



Development Status Report

N°: 1

ERF-HDF5 Status Report

Compression using HDF5 Filters

Business or Project N°: CSM 2009: New Output Form a t

Rev.	WRITTEN BY			CHECKED BY			APPROVED BY		
	Name(s)	Date	Visa	Name (s)	Date	Visa	Name (s)	Date	Visa
A	AFL	Jul'09							

Filing:Name of the filing responsible: **AFL**

Date : Jul'09

Visa :

This document is organized by:

- Pages:
- Attachments: -

Consultation:

Free :
 Available at the internal website
 ESI Group only : x
 Diffusion list only :
 Others ESI Group's Subsidiaries:

Reviews**Reviewed pages****A**

Original document

DISTRIBUTION :

Company	Name(s)	Total	Partial

ESI Group S.A.

Paris – Adresse du Siège Social : 6 Rue Hamelin – BP 2008-16 – 75761 PARIS CEDEX 16 - Tél. : 33 (0)1.53.65.14.14 –

Fax : 33 (0)1.53.65.14.12

Rungis PARC D AFFAIRES SILIC - 99 RUE DES SOLETS – BP 80112 – 94513 RUNGIS CEDEX - Tel : 33 (0)1.41.73.58.00 –

Fax : 33 (0)1.46.87.72.02

Lyon : "Le Discover" – 84 Bd. Vivier Merle – 69485 LYON CEDEX 03 – Tél. : 33 (0)4.78.14.12.00 – Fax : 33 (0)4.78.14.12.02

Aix en Provence : 5 Parc Club du Golf - 13856 AIX-EN-PROVENCE CEDEX 3 – Tél. : 33 (0)4.42.97.65.30 – Fax : 33 (0)4.42.97.65.39

Compiègne : 20 Rue du Fonds Pernant – Immeuble Thalassa – 60471 COMPIEGNE CEDEX - Tél. : 33 (0)3.44.30.43.60 –

Fax : 33 (0)1.44.86.87.77

Montpellier : Parc club du Millénaire – Bat 15 - 1025 Rue Henri Becquerel – 34000 MONTPELLIER - Tél. : +33 (0)4 67 64 50 43

RCS PARIS B381 080 225 000 26

Document status: DRAFT

Prepared by : Dr. Andreas Floss

Phone number : +41 21 6938321

E-mail address : afl@esi-group.com

Reviewed by :

<reviewer>

Dr. Raymond Ni

Thorsten Queckbörner

<email>

rni@esi-group.com

tqu@esigmbh.de

<phone number>

CSM Solver Development

ERF-HDF5 Status Report Compression using HDF5 Filters

ESI Software

July 2009
ESI Group
Dr. A. Floss

Contents

1	Introduction	5
2	HDF5 Filters.....	5
3	Tests.....	7
4	Remarks	9

1 Introduction

ERF-HDF5 is the new ESI Result output file standard. It is based on the file standard HDF5 (The HDF group, <http://hdfgroup.com/HDF5>).

The following ESI documents are available:

- /1/ ERF-HDF5 Specification Version 1.0 – ERF_HDF5_Specs_1.0.pdf
- /2/ ERF-HDF5 CSM Specification Version 1.0 – ERF-CSM_Specs_1.0.pdf
- /3/ PAM-CSM Reference Manual 2009 – Innovation section
- /4/ ERF-HDF5 Survey – erf_hdf5_survey.pdf
- /5/ Prototype API – in clearcase under /vobs/csm_pvob/csm-2009.0.erf
- /6/ Sample files – ftp server dropbox.esi.fr, account: erfscsupp, passw: \$!erfhdf5!\$

This document describes the usage of HDF5 filters for the compression of datasets. These compression filters are supposed to be an output option to be released with CSM version 2010.

Further, ESI is planning to collaborate with the Fraunhofer Institute SCAI about using the commercial compression software FEMZIP (www.scai.fraunhofer.de/kompression.html).

FEMZIP uses more radical compression schemes and can therefore achieve better compression rates compared to the methods described here. However, a price has to be paid in CPU time. Another disadvantage is that each receiving program has to be equipped with the FEMZIP software to uncompress the files, whereas the usage of native HDF5 filters does not require any changes of the reading program.

2 HDF5 Filters

Each data that is written or read to or from an HDF5 file, is passed through a sequence of processing steps, the HDF5 data pipeline. This pipeline may also contain filters to manipulate data. There are a number of filters available in HDF5 that can be directly added to the pipeline in an arbitrary order. It is also possible to add user-defined filters.

In this project three types of filters have been considered, the scale-offset filter, the shuffle filter and the deflate filter. The following table gives a brief description for each.

Table: HDF5 filter types

HDF5 filter	applicable to	Description
scaleoffset	Floating-point and Integer type	Integer data: The filter determines the min number of bits and stores data with the reduced bit-length. Floating-point data: After doing some offset and scaling operations data is transformed to integer and then treated as described above. For floating-point data the scaleoffset filter is lossy in nature!
shuffle	Floating-point, Integer and character type	Reorders the bytes of a data block. Example: three data elements of a 4-byte datatype stored as 012301230123, shuffling will re-order data as 000111222333 The shuffle filter is usually applied to a dataset immediately prior to the use of a compression filter to improve the compression ratio.
deflate	Floating-point, Integer and character type	same algorithm as used by the GNU gzip program (zlib)

The ERF-API functions have been extended in a way that each of the three filters can be switched on or off individually. However, the filters are applied in an unalterable order as shown in the figure below. Note that the shuffle filter alone does not do any compression and therefore should only be used in conjunction with the deflate filter.

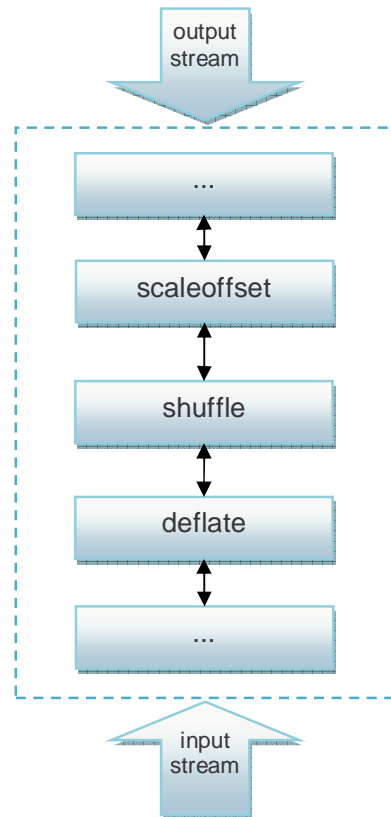


Figure: HDF5 filter pipeline

In the ERF-API the filters are only applied on datasets that contain arrays. There are only two parameters that have to be passed over, the scale exponent for the scaleoffset filter and the aggression level (0 to 9) for the deflate (gzip) filter.

The scale exponent for the scaleoffset filter controls the precision of the stored floating point data. For example, if the exponent is set to 2, the number 104.561 will be scaled to 10456.1. All numbers after the point (in the example the number 1) are supposed to be not significant and thrown off in the process of converting the data from floating point to integer. Before scaling the numbers the scaleoffset filter automatically does a range transformation by – as the name suggests – offsetting the data with the lowest number. Note that the data range has an impact on the compression ratio!

Table: Example demonstrating the operations of the scaleoffset filter

original data	offset	scale (exp 2)	float→int	data read
10.000	0.000	0.00	0	10.00
11.054	1.054	105.40	105	11.05
20.286	10.286	1028.60	1028	20.28

3 Tests

The HDF5 filters have been tested in various combinations using the stand-alone tool erfmerge (program to merge PAMCRASH DMP domain files together). This tool allows the user to control the usage of the compression filters via a bit-code (flag to switch on or off). The scale exponent for the scaleoffset filter can be specified for each ERF variable via an input file.

Table: Test cases

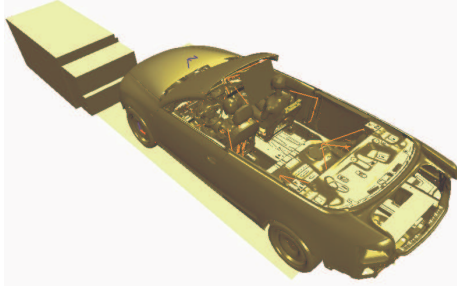
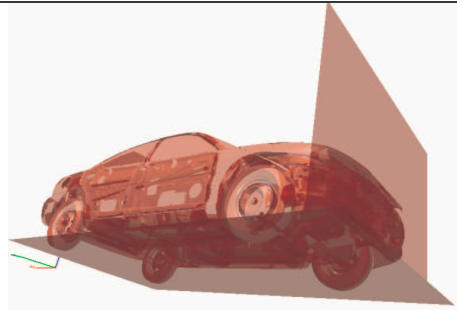

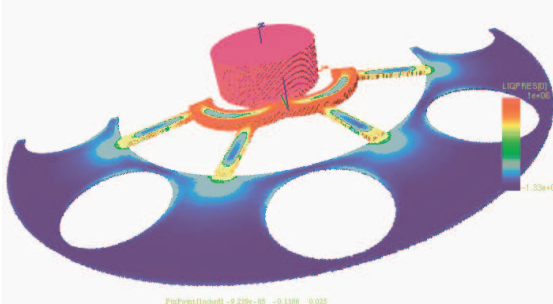
Test case	Model
Frontal Crash Audi Nodes: 685k Shells: 647k Solids: 41k Bars: 12k Contacts: 75	
Frontal Crash Neon Nodes: 300k Shells: 274k Solids: 3k Bars: 4k Contacts: 2	
Frontal Crash Passat Nodes: 1043k Shells: 980k Solids: 47k Bars: 24k Contacts: 7	
breakAPM (Finite-Volume model of the PROCAST porosity modul) Nodes: - Cells: 770k Solids: - Bars: - Contacts: -	

Table: Test results

file	flag	flag bin	time to convert [sec]	A: erf5 uncompr [MB]	B: erf5 compress [MB]	C: DSY+THP [MB]	A/B	A/C	C/B
passat	191	10111111	02:15	2682	487	1599	5.51	1.68	3.28
	255	11111111	01:47	2682	962	1599	2.79	1.68	1.66
	199	11000111	01:42	2682	918	1599	2.92	1.68	1.74
	63	111111	01:09	2682	645	1599	4.16	1.68	2.48
	7	111	02:26	2682	1017	1599	2.64	1.68	1.57
	192	11000000	00:57	2682	1657	1599	1.62	1.68	0.97
neon	191	10111111	00:18	514	140	323	3.68	1.59	2.31
	255	11111111	00:24	514	229	323	2.25	1.59	1.41
	199	11000111	00:23	514	223	323	2.30	1.59	1.45
	63	111111	00:21	514	210	323	2.45	1.59	1.54
	7	111	00:36	514	294	323	1.75	1.59	1.10
	192	11000000	00:13	514	304	323	1.69	1.59	1.06
audi	191	10111111	04:10	4380	986	4393	4.44	1.00	4.46
	255	11111111	03:07	4380	1676	4393	2.61	1.00	2.62
	199	11000111	03:18	4380	1614	4393	2.71	1.00	2.72
	63	111111	02:33	4380	1830	4393	2.39	1.00	2.40
	7	111	04:07	4380	2432	4393	1.80	1.00	1.81
	192	11000000	01:52	4380	3035	4393	1.44	1.00	1.45
breakAPM2 (procast)	191	10111111	00:16	350	55	n/a	6.37	n/a	n/a
	255	11111111	00:24	350	142	n/a	2.46	n/a	n/a
	199	11000111	00:24	350	135	n/a	2.59	n/a	n/a
	63	111111	00:14	350	56	n/a	6.27	n/a	n/a
	7	111	00:48	350	117	n/a	2.99	n/a	n/a
	192	11000000	00:07	350	192	n/a	1.83	n/a	n/a

Legend for the compression flag bits:

Filter	Scaleoffset		Shuffle			Deflate (gzip)		
Datatype	Real	Int	Char	Real	Int	Char	Real	Int
Bit	7	6	5	4	3	2	1	0
Value	128	64	32	16	8	4	2	1

Remark (1):

For the scaleoffset filter, exponents between 2 and 5 have been used (for coordinates: 2, for stresses and strains 5, for velocities: 3). The gzip aggression level has been set to 5.

Remark (2):

Column A of the table shows the original file sizes of the uncompressed ERF files. Column B contains the file sizes after filtering and compression using the stand-alone tool erfmerge. Column C gives the file sizes of the old DSY/THP format as a reference (CSM-PAMCRASH only).

Remark (3):

The fourth column of the table contains the elapsed time, needed to read, convert and re-write the ERF file using the stand-alone tool erfmerge.

4 Remarks

All three tested types of filters are available in the API functions, the scaleoffset filter for floating-point and integer data, and the shuffle and deflate filters for all types of data (including character arrays). The argument lists of the block writing API-functions have been extended by two additional arguments:

- the compression flag (as described above) and
- an integer array for the filter parameters (today it has only two elements: the precision scaling exponent for the scaleoffset filter and the aggression level for the deflate filter).

The user should have full access to these parameters in order to let him find the best combination and settings depending on the characteristics of his model (also to find the best compromise between time overhead and compression ratio). Practically, in an industrial environment the IT-departments may have to do a few tests to provide the users with proper settings.

More efforts have to be made to find appropriate precision scale exponents for each output variable. These exponents should be provided in a file containing a list of ERF variable strings as shown in the table below.

Table: File to specify the precision scale exponents per ERF variable

# scaleoffset exponents	
COORDINATE	2
VELO	3
SXYZ	5
EPSE	5
...	

Note that the exponents may depend on the unit system used for the simulation. Consider the following example:

original data	offset	scale (exp 3)	float→int	data read
1.0000	0.0000	0.0	0	1.000
2.6789	1.6789	1678.9	1678	2.678
3.6789	2.6789	2678.9	2678	3.678

The same values, but shifted by a factor 1000:

original data	offset	scale (exp 3)	float→int	data read
0.0010000	0.0000000	0.0000	0	0.001
0.0026789	0.0016789	1.6789	1	0.002
0.0036789	0.0026789	2.6789	2	0.003

With the same scaling exponent of 3 you will lose more precision as in the case above. To get the same precision you need to shift your exponent by a factor 1000 too (=> exp 6).