

# RFC: Enhancement to the tool *h5clear*

Vailin Choi

---

John Mainzer

---

The jira issue HDFS-10360 requested an enhancement to the tool *h5clear* so that the user can set the file's EOA (end-of-file address) stored in the superblock. This issue was a result of HDFS-10347, which reported that *h5clear* corrupted a user file and data was not accessible.

This RFC describes the cause and the solution for these jira issues.

---

## Introduction

For jira issue HDFS-10347, the user reported that an HDF5 file was corrupted after using *h5clear* on the file. The situation was described as below:

- a. The user had a crashed file, which was opened with SWMR-write access.
- b. When opening the file with SWMR-read access, data can be read without problem.
- c. *h5dump* was unable to open the file.
- d. *h5clear* was used to clear the *status\_flags* field for the file.
- e. After step #d, the user encountered the following problems:
  - *h5dump* printed the file's metadata but was unable to print the data.
  - When opening the file with SWMR-read access again, there was error stack from the library indicating that the data could not be read.

The investigation of this jira issue noted that the crashed file's EOA (end-of-file address) stored in the superblock was smaller than the file's actual EOF (end-of-file). When *h5clear* closed the file, the library truncated the file to the stored EOA, thus causing the errors listed in step #e above. Therefore, HDFS-10360 is filed to enhance the tool *h5clear* so that the user can set the stored EOA in the file's superblock and can avoid file truncation when not desired.

Also, the reason for a file opened with SWMR-read access (step #b) can read the data whereas *h5dump* cannot do so (step #c) was because the library allowed data access past the file's EOA for SWMR-read.

This RFC proposes adding two new options to the tool as described below.

---

## Current implementation

There are two existing options to the tool *h5clear*:

- *-s*
  - Clear the *status\_flags* field in the file's superblock
- *-m*
  - Remove the existing metadata cache image from the file

## New options

We will introduce two new options to the tool *h5clear*:

- *--increment=C*
  - Set the file's EOA to the maximum of (EOA, EOF) + C
  - C is an integer  $\geq 0$ ; it is optional and will default to one megabytes when not set
  - On file close, the EOA and EOF will be the same
- *--filesize*
  - Print the file's EOA and EOF
  - Cannot combine with other options since the file is opened read-only for this option

## New private file access properties

We will create two new private file access properties, which will only be used internally by the tool. They will be set according to the input options before the tool opens the file:

- *H5F\_ACS\_SKIP\_EOF\_CHECK\_NAME*
  - Set with the *--increment* or *--filesize* options
- *H5F\_ACS\_NULL\_FSM\_ADDR\_NAME*
  - Set with the *--increment* option

When the library opens the file, it will do the following:

- When the file has persistent free-space managers:
  - It is possible that the crashed file has outdated free-space info and might create an inconsistent state when loading the free-space managers and when using the free-space recorded in the managers. If the *H5F\_ACS\_NULL\_FSM\_ADDR\_NAME* property is set, the library will nullify the free-space managers' addresses stored in the superblock extension FSINFO message, thus dropping all free-space info to the floor to avoid

possible confusion and inconsistency.

- When the file's stored EOA is greater than the actual EOF:
  - The library has an existing check on whether the file's stored EOA is greater than the actual EOF. It will return error saying the file is truncated if the check is true. If the `H5F_ACS_SKIP_EOF_CHECK_NAME` property is set, the library will skip this check so as to allow the tool to print out or increment the EOA.

## Additions to the API

We will introduce two new public routines, which will be used by the tool.

### **H5Fincrement\_filesize(hid\_t fid, hsize\_t increment)**

This routine sets the file's EOA to the maximum of (EOA, EOF) + *increment*.

The parameters are:

- *fid*: the file identifier
- *increment*: the number of bytes to be added to the maximum of (EOA, EOF)

Note that this routine will only work for virtual file drivers that are SWMR-compatible.

### **H5Fget\_eoa(hid\_t fid, haddr\_t \*eoa)**

This routine retrieves the file's EOA and returns it in the parameter *eoa*.

The parameters are:

- *fid*: the file identifier
- *eoa*: the file's EOA

Note that this routine will only work for virtual file drivers that are SWMR-compatible.

## Testing

- Add tests to `tools/test/misc/testh5clear.sh.in` to verify the expected output from the tool's two new options.
- Add tests to `test/tfile.c` to verify the expected output from the two new public routines.
- Test files to use for testing (see `test/misc/h5clear_gentest.c` for details):
  - Generate HDF5 files with/without persisting free-space and with/without user block. Then write different EOA values (smaller or greater than EOF) to the stored EOA location.
  - Generate an HDF5 file based on the user-supplied test file attached to the jira issue H5FFV-10347: TEST\_CPS\_PF1\_QVAL\_sdn\_archiver\_1512642969571559452.h5. This test file has an EOA stored in the superblock that is less than the EOF.

- o Generate an HDF5 file with persisting free-space and then just exit without flushing and closing the file.

## Revision History

*February 22, 2018*      Version 0 - Initial draft

*April 20, 2018*      Version 1—revised according to implementation