, {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) php_in2 = 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 test.

**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.

**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!

```
// 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!

**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!

**Status:** FAILURE!

### 1.47 Test `tst_h5_endians.c`

No description.

ESDM does not support HDF5!

ESDM does not support endianness!

**Status:** FAILURE!

### **1.48 Test `tst_interops5.c`**

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

ESDM does not support HDF5!

**Status:** FAILURE!

### **1.49 Test `tst_interops6.c`**

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

ESDM does not support HDF5!

**Status:** FAILURE!

### **1.50 Test `tst_interops.c`**

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

ESDM does not support HDF5!

**Status:** FAILURE!

### **1.51 Test `tst_large.c`**

Test netcdf-4 large file fill values.

WARN ESDM\_set\_fill():503 NOT IMPLEMENTED

**Status:** INCONCLUSIVE!

### **1.52 Test `tst_large2.c`**

Test large file problems reported by user.

Too large to test.

**Status:** INCONCLUSIVE!

### 1.53 Test `tst_mem.c`

Test internal netcdf-4 file code.

\*\*\* Tests successful!

**Status:** SUCCESS!

### 1.54 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!

### 1.55 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!

### 1.56 Test `tst_opaques.c`

Test netcdf-4 opaque types.

ESDM does not support user defined datatypes from NetCDF!

**Status:** FAILURE!

### 1.57 Test `tst_parallel.c`

This program tests netcdf-4 parallel I/O.

WARN ESDM\_var\_par\_access():520 NOT IMPLEMENTED

**Status:** INCONCLUSIVE!

### 1.58 Test `tst_put_vars.c`

No description.

\*\*\* SUCCESS writing example file `tst_put_vars.nc`!

**Status:** SUCCESS!

### 1.59 Test `tst_rehash.c`

Tests to see if the hashmap is being properly updated.

Tests successful!

**Status:** SUCCESS!

### 1.60 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!

### 1.61 Test `tst_rename2.c`

Test more renames of vars and dims.

\*\*\* Tests successful!

**Status:** SUCCESS!

## 1.62 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!

## 1.63 Test `tst_strings.c`

Test netcdf-4 string types.

ESDM does not support NC\_STRING!

**Status:** FAILURE!

## 1.64 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!

## 1.65 Test `tst_sync.c`

Test netcdf-4 syncs.

Remove line 128. Expected error: NC\_EINDEFINE

\*\*\* Tests successful!

**Status:** PARTIAL SUCCESS!

## 1.66 Test `tst_types.c`

Test netcdf-4 types.

ESDM does not support user defined datatypes from NetCDF!

**Status:** FAILURE!

### 1.67 Test `tst_udf.c`

Test user-defined formats.

ESDM does not support user defined datatypes from NetCDF!

**Status:** FAILURE!

### 1.68 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!

### 1.69 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!

### 1.70 Test `tst_v2.c`

Test internal netcdf-4 file code.

\*\*\* Tests successful!

**Status:** SUCCESS!

### 1.71 Test `tst_varms.c`

Test netcdf-4 mapped var operations.

\*\*\* Tests successful!

**Status:** SUCCESS!

## 1.72 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!

## 1.73 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!

## 1.74 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!

## 1.75 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!

## 1.76 Test `tst_vl.c`

Test netcdf-4 variable length code.

ESDM does not support user defined datatypes from NetCDF!

**Status:** FAILURE!



Test Name	Test Objective	Steps	Expected Output	Actual Output	Status
h5testszip.c					
cdm_sea_soundings.c					
test_szip.c					
tst_atts1.c					
tst_atts2.c					
tst_atts3.c					
tst_atts.c					
tst_atts_string_rewrite.c					
tst_attsperf.c					
tst_bug324.c					
tst_camrun.c					
tst_chunks2.c					
tst_chunks3.c					
tst_chunks.c					
tst_compounds2.c					
tst_compounds3.c					
tst_converts2.c					
tst_converts.c					
tst_coords2.c					
tst_coords3.c					
tst_coords.c					
tst_create_files.c					
tst_dims2.c					
tst_dims3.c					
tst_elatefill.c					
tst_empty_vlen_unlim.c					
tst_endian_fill.c					
tst_enums.c					
tst_files3.c					
tst_files5.c					
tst_files4.c					
tst_files6.c					
tst_files.c					
tst_fill_attr_vanish.c					
tst_fillbug.c					
tst_fills2.c					
tst_fills.c					
tst_filterparser.c					
tst_grps2.c					
tst_grps.c					
tst_h5_endians.c					
tst_hdf5_file_compat.c					
tst_h_refs.c					
tst_h_scalar.c					
tst_h_strbug.c					
tst_interops5.c					
tst_interops6.c					
tst_interops.c					
tst_large2.c					
tst_large.c					

Filename	Compatible	Almost Compatible	Not Compatible Yet	Almost Incompatible	Incompatible
tst_mem.c					
tst_mode.c					
tst_mpi_parallel.c					
tst_opaques.c					
tst_parallel.c					
tst_put_vars.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_v2.c					
tst_varms.c					
tst_vars2.c					
tst_vars3.c					
tst_vars4.c					
tst_vars.c					
tst_vl.c					
t_type.c					

Table 1.2: Test Case Report: ESDM and NetCDF compatibility

Filename	Compatible	Almost Compatible	Not Compatible Yet	Almost Incompatible	Incompatible
cdm_sea_soundings.c					✓
test_szip.c					✓
tst_atts1.c					
tst_atts2.c					✓
tst_atts3.c					
tst_atts.c					
tst_attspref.c					
tst_bug324.c					
tst_camrun.c					
tst_chunks2.c					
tst_chunks3.c					
tst_chunks.c					
tst_compounds2.c					
tst_compounds3.c					
tst_converts2.c					
tst_converts.c					
tst_coords2.c					
tst_coords3.c					
tst_coords.c					
tst_create_files.c					
tst_dims2.c					
tst_dims3.c					
tst_empty_vlen_unlim.c					
tst_endian_fill.c					
tst_enums.c					
tst_files4.c					
tst_files6.c					
tst_files.c					
tst_fill_attr_vanish.c					
tst_fillbug.c					
tst_fills2.c					
tst_fills.c					
tst_grps2.c					
tst_grps.c					
tst_h5_endians.c					
tst_hdf5_file_compat.c					
tst_h_refs.c					
tst_h_scalar.c					
tst_h_strbug.c					
tst_interops5.c					
tst_interops6.c					
tst_interops.c					
tst_large2.c					
tst_large.c					

Table 1.3

Filename	Compatible	Almost Compatible	Not Compatible Yet	Almost Incompatible	Incompatible
tst_mem.c					
tst_mode.c					
tst_opaques.c					
tst_parallel.c					
tst_put_vars.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_v2.c					
tst_varms.c					
tst_vars2.c					
tst_vars3.c					
tst_vars4.c					
tst_vars.c					
tst_vl.c					
h5testszip.c	✓				
tst_atts_string_rewrite.c	✓				
tst_elatefill.c	✓				
tst_files3.c	✓				
tst_files5.c	✓				
tst_filterparser.c	✓				
tst_mpi_parallel.c	✓				
tst_rehash.c	✓				
t_type.c	✓				

Table 1.4

Filename	Justification
cdm_sea_soundings.c	ESDM does not support user defined datatypes from NetCDF!
test_szip.c	ESDM does not support compression!
tst_atts1.c	
tst_atts2.c	ESDM does not support user defined datatypes from NetCDF!
tst_atts3.c	
tst_atts.c	
tst_attspref.c	Too long to test now.
tst_bug324.c	ESDM does not support classic format from NetCDF!
tst_camrun.c	
tst_chunks2.c	ESDM does not support compression!
tst_chunks3.c	ESDM does not support compression!
tst_chunks.c	ESDM does not support compression!
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_converts2.c	
tst_converts.c	
tst_coords2.c	ESDM does not support groups from NetCDF!
tst_coords3.c	ESDM does not support groups from NetCDF!
tst_coords.c	ESDM does not support groups from NetCDF!
tst_create_files.c	ESDM does not support groups from NetCDF!
tst_dims2.c	ESDM does not preserve the ID of the dimensions once it is closed!
tst_dims3.c	ESDM does not support groups from NetCDF!
tst_empty_vlen_unlim.c	ESDM does not support user-defined datatypes from NetCDF!
tst_endian_fill.c	ESDM does not support endianness from NetCDF!
tst_enums.c	ESDM does not support user-defined datatypes from NetCDF!
tst_files4.c	ESDM does not support groups from NetCDF!
tst_files6.c	ESDM does not support HDF5 file with circular group structure!
tst_files.c	
tst_fill_attr_vanish.c	WARN ESDM_set_fill():370 NOT IMPLEMENTED
tst_fillbug.c	WARN_NOT_IMPLEMENTED
tst_fills2.c	WARN_NOT_IMPLEMENTED
tst_fills.c	WARN_NOT_IMPLEMENTED
tst_grps2.c	ESDM does not support groups from NetCDF!
tst_grps.c	ESDM does not support groups from NetCDF!
tst_h5_endians.c	ESDM does not support endianness from NetCDF!
tst_hdf5_file_compat.c	ESDM does not support HDF5 format!
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_interops5.c	ESDM does not support HDF5 format!
tst_interops6.c	ESDM does not support HDF5 format!
tst_interops.c	ESDM does not support HDF5 format!
tst_large2.c	Broke my computer
tst_large.c	WARN ESDM_set_fill():526 NOT IMPLEMENTED (Tests successful!)

Table 1.5

Filename	Justification
tst_mem.c	Tests successful!
tst_mode.c	ESDM does not support compression! (Tests successful!)
tst_opaques.c	ESDM does not support user-defined datatypes from NetCDF!
tst_parallel.c	Not tested yet.
tst_put_vars.c	SUCCESS writing example file!
tst_rename2.c	ESDM does not support NC_NAT from NetCDF!
tst_rename.c	It works until get to a function requiring groups!
tst_simplerw_coll_r.c	ESDM does not support collective access of metadata with HDF5!
tst_strings2.c	Tests successful!
tst_strings.c	ESDM does not support NC_STRING from NetCDF!
tst_sync.c	It does not work just because ESDM does not use the NC_EINDEFINE flag!
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	Sometimes works, sometimes does not work!
tst_utf8.c	ESDM does not support Unicode names encoded with UTF-8!
tst_v2.c	Tests successful!
tst_varms.c	ESDM does not support mapped var operations!
tst_vars2.c	ESDM does not support NC_NAT from NetCDF!
tst_vars3.c	
tst_vars4.c	ESDM does not support compression!
tst_vars.c	ESDM does not support groups from NetCDF!
tst_vl.c	ESDM does not support user-defined datatypes from NetCDF!
h5testszip.c	PASS
tst_atts_string_rewrite.c	Tests successful!
tst_elatefill.c	line 41 expecting NC_ELATEFILL but got 0
tst_files3.c	Working ESDM does not support compression!
tst_files5.c	Not working anymore (Tests successful!)
tst_filterparser.c	SUCCESS!!
tst_mpi_parallel.c	Tests successful!
tst_rehash.c	Tests successful!
t_type.c	Tests successful!

Table 1.6: Additional comments on ESDM and NETCDF compatibility



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		✓		
test_filter.c		✓		
test_filter_misc.c		✓		
tst_ar4_3d.c		✓		
tst_ar4_4d.c		✓		
tst_ar4.c		✓		
tst_files2.c		✓		
tst_h_many_atts.c		✓		
tst_knmi.c		✓		
tst_put_vars_two_unlim_dim.c		✓		
tst_utils.c		✓		
tst_compounds.c			✓	
tst_dims.c			✓	
tst_interops4.c			✓	
tst_nc4perf.c			✓	
tst_parallel3.c			✓	
tst_parallel4.c			✓	
tst_parallel5.c			✓	
tst_xplatform2.c			✓	
tst_xplatform.c			✓	

Table 1.7: List of nc\_test4 files.

File	Not Tested	Not Working Build	Not Working Run	Working
renamegroup.c				✓
test_szip.c				✓
tst_atts1.c				✓
tst_atts2.c				✓
tst_atts3.c				✓
tst_atts.c				✓
tst_attspref.c				✓
tst_atts_string_rewrite.c				✓
tst_bug324.c				✓
tst_camrun.c				✓
tst_chunks2.c				✓
tst_chunks3.c				✓
tst_chunks.c				✓
tst_compounds2.c				✓
tst_compounds3.c				✓
tst_converts2.c				✓
tst_converts.c				✓
tst_coords2.c				✓
tst_coords3.c				✓
tst_coords.c				✓
tst_create_files.c				✓
tst_dims2.c				✓
tst_dims3.c				✓
tst_elatefill.c				✓
tst_empty_vlen_unlim.c				✓
tst_endian_fill.c				✓
tst_enums.c				✓
tst_files3.c				✓
tst_files4.c				✓
tst_files5.c				✓
tst_files6.c				✓
tst_files.c				✓
tst_fill_attr_vanish.c				✓
tst_fillbug.c				✓
tst_fills2.c				✓
tst_fills.c				✓
tst_filterparser.c				✓
tst_grps2.c				✓
tst_grps.c				✓
tst_h5_endians.c				✓
tst_hdf5_file_compat.c				✓
tst_h_refs.c				✓
tst_h_scalar.c				✓
tst_h_strbug.c				✓
tst_interops5.c				✓
tst_interops6.c				✓
tst_interops.c				✓
tst_large2.c				✓
tst_large.c				✓
tst_mem.c				✓

Table 1.6: List of nc-test4 files.

File	Not Tested	Not Working Build	Not Working Run	Working
tst_mode.c				✓
tst_mpi_parallel.c				✓
tst_opaques.c				✓
tst_parallel.c				✓
tst_put_vars.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_v2.c				✓
tst_varms.c				✓
tst_vars2.c				✓
tst_vars3.c				✓
tst_vars4.c				✓
tst_vars.c				✓
tst_vl.c				✓
t_type.c				✓

Table 1.9: List of nc.test4 files.

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 1.10: Conversion 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 1.11: Conversion between ESDM and NetCDF4 datatypes – Datatypes sorted by NETCDF4 description.

Functionality	NetCDF Support	ESDM Support
	Datatypes	
NC Data Type	NC Description	ESDM Data Type
NC_NAT	NAT = Not A Type (c.f. NaN)	NOT SUPPORTED
NC_BYTE	signed 1 byte integer	SMD_DTYPE_INT1
NC_CHAR	ISO/ASCII character	SMD_DTYPE_CHAR
NC_SHORT	signed 2 byte integer	SMD_DTYPE_INT2
NC_INT	signed 4 byte integer	SMD_DTYPE_INT4
NC_LONG	deprecated, but required for backward compatibility	SMD_DTYPE_INT4
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_UINT1
NC_USHORT	unsigned 2-byte int	SMD_DTYPE_UINT2
NC_UINT	unsigned 4-byte int	SMD_DTYPE_UINT4
NC_INT64	signed 8-byte int	SMD_DTYPE_INT8
NC_UINT64	unsigned 8-byte int	SMD_DTYPE_UINT8
NC_STRING	string	SMD_DTYPE_STRING
NC_VLEN	used internally for vlen types	NOT SUPPORTED
NC_OPAQUE	used internally for opaque types	NOT SUPPORTED
NC_COMPOUND	used internally for compound types	NOT SUPPORTED
NC_ENUM	used internally for enum types	NOT SUPPORTED
	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
	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_WRITE—NC_SHARE	Open the dataset with read-write access 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 1.12: Functionality Supported by NetCDF and ESDM.