

Excessive memory usage and file issues due to objects being left open

Leaving objects open in an HDF5 application can cause excessive memory usage and potentially corrupt an HDF5 file.

If objects in an HDF5 file (group, dataset, attribute, ...) are not closed, then the file does not get closed, which can cause problems. If secondary objects (dataspaces, property lists ...) are left open, excessive memory usage may occur.

There are some things you can do to help resolve the issue:

1. Use `H5P_SET_FCLOSE_DEGREE` to do a *strong* close of a file. See the example program `h5close.c`.
2. Call `H5_CLOSE` to close everything.
3. Call `H5F_GET_OBJ_COUNT` to get the number of open file object identifiers, and then call `H5F_GET_OBJ_IDS` to get the list of open object identifiers. See the example program: `h5ckopen.c`

Note that these routines do not work with secondary objects such as dataspaces, property lists, and datatypes.

If you are certain that all object identifiers are closed, and you continue to see excessive memory usage, try using `H5_GARBAGE_COLLECT` and /or `H5_SET_FREE_LIST_LIMITS`, or compiling the library with free lists disabled, to see if free lists are involved. Free lists can be turned off by building HDF5 with `--enable-using-memchecker=yes` (using `configure`) or `HDF5_ENABLE_USING_MEMCHECKER` in `CMake`.