

Example of Szip Usage in HDF4

The following is one of several sample programs illustrating the use of Szip compression in HDF4. The entire set can be found in the subdirectory [h4_examples/](#).

```
/* This program writes and reads 16-bit integer data using SZIP compression */

#include <hdf.h>
#include <szlib.h>

#define FILE_NAME16 "SDS_16_sziped.hdf"
#define SDS_NAME "SzipedData"

#define RANK 2
#define WIDTH 6
#define LENGTH 9

int main()
{
    /****** Variable declaration *****/

    int32 sd_id, sds_id;
    intn status;
    int32 dim_sizes[2], array_rank, num_type, attributes;
    char name[MAX_NC_NAME];
    comp_info c_info;
    int32 start[2], edges[2];
    int16 fill_value = 0; /* Fill value */
    int i, j;
    int num_errs = 0; /* number of errors so far */
    int16 out_data[LENGTH][WIDTH];
    int16 in_data[LENGTH][WIDTH] = {
        100,100,200,200,300,400,
        100,100,200,200,300,400,
        100,100,200,200,300,400,
        300,300, 0,400,300,400,
        300,300, 0,400,300,400,
        300,300, 0,400,300,400,
        0, 0,600,600,300,400,
        500,500,600,600,300,400,
        0, 0,600,600,300,400};

    /****** End of variable declaration *****/

    /* Create the file and initialize SD interface */
    sd_id = SDstart (FILE_NAME16, DFACC_CREATE);

    /* Create the SDS */
    dim_sizes[0] = LENGTH;
    dim_sizes[1] = WIDTH;
    sds_id = SDcreate (sd_id, SDS_NAME, DFNT_INT16, RANK, dim_sizes);

    /* Define the location, pattern, and size of the data set */
    for (i = 0; i < RANK; i++) {
        start[i] = 0;
        edges[i] = dim_sizes[i];
    }

    /* Fill the SDS array with the fill value */
    status = SDsetfillvalue (sds_id, (VOIDP)&fill_value);

    /* Initialize parameters for SZIP */
    /* other parameters are automatically set by HDF */
    c_info.szip.pixels_per_block = 2; /* even number from 2-32 */
    c_info.szip.options_mask = SZ_NN_OPTION_MASK; /* or SZ_EC_OPTION_MASK */
    c_info.szip.options_mask |= SZ_RAW_OPTION_MASK;

    /* Set the compression */
    status = SDsetcompress (sds_id, COMP_CODE_SZIP, &c_info);
}
```

```

/* Write data to the SDS */
status = SDwritedata(sds_id, start, NULL, edges, (VOIDP)in_data);

/* Terminate access to the data set */
status = SDendaccess (sds_id);

/* Terminate access to the SD interface and close the file to
flush the compressed info to the file */
status = SDend (sd_id);

/*
* Verify the compressed data
*/

/* Reopen the file and select the first SDS */
sd_id = SDstart (FILE_NAME16, DFACC_READ);

sds_id = SDselect (sd_id, 0);

/* Wipe out the output buffer */
memset(&out_data, 0, sizeof(out_data));

/* Read the data set */
start[0] = 0;
start[1] = 0;
edges[0] = LENGTH;
edges[1] = WIDTH;
status = SDreaddata (sds_id, start, NULL, edges, (VOIDP)out_data);

/* Compare read data against input data */
for (j = 0; j < LENGTH; j++)
{
    for (i = 0; i < WIDTH; i++)
        if (out_data[j][i] != in_data[j][i])
        {
            fprintf(stderr, "Bogus val in loc [%d][%d] in compressed dset, want %ld got %ld\n", j, i,
(long)in_data[j][i], (long)out_data[j][i]);
            num_errs++;
        }
}

/* Terminate access to the data set */
status = SDendaccess (sds_id);

/* Terminate access to the SD interface and close the file */
status = SDend (sd_id);

/* Return the number of errors that's been kept track of so far */
return 0;
}

```

[Return to "Szip Compression in HDF5"](#)