HDF5 File Space Management

Introduced with

HDF5 Release 1.10.0

The HDF Group

http://www.HDFGroup.org

Copyright Notice and License Terms for HDF5 (Hierarchical Data Format 5) Software Library and Utilities

Contributors: National Center for Supercomputing Applications (NCSA) at the University of Illinois, Fortner Software, Unidata Program Center (netCDF), The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler (gzip), and Digital Equipment Corporation (DEC).

Portions of HDF5 were developed with support from the Lawrence Berkeley National Laboratory (LBNL) and the United States Department of Energy under Prime Contract No. DE-AC02-05CH11231.

Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL). The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials:

> This work was partially produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL.

Contents

1. Introduction

The space within an HDF5 file is called its *file space*. When a user first creates an HDF5 file, the HDF5 Library (also referred to in this document as *the library*) immediately allocates space to store file metadata. *File metadata* is information the library uses to describe the HDF5 file and to identify its associated objects. When a user subsequently creates HDF5 objects, the library allocates file space to store data values and the necessary additional file metadata. When a user removes HDF5 objects from an HDF5 file, the space associated with those objects becomes *free space*. The library manages this free space. The library's *file space management* activities encompass both the allocation of space and the management of free space.

The library has a variety of mechanisms that allow it to implement several different *file space management strategies*. Users can select a strategy when they create an HDF5 file. Depending on the file's usage patterns, one strategy may be better than the others. Users of HDF5 files that have large datasets added and removed on a regular basis might prefer one strategy while users of HDF5 files that are fairly static might prefer a different strategy.

This document describes the available file space allocation mechanisms and strategies, the tools - API and command line - that are available to set or change a strategy, and how the file space management strategies affect the file size and access time for various HDF5 file usage patterns.

1.1. Definitions and Concepts

The following are some terms and concepts used in this document.

Session

A session is composed of the actions on a file from when it is opened until it is closed. The next time a file is opened would be considered another session. Depending on the file space management strategy that was chosen for the file, file space information might be tracked during the current session, tracked over multiple sessions, or not tracked during any session.

Tracked Free Space and Unaccounted Space

Space within an HDF5 file is released when an object is removed from the file. The freed space is monitored while the file is open and is reported as *tracked free space*. Depending on the file space management strategy that was chosen for the file, the tracked free space may be stored after the file is closed (after the end of the current session). If the tracked free space is not stored after the file is closed, then this free space will be considered unaccounted space the next time the file is open.

Tracked free space is sometimes referred to as tracked space.

Tracked free space and unaccounted space are reported in the output of the command line `h5stat -S`.

Tracked free space and unaccounted space can be reclaimed with the h5repack tool.

VFD

VFD is short for Virtual File Driver.

Raw Data

Raw data are the data values in HDF5 dataset objects. For example, in a file that holds weather data, the raw data might include temperatures at different locations and at a variety of times.

File Metadata

File metadata is information the library uses to describe the HDF5 file and to identify its associated objects. One example is the file space management strategy used by a file. The strategy is stored in file metadata. For more information, see the "HDF5 Metadata" paper.

2. File Space Allocation Mechanisms

The HDF5 Library has three different mechanisms for allocating space to store file metadata and raw data. These are described in the sections below.

2.1. Free-Space Manager

The HDF5 Library's free-space manager tracks sections in the HDF5 file that are not being used to store file metadata or raw data. These sections will be of various sizes. When the library needs to allocate space, the free-space manager searches the tracked free space for a section of the appropriate size to fulfill the request. If a suitable section is found, the allocation can be made from the file's existing free space. If the free-space manager cannot fulfill the request, the request falls through to the aggregator level.

2.2. Aggregators

The HDF5 Library has two aggregators. Each aggregator manages a block of contiguous bytes in the file that has not been allocated previously. One aggregator allocates space for file metadata from the block it manages; the other aggregator handles allocations for raw data. The maximum number of bytes in each aggregator's block is tunable.

If the library's allocation request exceeds the maximum number of bytes an aggregator's block can contain, the aggregator cannot fulfill the request, and the request falls through to the virtual file driver level.

After space has been allocated from an aggregator's block, that space is no longer managed by the aggregator. If at some point in the future that space is freed, then the free-space manager would be in charge of the space and not the aggregator. In other words, the freed space would not revert back to the aggregator. Unallocated bytes in the block continue to be managed by the aggregator.

If an aggregator does not have enough space in its block to fulfill a request, it will then request a new block of contiguous bytes from the virtual file driver. Any unallocated space from the old block will become free space.

2.3. Virtual File Driver

The HDF5 Library's virtual file driver (VFD) interface dispatches requests for additional space to the allocation routine of the file driver associated with an HDF5 file. For example, if the POSIX file driver, `H5FD_SEC2`, is being used, its allocation routine will increase the size of the single file on disk that stores the HDF5 file contents to accommodate the additional space that was requested. For more information on VFDs, see the "Alternate File Storage Layouts and Low-level File Drivers" section in "The HDF5 File" chapter in the *HDF5 User's Guide*.

3. File Space Management Strategies

The file space allocation mechanisms described above can be used to implement a variety of file space management strategies. The strategies differ in two main ways: when the library will track free space and how many of the mechanisms the library will use to allocate space for file metadata and raw data. The strategies are listed in the table below and are described in more detail in the sections following the table.

Table 1. File space management strategies

Strategy Name	The strategy might be useful under these conditions:	Implementation Comments
All Persist	Use with files where raw data and metadata are added and removed frequently and where the files are opened and closed frequently. Maximizes the use of space in a file over any number of sessions.	• Uses all of the file space allocation mechanisms • Tracks file free space across sessions
All	Use with files where raw data and metadata are added and removed frequently. Maximizes the use of space in a file during a single session.	• Uses all of the file space allocation mechanisms • Tracks file free space only in the current session
Aggregator VFD	Use with files where small datasets might be added and where few if any datasets are removed. Adding small datasets means the library can take advantage of the aggregators. Maximizes rate at which small datasets are written to the file.	• Uses the aggregator and VFD mechanisms • Never tracks free space
VFD Only	Use with files where large amounts of raw data are added and where few if any datasets are removed. Maximizes rate at which data is written to the file.	• Uses only the VFD mechanism • Never tracks free space

For more information on implementing one of these strategies, see the "Setting or Changing a Strategy" section on page 11.

3.1. The All Persist Strategy

The aim of the All Persist strategy is to maximize the use of space within an HDF5 file over a number of sessions.

With the All Persist strategy, the HDF5 Library's free-space manager tracks the free space that results from manipulating HDF5 objects in an HDF5 file. The tracked free space information is saved when the HDF5 file is closed and is reloaded when the file is re-opened. The tracked free space information **persist**s across HDF5 file sessions, and the free-space manager remains aware of free space sections that became available in any file session.

With this strategy, when space is needed for file metadata or raw data, the HDF5 Library first requests space from the free-space manager. If the request is not satisfied, the library requests space from the aggregators. If the request is still not satisfied, the library requests space from the virtual file driver. That is, the library will use all of the mechanisms for allocating space.

The All Persist strategy offers every possible opportunity for reusing free space. The HDF5 file will contain extra file metadata for the tracked free space. The library will perform additional "accounting" operations to track free space and to search the free space sections when allocating space for file metadata and raw data.

This strategy was added to the library in HDF5 release 1.10.0.

3.2. The All Strategy

The aim of the All strategy is to maximize the use of space within an HDF5 file during a single session.

With this strategy, the HDF5 Library's free-space manager tracks the free space that results from manipulating HDF5 objects in an HDF5 file. The free-space manager is aware of free space sections that became available in the current file session, but the tracked free space information is not saved when the HDF5 file is closed. Free space that exists when the file is closed becomes unaccounted space in the HDF5 file. Unallocated space in the aggregators' blocks may also become unaccounted space when the session ends.

As with the All Persist strategy, the library will try all of the mechanisms for allocating space. When space is needed for file metadata or raw data, the HDF5 Library first requests space from the free-space manager. If the request is not satisfied, the library requests space from the aggregators. If the request is still not satisfied, the library requests space from the virtual file driver.

The All strategy allows free space incurred in the current session to be reused in the current session. Since the free space is not saved after the file is closed, no extra file metadata is needed to keep track of the free space.

The All strategy is the HDF5 Library's default file space management strategy. Prior to HDF5 Release 1.10.0, the All strategy was the only file space management strategy directly supported by the library.

3.3. The Aggregator VFD Strategy

The aim of the Aggregator VFD strategy is to deliver better access performance in situations where many small datasets might be written to a file. Space requests to hold small datasets would go to the aggregators. Not having to find an appropriate size among the tracked free space will improve access performance.

With the Aggregator VFD strategy, the library does not track the free space that results from manipulating HDF5 objects in an HDF5 file. All free space immediately becomes unaccounted space. Unallocated bytes in the aggregators' blocks when the file is closed will also become unaccounted space.

With this strategy, when space is needed for file metadata or raw data, the HDF5 Library first requests space from the aggregators. If the request is not satisfied, the library requests space from the virtual file driver. That is, the library will use the aggregators and VFD mechanisms to fill requests for file space.

Because there are different aggregators for file metadata and raw data, this strategy tends to co-locate file metadata more than some other strategies that can reuse free space scattered throughout the file.

This strategy was added to the library in HDF5 release 1.10.0.

3.4. The VFD Strategy

The aim of the VFD strategy is to maximize file writing performance. Because allocation requests go directly to the virtual file driver, this strategy is best suited for HDF5 files whose primary file usage pattern consists of writing large amounts of raw data to extend dataset objects.

With the VFD strategy, the HDF5 Library does not track the free space that results from the manipulation of HDF5 objects in an HDF5 file. All free space immediately becomes unaccounted space.

With this strategy, when space is needed for file metadata or raw data, the HDF5 Library requests space from the VFD.

This strategy was added to the library in HDF5 release 1.10.0.

4. Setting or Changing a File Space Management Strategy

The purpose of this chapter is to describe how to set or change the file space management strategy for a file. The file space management strategy that a file operates under is set when the file is created. The strategy can be changed when the h5repack command line utility program is run.

4.1. Specifying a Strategy at File Creation with H5Pset_file_space

To set the strategy when an HDF5 file is created, use the `H5Pset_file_space` routine.

The signature for the routine is:

```
herr_t H5Pset_file_space(hid_t fcpl_id, H5F_file_space_t strategy, hsize_t
threshold)
```

The first parameter, `fcpl_id`, is the file creation property list identifier that will be used when the HDF5 file is created. The second parameter, `strategy`, is one of the four strategies described above. Valid values for this parameter are also shown in Table 2 on page 12. The third parameter, `threshold`, is the free-space section threshold used by the library's free-space manager. The threshold is the minimum size in bytes of the free-space sections that are tracked.

The library provides the companion routine H5Pget_file_space. This routine retrieves the file space management information for an HDF5 file. See the entry in the *HDF5 Reference Manual*.

The following code sample shows how to create an empty HDF5 file, *persist.h5,* with the All Persist file space management strategy:

```
/* Create a file creation property list template */
fcpl_id = H5Pcreate(H5P_FILE_CREATE);

/* Set the file space management strategy */
/* Don't update the free-space section threshold */
H5Pset_file_space(fcpl_id, H5P_FILE_SPACE_ALL_PERSIST, (hsize_t)0);

/* Create an HDF5 file with the file creation property list fcpl_id */
fid = H5Fcreate("persist.h5", H5F_ACC_TRUNC, fcpl_id, H5P_DEFAULT);

/* Close the file */
H5Fclose(fid);
```

Example 1. Use H5Pset_file_space to set the file space management strategy

To see what a file's strategy is, use the h5dump command line utility program.

For more information on creating a file, see the entry in the *HDF5 Reference Manual* for `H5Fcreate` and "The HDF5 File" chapter in the *HDF5 User's Guide*.

4.2. Changing the Strategy with h5repack

It is not always possible to know in advance how a file will be used. Running the `h5stat -S` command line utility may show that a given file has a large amount of unaccounted space. This would indicate that the file space management strategy might need to be changed. The h5repack command line utility program can be used to reclaim the unaccounted space, and it can also be used to change the strategy.

The `-S` or `-fs_strategy` options can be used with h5repack to specify the strategy to be used with the output file. The example below shows h5repack specifying the VFD strategy. The input file name is no_persist_A.h5, and the output file name is no_persist_outvfd.h5.

```
h5repack -S SPACE_VFD no_persist_A.h5 no_persist_outvfd.h5
```

Example 2. Using h5repack to change the file space management strategy

The valid values for the `-S` and `-fs_strategy` options are listed in the table below.

Table 2. Strategies and Values

Strategy	h5repack Value	H5Pset_file_space Value
All Persist	ALL_PERSIST	H5F_FILE_SPACE_ALL_PERSIST
All	ALL	H5F_FILE_SPACE_ALL
Aggregator VFD	AGGR_VFD	H5F_FILE_SPACE_AGGR_VFD
VFD	SPACE_VFD	H5F_FILE_SPACE_VFD

4.3. Summary of Strategies and Implementation

The file space management strategies and file allocation mechanisms that have been discussed above are summarized in the table below.

Table 3. Summary of file space management strategies and mechanisms

Strategy	H5Pset_file_space Value h5repack Value	Track Free Space		Allocate Space Using		
		Across Multiple Sessions	Within a Single Session	Free-space Manager	Aggregators	VFD
All Persist	H5F_FILE_SPACE_ALL_PERSIST ALL_PERSIST	Y	Y	Y	Y	Y
All	H5F_FILE_SPACE_ALL ALL	N	Y	Y	Y	Y
Aggregator VFD	H5F_FILE_SPACE_AGGR_VFD AGGR_VFD	N	N	N	Y	Y
VFD	H5F_FILE_SPACE_VFD SPACE_VFD	N	N	N	N	Y

5. Example File Space Management Scenarios

The scenarios in this chapter describe a variety of situations and show how different file space management strategies produce different results.

A session for an HDF5 file starts when the file is opened and ends when it closes. Several scenarios have multiple sessions.

The h5dump and h5stat command line utility programs are used to summarize the changes made in each scenario. The h5dump utility displays the contents of a given HDF5 file. If the –H option is used with h5dump, only the header will be displayed. The h5stat program reports information on the file space for a given HDF5 file. If the –S option is used with h5stat, only summary information will be shown.

5.1. Empty Files

The size and structure of an empty file depends on the file space management strategy that was selected for the file when it was created. The All Persist strategy requires more space since it will be tracking free space over multiple sessions. The All strategy will not require as much space as a file using the All Persist strategy since the file will not be tracking free space as long. The Aggregator VFD and VFD strategies do not track free space, and files created with these strategies are the smallest. Files were created with each strategy. Each file was examined with h5dump and h5stat. The results are shown below.

5.1.1. Create an Empty File with the All Strategy

If an empty file is created with the All file space management strategy, some file space is set aside for the file metadata where free space might be tracked. See the output below:

The command line `h5dump no_persist.h5` produces the following output:

```
HDF5 "no_persist.h5" {
GROUP "/" {
}
}
```

This reveals that the HDF5 Library automatically created the root group and allocated space for initial file metadata when *no_persist.h5* was created. This empty HDF5 file does not yet contain any user-created HDF5 objects.

The command line `h5stat –S no_persist.h5` produces the following output:

```
Filename: no_persist.h5
File space management strategy: H5F_FILE_SPACE_ALL
```

```
Summary of file space information:
  File metadata: 800 bytes
  Raw data: 0 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 0 bytes
Total space: 800 bytes
```

Note that this file contains 800 bytes of file metadata and nothing else; there is no user data and no free space in the file. The file size of the empty HDF5 file *no_persist.h5* equals the size of the file metadata.

5.1.2. Create an Empty File with the All Persist Strategy

If an empty file is created with the All Persist file space management strategy, more file space is set aside for file metadata than is set aside in a file created with the All strategy. More file space is required so that free space can be tracked across sessions. See the output below:

The command line `h5dump persist.h5` produces the following output:

```
HDF5 "persist.h5" {
GROUP "/" {
}
}
```

This shows that the HDF5 Library automatically created the root group and allocated space for initial file metadata when *no_persist.h5* was created. This empty HDF5 file does not yet contain any user-created HDF5 objects.

The command line `h5stat -S persist.h5` produces the following output:

```
Filename: persist.h5
File space management strategy: H5F_FILE_SPACE_ALL_PERSIST
Summary of file space information:
  File metadata: 840 bytes
  Raw data: 0 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 0 bytes
Total space: 840 bytes
```

Note that this file contains 840 bytes of file metadata.

5.1.3. Create an Empty File with the Aggregator VFD Strategy

If an empty file is created with the Aggregator VFD file space management strategy, less file space is set aside for file metadata than is set aside in a file created with the All or All Persist strategies. Less file space is required since free space is not tracked. See the output below:

The command line `h5dump aggrvfd.h5` produces the following output:

```
HDF5 "aggrvfd.h5" {
GROUP "/" {
}
}
```

The command line `h5stat -S aggrvfd.h5` produces the following output:

```
Filename: aggrvfd.h5
File space management strategy: H5F_FILE_SPACE_AGGR_VFD
Summary of file space information:
  File metadata: 792 bytes
  Raw data: 0 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 0 bytes
Total space: 792 bytes
```

Note that this file contains 792 bytes of file metadata. This amount is less than the amounts of metadata set aside in files created with the All Persist (840) and All (800) strategies.

5.1.4. Create an Empty File with the VFD Strategy

If an empty file is created with the VFD file space management strategy, less file space is set aside for file metadata than is set aside in a file created with the All or All Persist strategies. Less file space is required since free space is not tracked. See the output below:

The command line `h5dump vfd.h5` produces the following output:

```
HDF5 "vfd.h5" {
GROUP "/" {
}
}
```

The command line `h5stat -S vfd.h5` produces the following output:

```
Filename: vfd.h5
File space management strategy: H5F_FILE_SPACE_VFD
Summary of file space information:
  File metadata: 792 bytes
  Raw data: 0 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 0 bytes
Total space: 792 bytes
```

Note that this file contains 792 bytes of file metadata. This amount is less than the amounts of metadata set aside in files created with the All Persist (840) and All (800) strategies.

5.2. Scenario A: The All Strategy, Multiple Sessions

The default file space management strategy is the All strategy. The All strategy tracks free space that becomes available during the current session. See page 9 for more information. This scenario uses four sessions to show how file size changes as datasets are added and deleted.

5.2.1. Session 1: Create a File and Add Datasets

In the first session, a user creates an HDF5 file named *no_persist_A.h5,* adds four datasets (*dset1*, *dset2*, *dset3, and dset4*) of different sizes, and closes the file. No file space management strategy is specified, so the file is created with the All strategy.

The command line `h5dump -H no_persist_A.h5` produces the following output:

```
HDF5 "no_persist_A.h5" {
GROUP "/" {
   DATASET "dset1" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 10 ) / ( 10 ) }
   }
   DATASET "dset2" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 30000 ) / ( 30000 ) }
   }
   DATASET "dset3" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 50 ) / ( 50 ) }
   }
   DATASET "dset4" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 100 ) / ( 100 ) }
   }
}
}
```

The command line `h5stat –S no_persist_A.h5` produces the following output:

```
Filename: no_persist_A.h5
File space management strategy: H5F_FILE_SPACE_ALL
Summary of file space information:
  File metadata: 2216 bytes
  Raw data: 120640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 1976 bytes
Total space: 124832 bytes
```

The data values in the four new dataset objects occupy the 120640 bytes of raw data space. The amount of tracked free space in the file is 0 bytes, while there are 1976 bytes of unaccounted space. The unaccounted space is due to the file space management strategy in use for the *no_persist_A.h5* HDF5 file.

The HDF5 Library's default file space management strategy does not retain tracked free space information across multiple sessions with an HDF5 file. This means the information about free space that is collected by the library during the current session (since the file was opened) is not saved when the file is closed. With the default strategy, free space that is incurred during a particular session can be reused during that session, but is unavailable for reuse in all future sessions. This unavailable file free space is reported as "unaccounted space" in the `h5stat -S` output.

As demonstrated in this example, file free space can be created not only when HDF5 objects are deleted from a file, but also when they are added. This is because adding an object may introduce gaps in the file as new space is allocated for file metadata and HDF5 dataset values. HDF5 files that might develop large amounts of unaccounted space are candidates for non-default file space management strategies if file size is a concern.

5.2.2. Session 2: Delete One Dataset

In session 2 with *no_persist_A.h5,* a user opens the file, deletes an existing dataset (*dset2*), and closes the file.

The command line `h5dump -H no_persist_A.h5` produces the following output:

```
HDF5 "no_persist_A.h5" {
GROUP "/" {
   DATASET "dset1" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 10 ) / ( 10 ) }
   }
   DATASET "dset3" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 50 ) / ( 50 ) }
   }
   DATASET "dset4" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 100 ) / ( 100 ) }
   }
}
}
```

The command line `h5stat -S no_persist_A.h5` produces the following output:

```
Filename: no_persist_A.h5
File space management strategy: H5F_FILE_SPACE_ALL
Summary of file space information:
  File metadata: 1944 bytes
  Raw data: 640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 122248 bytes
Total space: 124832 bytes
```

At this point, the amount of unaccounted space consists of the 1976 bytes that were there from Session 1, and the additional free space incurred due to the deletion of *dset2*. Note that the file space is now almost 98% unaccounted space and the 120000 bytes of space that originally stored the data values for *dset2* make up a substantial fraction of that.

5.2.3. Session 3: Add One Dataset

In session 3 with *no_persist_A.h5,* a user opens the file, adds a new dataset (*dset5*), and then closes it.

The command line `h5dump -H no_persist_A.h5` produces the following output:

```
HDF5 "no_persist_A.h5" {
GROUP "/" {
   DATASET "dset1" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 10 ) / ( 10 ) }
   }
   DATASET "dset3" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 50 ) / ( 50 ) }
   }
   DATASET "dset4" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 100 ) / ( 100 ) }
   }
   DATASET "dset5" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 1000 ) / ( 1000 ) }
   }
}
}
```

The command line `h5stat -S no_persist_A.h5` produces the following output:

```
Filename: no_persist_A.h5
File space management strategy: H5F_FILE_SPACE_ALL
Summary of file space information:
  File metadata: 2216 bytes
  Raw data: 4640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 124024 bytes
Total space: 130880 bytes
```

Note that the amount of unaccounted space increases as well as the file size. This is due to the strategy of not reusing free space when the file is closed.

5.2.4. Session 4: Add another Dataset

In session 4 with *no_persist_A.h5,* a user re-opens the file, adds a new dataset (*dset6*) with size slightly smaller than *dset2*, and then closes the file.

The command line `h5dump -H no_persist_A.h5` produces the following output:

```
HDF5 "no_persist_A.h5" {
GROUP "/" {
   DATASET "dset1" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 10 ) / ( 10 ) }
   }
   DATASET "dset3" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 50 ) / ( 50 ) }
   }
   DATASET "dset4" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 100 ) / ( 100 ) }
   }
   DATASET "dset5" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 1000 ) / ( 1000 ) }
   }
   DATASET "dset6" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 28000 ) / ( 28000 ) }
   }
}
}
```

The command line `h5stat -S no_persist_A.h5` produces the following output:

```
Filename: no_persist_A.h5
File space management strategy: H5F_FILE_SPACE_ALL
Summary of file space information:
  File metadata: 2488 bytes
  Raw data: 116640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 125800 bytes
Total space: 244928 bytes
```

In this latest session, the amount of unaccounted space continues to increase. The file size has almost doubled at this point and half of it is unaccounted space. The HDF5 file *no_persist_A.h5* now contains fragments of lost space resulting from the manipulation of the HDF5 objects in the file and the use of the default file space management strategy. HDF5 files that might develop large amounts of unaccounted space are candidates for non-default file space management strategies if file size is a concern.

Compare the size of this file with the size of the file in session 4 of Scenario B below. See page 22.

5.3. Scenario B: The All Persist Strategy, Multiple Sessions

The All Persist strategy is an alternative to the All strategy. The All Persist strategy tracks free space over multiple sessions. For more information on the All Persist strategy, see page 9. This scenario uses four sessions to show how file size changes as datasets are added and deleted. The sequence of adding and deleting scenarios over multiple sessions is the same as the sequence used in Scenario A.

5.3.1. Session 1: Create a File and Add Datasets

In this session, a user creates an HDF5 file called *persist_B.h5* using the All Persist strategy, adds the same four datasets (*dset1*, *dset2*, *dset3, and dset4*) that were added to *no_persist_A.h5* in Scenario A, Session 1, and closes the file.

The command line `h5dump -H persist_B.h5` produces the same output as the command line `h5dump -H no_persist_A.h5` produced in Session 1 of Scenario A. See page 16.

The command line `h5stat -S persist_B.h5` produces the following output:

```
Filename: persist_B.h5
File space management strategy: H5F_FILE_SPACE_ALL_PERSIST
Summary of file space information:
  File metadata: 2391 bytes
  Raw data: 120640 bytes
  Amount/Percent of tracked free space: 1854 bytes/1.5%
  Unaccounted space: 0 bytes
Total space: 124885 bytes
```

In contrast to *no_persist_A.h5* after Session 1, *persist_B.h5* contains no unaccounted space. It does, however, contain 1854 bytes of tracked free space. The amount of file metadata in *persist_B.h5* (2391 bytes) is slightly larger than what was in *no_persist_A.h5* (2216 bytes). This increase is due to the extra metadata used by the library to save the tracked free space information.

The `h5stat -s` command (lower case S) shows more detail about the distribution of tracked free space in *persist_B.h5*:

```
Filename: persist_B.h5
Free-space section threshold: 1 bytes
Small size free-space sections (< 10 bytes):
        Total # of small size sections: 0
Free-space section bins:
        # of sections of size 10 - 99: 1
        # of sections of size 1000 - 9999: 1
        Total # of sections: 2
```

There are two free-space sections in *persist_B.h5*; one section contains between 10 and 99 bytes, and the second contains between 1000 and 9999 bytes.

5.3.2. Session 2: Delete One Dataset

In this session, a user reopens *persist_B.h5*, deletes *dset2,* and closes the file.

The command line `h5dump -H persist_B.h5` produces the same output as the command line `h5dump -H no_persist_A.h5` produced in Session 2 of Scenario A. See page 17.

After the file has been closed, `h5stat -S` produces the following output:

```
Filename: ./persist_B.h5
File space management strategy: H5F_FILE_SPACE_ALL_PERSIST
Summary of file space information:
  File metadata: 2155 bytes
  Raw data: 640 bytes
  Amount/Percent of tracked free space: 122126 bytes/97.8%
  Unaccounted space: 0 bytes
Total space: 124921 bytes
```

The amount of tracked free space after the deletion of *dset2* reflects the 1854 bytes of tracked free space that was previously in the file and the free space adjustments resulting from the changes in Session 2.

In this scenario, when *dset2* was deleted, the bytes that were used for that dataset's raw data and file metadata were added to the file's tracked free space by the HDF5 Library. The tracked free space information was saved (persisted) when the file was closed.

The command line `h5stat -s` shows the distribution of free space in *persist_B.h5* at the end of Session 2:

```
Filename: ./persist_B.h5
Free-space section threshold: 1 bytes
Small size free-space sections (< 10 bytes):
        Total # of small size sections: 0
Free-space section bins:
        # of sections of size 10 - 99: 1
        # of sections of size 100 - 999: 1
        # of sections of size 1000 - 9999: 1
        # of sections of size 100000 - 999999: 1
        Total # of sections: 4
```

Note that *persist_B.h5* now has two additional free-space sections resulting from the manipulation of the HDF5 objects in the file during Session 2.

5.3.3. Session 3: Add One Dataset

In this session with *persist_B.h5*, a user opens the file, adds a new dataset (*dset5*), and closes the file.

The command line `h5dump -H persist_B.h5` produces the same output as the command line `h5dump -H no_persist_A.h5` produced in Session 3 of Scenario A. See page 18.

The command line `h5stat -S persist_B.h5` produces the following output:

```
Filename: persist_B.h5
File space management strategy: H5F_FILE_SPACE_ALL_PERSIST
Summary of file space information:
  File metadata: 2409 bytes
  Raw data: 4640 bytes
  Amount/Percent of tracked free space: 117854 bytes/94.4%
  Unaccounted space: 0 bytes
Total space: 124903 bytes
```

The command line `h5stat -s` shows the distribution of free space in *persist_B.h5* at the end of Session 3 and produces the following output:

```
Filename: persist_B.h5
Free-space section threshold: 1 bytes
Small size free-space sections (< 10 bytes):
        Total # of small size sections: 0
Free-space section bins:
        # of sections of size 10 - 99: 1
        # of sections of size 1000 - 9999: 1
        # of sections of size 100000 - 999999: 1
        Total # of sections: 3
```

Note that the amount of tracked free space has decreased. The library is able to reuse the free space tracked by the free-space manager.

5.3.4. Session 4: Add another Dataset

In this session with *persist_B.h5,* a user re-opens the file, adds a new dataset (*dset6*) with size slightly smaller than the deleted *dset2*, and then closes the file.

The command line `h5dump -H persist_B.h5` produces the same output as the command line `h5dump -H no_persist_A.h5` produced in Session 4 of Scenario A. See page 19.

The command line `h5stat -S persist_B.h5` produces the following output:

```
Filename: persist_B.h5
File space management strategy: H5F_FILE_SPACE_ALL_PERSIST
Summary of file space information:
  File metadata: 2681 bytes
  Raw data: 116640 bytes
  Amount/Percent of tracked free space: 5582 bytes/4.5%
  Unaccounted space: 0 bytes
Total space: 124903 bytes
```

The command line `h5stat -s` shows the distribution of free space in *persist_B.h5* at the end of Session 4 and produces the following output:

```
Filename: persist_B.h5
Free-space section threshold: 1 bytes
Small size free-space sections (< 10 bytes):
        Total # of small size sections: 0
```

```
Free-space section bins:
        # of sections of size 10 - 99: 1
        # of sections of size 1000 - 9999: 2
        Total # of sections: 3
```

Note that the amount of tracked free space decreases a lot. The library is able to reuse the free space tracked by the free-space manager for the addition of *dset6*. The file size is 50% less than the file size of *no_persist_A.h5* in Scenario A at the end of Session 4 (page 19).

5.4. Scenarios C – F, Single Sessions

In Scenarios C to F, datasets are added and deleted in the same sequence as in Sessions 1 – 3 in Scenarios A and B. Besides using only the actions from Sessions 1 – 3, the difference is all the actions in the scenarios below occur in one session.

5.4.1. Scenario C: Create File, Manipulate Objects, All Strategy

In the only session of this scenario, a user creates an HDF5 file named *no_persist_C.h5* using the All strategy. The user then adds four datasets (*dset1, dset2, dset3,* and *dset4*), deletes *dset2*, and adds *dset5* before closing the file. The file management strategy is the same strategy that was used in Scenario A. The HDF5 objects are manipulated in the same order as they were in Sessions 1-3 of Scenario A. See page 16.

The command line `h5stat –S no_persist_C.h5` produces the following output:

```
Filename: no_persist_C.h5
File space management strategy: H5F_FILE_SPACE_ALL
Summary of file space information:
  File metadata: 2216 bytes
  Raw data: 4640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 117976 bytes
Total space: 124832 bytes
```

The file size for *no_persist_C.h5* is about 6000 bytes smaller than the file size for *no_persist_A.h5* after Session 3 of Scenario A. The reason for this is the HDF5 Library was able to reuse some of the free space it was tracking when all of the object manipulations took place in a single session. *no_persist_C.h5* still has a substantial amount of unaccounted space (117976 bytes) – almost 95% of the total file space.

5.4.2. Scenario D: Create File, Manipulate Objects, All Persist Strategy

In the only session of this scenario, a user creates an HDF5 file named *persist_D.h5* using the All Persist strategy. The user then adds four datasets (*dset1, dset2, dset3,* and *dset4*), deletes *dset2*, and adds *dset5* before closing the file. The file management strategy is the same strategy that was used in Scenario B.

The HDF Group

The HDF5 objects are manipulated in the same order as they were in Sessions 1-3 of Scenario B. See page 20.

The command line `h5stat -S persist_D.h5` produces the following output:

```
Filename: persist_D.h5
File space management strategy: H5F_FILE_SPACE_ALL_PERSIST
Summary of file space information:
  File metadata: 2409 bytes
  Raw data: 4640 bytes
  Amount/Percent of tracked free space: 117854 bytes/94.4%
  Unaccounted space: 0 bytes
Total space: 124903 bytes
```

The file size for *persist_D.h5* is the same as the file size for *persist_B.h5* after Session 3 of Scenario B. See page 21.

The file space information for *persist_D.h5* (Scenario D) and *no_persist_C.h5* (Scenario C) are very similar. The file size of *no_persist_C.h5* is a bit smaller. For both files, the library's free-space manager tracks and reuses the free space when manipulating the objects in the session. Looking at the size of the file metadata for the two files, the greater amount of file metadata in *persist_D.h5* is due to the extra metadata needed to keep free space information persistent when the file is closed. This demonstrates that using the All strategy, as was done for *no_persist_C.h5,* has some saving in file space compared to using the All Persist strategy when the HDF5 object manipulation occurs in a single session. The exact amount of space savings will depend on the number and size of the HDF5 objects that are added and deleted and the value of the free-space section threshold.

5.4.3. Scenario E: Create File, Manipulate Objects, Aggregator VFD Strategy

In the only session of this scenario, a user creates an HDF5 file named aggrvfd_E.*h5* using the Aggregator VFD strategy. For more information on this strategy, see page 10. The user then adds four datasets (*dset1, dset2, dset3,* and *dset4*), deletes *dset2*, and adds *dset5* before closing the file. The HDF5 objects are manipulated in the same order as they were in Sessions 1-3 of Scenarios A and B.

The command line `h5stat -S aggrvfd_E.h5` produces the following output:

```
Filename: aggrvfd_E.h5
File space management strategy: H5F_FILE_SPACE_AGGR_VFD
Summary of file space information:
  File metadata: 2208 bytes
  Raw data: 4640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 121936 bytes
Total space: 128784 bytes
```

See the "Comparing Scenarios A - F" section below for more information.

5.4.4. Scenario F: Create File, Manipulate Objects, VFD Strategy

In the only session of this scenario, a user creates an HDF5 file named vfd_F.*h5* using the VFD strategy. For more information on this strategy, see page 10. The user then adds four datasets (*dset1, dset2, dset3,* and *dset4*), deletes *dset2*, and adds *dset5* before closing the file. The HDF5 objects are manipulated in the same order as they were in Sessions 1-3 of Scenarios A and B.

The command line `h5stat -S vfd_F.h5` produces the following output:

```
Filename: vfd_F.h5
File space management strategy: H5F_FILE_SPACE_VFD
Summary of file space information:
  File metadata: 2208 bytes
  Raw data: 4640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 120272 bytes
Total space: 127120 bytes
```

See the "Comparing Scenarios A - F" below for more information.

5.4.5. Comparing Scenarios A - F

The table below compares the various scenarios that have been described so far in this document. The values for the size columns are some number of bytes.

Table 4. Comparing Scenarios A – F

Scenario / # Sessions	Strategy	File Name	File Size	File Metadata	Raw Data	Tracked Free Space	Unaccounted Space
A / 3	All	no_persist_A.h5	130880	2216	4640	0	124024
B / 3	All Persist	persist_B.h5	124903	2409	4640	117854	0
C / 1	All	no_persist_C.h5	124832	2216	4640	0	117976
D / 1	All Persist	persist_D.h5	124903	2409	4640	117854	0
E / 1	Aggregator VFD	aggrvfd_E.h5	128784	2208	4640	0	121936
F / 1	VFD	vfd_F.h5	127120	2208	4640	0	120272

Here are some observations:

- The file from Scenario A was written over three sessions and has the largest file size.
- The files from Scenarios B and D are a bit larger than the file in Scenario C, but the unused space in Scenarios B and D is tracked free space and may be reused in later sessions if more HDF5 objects are added to the file.
- The files from Scenarios E and F are larger than the files from Scenarios C and D. These four files were created in a single session. The reason the files in E and F are larger is the Aggregator VFD and VFD strategies do not track free space and do not reuse any space that is released as HDF5 objects are manipulated. The file in Scenario E is larger than the file in Scenario F because bytes in the aggregators' blocks have become unaccounted in the process of managing space. See the "Scenarios K – N" section on page 29 to see the effect of a different usage pattern on files using the Aggregator VFD and VFD strategies.

5.5. Scenarios G – J, Single Sessions

Scenarios G to J are the same as Scenarios C to F except that *dset6* is added in scenarios G to J and was not added in scenarios C to F. So, Scenarios G to J show what happens in a single session when a dataset that is slightly smaller than a previously deleted dataset is added to the file. The results in Scenarios G to J could also be compared to the results of Sessions 1 to 4 in Scenarios A and B. The actions are the same in Scenarios A, B, and G to J, but the number of sessions is different.

5.5.1. Scenario G: Create File, Manipulate Objects, All Strategy

In the only session of this scenario, a user creates an HDF5 file named *no_persist_G.h5* using the All strategy. The user then adds four datasets (*dset1, dset2, dset3,* and *dset4*), deletes *dset2*, and adds *dset5* and *dset6* before closing the file. The file management strategy is the same strategy that was used in Scenario A. The HDF5 objects are manipulated in the same order as they were in Sessions 1-4 of Scenario A. See page 16.

The command line `h5stat -S no_persist_G.h5` produces the following output:

```
Filename: no_persist_G.h5
File space management strategy: H5F_FILE_SPACE_ALL
Summary of file space information:
  File metadata: 2488 bytes
  Raw data: 116640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 5704 bytes
Total space: 124832 bytes
```

The file size for *no_persist_G.h5* is about half the file size for *no_persist_A.h5* after Session 4 of Scenario A. The reason for this is the HDF5 Library was able to reuse some of the free space it was tracking when all of the object manipulations took place in a single session.

5.5.2. Scenario H: Create File, Manipulate Objects, All Persist Strategy

In the only session of this scenario, a user creates an HDF5 file named *persist_H.h5* using the All Persist strategy. The user then adds four datasets (*dset1, dset2, dset3,* and *dset4*), deletes *dset2*, and adds *dset5* and *dset6* before closing the file. The file management strategy is the same strategy that was used in Scenario B. The HDF5 objects are manipulated in the same order as they were in Sessions 1-4 of Scenario B. See page 20.

The command line `h5stat -S persist_H.h5` produces the following output:

```
Filename: persist_H.h5
File space management strategy: H5F_FILE_SPACE_ALL_PERSIST
Summary of file space information:
  File metadata: 2681 bytes
  Raw data: 116640 bytes
  Amount/Percent of tracked free space: 5582 bytes/4.5%
```

```
    Unaccounted space: 0 bytes
  Total space: 124903 bytes
```

As you might expect, the output from h5stat for the file in this scenario is the same as the output for the file in Scenario B. Since free space is tracked over sessions in Scenario B, changing the number of sessions does not produce any improvements. For more information on Session 4 of Scenario B, see page 22.

5.5.3. Scenario I: Create File, Manipulate Objects, Aggregator VFD Strategy

In the only session of this scenario, a user creates an HDF5 file named *aggrvfd_I.h5* using the Aggregator VFD strategy. The user then adds four datasets (*dset1, dset2, dset3,* and *dset4*), deletes *dset2*, and adds *dset5* and *dset6* before closing the file. The HDF5 objects are manipulated in the same order as they were in Sessions 1-4 of Scenarios A and B.

The command line `h5stat –S aggrvfd_I.h5` produces the following output:

```
Filename: aggrvfd_I.h5
File space management strategy: H5F_FILE_SPACE_AGGR_VFD
Summary of file space information:
  File metadata: 2480 bytes
  Raw data: 116640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 121664 bytes
Total space: 240784 bytes
```

Not tracking free space at all yields a file size close to the file size in Scenario A Session 4 where file space was not tracked over more than one session. For more information on Scenario A Session 4, see page 19.

The difference between this scenario and Scenario E (page 24) is the addition of another dataset. The raw data is larger because of the dataset. The file metadata is slightly larger, and the unaccounted space is about the same.

5.5.4. Scenario J: Create File, Manipulate Objects, VFD Strategy

In the only session of this scenario, a user creates an HDF5 file named *vfd_J.h5* using the VFD strategy. The user then adds four datasets (*dset1, dset2, dset3,* and *dset4*), deletes *dset2*, and adds *dset5* and *dset6* before closing the file. The HDF5 objects are manipulated in the same order as they were in Sessions 1-4 of Scenarios A and B.

The command line `h5stat –S vfd_J.h5` produces the following output:

```
Filename: vfd_J.h5
File space management strategy: H5F_FILE_SPACE_VFD
Summary of file space information:
  File metadata: 2480 bytes
  Raw data: 116640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
```

```
    Unaccounted space: 120272 bytes
Total space: 239392 bytes
```

The h5stat output is similar to Scenario F (page 25) except for the additional dataset. The file size is slightly smaller than Scenario I (page 27).

5.5.5. Comparing Scenarios A, B, G - J

The table below compares Scenarios A, B, and G to J. The values for the size columns are some number of bytes.

Table 4. Comparing Scenarios A, B, G - J

Scenario / # Sessions	Strategy	File Name	File Size	File Metadata	Raw Data	Tracked Free Space	Unaccounted Space
A / 4	All	no_persist_A.h5	244928	2488	116640	0	125800
B / 4	All Persist	persist_B.h5	124903	2681	116640	5582	0
G / 1	All	no_persist_G.h5	124832	2488	116640	0	5704
H / 1	All Persist	persist_H.h5	124903	2681	116640	5582	0
I / 1	Aggregator VFD	aggrvfd_I.h5	240784	2480	116640	0	121664
J / 1	VFD	vfd_J.h5	239392	2480	116640	0	120272

Here are some observations:

- The file from Scenario A was written over four sessions and has the largest file size.
- The files from Scenarios B and H are a bit larger than the file in Scenario G, but the unused space in Scenarios B and H is tracked free space and may be reused in later sessions if more HDF5 objects are added to the file.
- The files from Scenarios I and J are larger than the files from Scenarios G and H. These four files were created in a single session. The reason the files in I and J are larger is the Aggregator VFD and VFD strategies do not track free space and do not reuse any space that is released as HDF5 objects are manipulated. The file in Scenario I is larger than the file in Scenario J because bytes in the aggregators' blocks have become unaccounted in the process of managing space. See the "Scenarios K – N" section on page 29 to see the effect of a different usage pattern on files using the Aggregator VFD and VFD strategies.

Other Aggregator VFD and VFD Strategy Notes

- Since files using the VFD strategy do not have any space tracking or special handling features, there is no difference between a scenario where the actions take place in a single session and a scenario where the actions take place over multiple sessions.
- Suppose we compare two files that both use the Aggregator VFD strategy, and suppose that the same actions are done to both files. The file where the actions are done over multiple sessions will grow slightly compared to the file where the actions are done in a single session. The reason for the growth in the first file is that file space is allocated for the aggregators each time the file is opened. When the file is closed, any of the unused allocated space becomes unallocated space.

5.6. Scenarios K – N, no Objects Deleted

In Scenarios K - N, datasets are added to each file and no objects are deleted. Scenario K uses the All strategy, Scenario L uses the All Persist strategy, Scenario M uses the Aggregator VFD strategy, and Scenario N uses the VFD strategy. The activity in each scenario occurs in one session. See the "Comparing Scenarios K to N" on page 31 for some observations.

5.6.1. Scenario K: Create File, Add Objects, All Strategy

In the only session of this scenario, a user creates an HDF5 file named *no_persist_K.h5* using the All strategy (see page 10). The user then adds six datasets (*dset1, dset2, dset3, dset4, dset5, and dset6*) and closes the file.

The command line `h5stat -S no_persist_K.h5` produces the following output:

```
Filename: no_persist_K.h5
File space management strategy: H5F_FILE_SPACE_ALL
Summary of file space information:
  File metadata: 2488 bytes
  Raw data: 232640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 1704 bytes
Total space: 236832 bytes
```

5.6.2. Scenario L: Create File, Add Objects, All Persist Strategy

In the only session of this scenario, a user creates an HDF5 file named *persist_L.h5* using the All Persist strategy (see page 9). The user then adds six datasets (*dset1, dset2, dset3, dset4, dset5, and dset6*) and closes the file.

The command line `h5stat -S persist_L.h5` produces the following output:

```
Filename: persist_L.h5
File space management strategy: H5F_FILE_SPACE_ALL_PERSIST
Summary of file space information:
  File metadata: 2663 bytes
  Raw data: 232640 bytes
  Amount/Percent of tracked free space: 1582 bytes/0.7%
  Unaccounted space: 0 bytes
Total space: 236885 bytes
```

5.6.3. Scenario M: Create File, Add Objects, Aggregator VFD Strategy

In the only session of this scenario, a user creates an HDF5 file named *aggrvfd_M.h5* using the Aggregator VFD strategy (see page 10). The user then adds six datasets (*dset1, dset2, dset3, dset4, dset5, and dset6*) and closes the file.

The command line `h5stat -S aggrvfd_M.h5` produces the following output:

```
Filename: aggrvfd_M.h5
File space management strategy: H5F_FILE_SPACE_AGGR_VFD
Summary of file space information:
  File metadata: 2480 bytes
  Raw data: 232640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 1664 bytes
Total space: 236784 bytes
```

5.6.4. Scenario N: Create File, Add Objects, VFD Strategy

In the only session of this scenario, a user creates an HDF5 file named *vfd_N.h5* using the VFD strategy. The user then adds six datasets (*dset1, dset2, dset3, dset4, dset5, and dset6*) and closes the file.

The command line `h5stat -S vfd_N.h5` produces the following output:

```
Filename: vfd_N.h5
File space management strategy: H5F_FILE_SPACE_VFD
Summary of file space information:
  File metadata: 2480 bytes
  Raw data: 232640 bytes
  Amount/Percent of tracked free space: 0 bytes/0.0%
  Unaccounted space: 0 bytes
Total space: 235120 bytes
```

5.6.5. Comparing Scenarios K to N

The table below compares Scenarios K, L, M, and N. The values for the size columns are some number of bytes.

Table 5. Summary of Scenarios K – N

Scenario / # Sessions	Strategy	File Name	File Size	File Metadata	Raw Data	Tracked Free Space	Unaccounted Space
K /1	All	no_persist_K.h5	236832	2488	232640	0	1704
L / 1	All Persist	persist_L.h5	236885	2663	232640	1582	0
M/1	Aggregator VFD	aggrvfd_M.h5	236784	2480	232640	0	1664
N/ 1	VFD	vfd_N.h5	235120	2480	232640	0	0

The table above shows the file space information for HDF5 files after six datasets have been added.

The files in Scenarios M and N are smaller than the files in Scenarios K and L. The file in Scenario N, managed with the VFD strategy, has the smallest size with no tracked free space or unaccounted space. Even though the file in Scenario M has less saving in file space than the file in Scenario N, it will have the benefit of better I/O performance due to the use of aggregators for servicing space allocation requests. Metadata in Scenario M will also tend to be more concentrated in contiguous blocks than in the file in Scenario N.

See the “Other Aggregator VFD and VFD Strategy Notes” section on page 28 for more information.