VERAOUT - VERA HDF5 Output Specification

Andrew Godfrey

Scott Palmtag

Greg Davidson

Ben Collins

Revision 1
Milestone L3:PHI.VCS.P8.01
May 2, 2014

Introduction

This document describes the revised specification (Revision 1) for the format of the common VERA Output file (VERAOUT). VERAOUT is a binary HDF5 file that all of the physics codes in VERA-CS produce. Having all physics codes write to a single file simplifies post-processing and data comparison. The edits and format written to the VERAOUT file is described below.

This specification covers common scalar and array data typically produced by neutronic and fuel performance codes. This document will be periodically revised to add additional data as more physics codes and/or more data is added to VERA-CS. A version number is written to the HDF file to keep track of different versions of the VERAOUT so that backwards compatibility may be maintained.

A sample VERAOUT reader has been written to serve as an example for users wishing to do data post-processing. Information on this program is included below.

Dimensions

For the purposes of this specification, the following dimensions are defined:

- NASSX – Maximum number of assemblies across the core horizontally in full symmetry
- NASSY – Maximum number of assemblies down the core vertically in full symmetry
- NPIN – Maximum number of fuel pins across a fuel assembly in the core. Assemblies are assumed to be symmetric.
- NAX – Number of axial levels edited in the fuel
- NASS – Total number of fuel assemblies in the problem considering the symmetry of the calculation.

When reading a VERAOUT data file, the dimensions are determined by querying the datasets. The dimensions themselves are not written to the data file.

HDF Group Structure

The datasets are organized on the HDF5 file according to "groups". There is an HDF group named "CORE" that contains the core geometry and other variables that correspond to the CORE block in the common input. There are multiple HDF groups named "STATE_nnnn", where "nnnn" is a four-digit statepoint number. If a statepoint has less than 4 digits, the statepoint number should be padded with zeroes in the front of the number (i.e. 0001, 0002, etc.). If the statepoint has more than 4 digits (highly unlikely), it should print a 5 digit number. If a physics code does not support multiple statepoints, it should write data to STATE_0001.

All HDF group names (STATE_nnnn and CORE) are all uppercase. HDF datasets are all lowercase.

An example of a typical HDF5 group structure is:

```
./
./CORE/
./STATE_0001/
./STATE_0002/
./STATE_0003/
```

There are several restrictions on a VERAOUT file:

- The size of the core must be constant throughout the file
- The symmetry of the calculation must be constant throughout the file
- The number of axial elevations must be constant throughout the file
- The VERAOUT version must be constant throughout the file

The reason for having the geometry constant throughout the file to simplify the post processing. If the data sizes do not change, the reading code does not have to check and reallocate the data at each statepoint. If the user does change the geometry, the new data should be written to a different HDF output file.

Datasets

The following datasets are written to the VERAOUT file. Unless marked as REQUIRED, all distributions are optional. Most datasets are optional because different physics codes produce different datasets. The user should always check for the presence of these datasets before attempting to read them.

Table 1 lists all of the datasets in the root HDF5 group.

Table 2 lists all of the datasets in the CORE HDF5 group.

Table 3 lists all of the datasets in the HDF5 statepoint groups (./STATE_nnnn).

A detailed description of each dataset follows the tables.

Note that datasets are written to the file in all lower-case. (Group names are upper case, but datasets are lower case). Even though the tables show the datasets in upper case, they should be lowercase on the HDF file.

All array dimensions are given in Fortran order. C order is the opposite of the Fortran order. For example, an array with Fortran size (i,j,k) would have C order [k][j][i].

All pin-by-pin arrays are 4-dimensional and should have the ordering (NASS, NAX, NPIN, NPIN). Pin arrays should only be written for the the assemblies in the calculated geometry. Assemblies on the axes of symmetry should be expanded to full symmetry (though this may not be used). The location in the core of an assembly is given by its index in CORE_MAP.

The order of arrays has been selected so that the maps will show up in the GUI editor "HDFview" in the same orientation as the input file and the calculated geometry. If there are any discrepancies in the array order, it should be written such that HDFview shows the orientation in the actual geometry.

Table 1: Datasets in Root Group

Dataset	Dimension	Type	Required
input_parameter_list	group	(various)	
veraout_version	0	integer	Required
title	0	string	Required
build_date	string	string	
run_date	string	string	Required
run_time	string	string	Required

Table 1: Datasets in the CORE Group

Dataset	Dimension	Type	Required
core_map	2	integer	Required
core_sym	0	integer	Required
axial_mesh	1	double	Required
rated_power	0	double	Required*
rated_flow	0	double	Required*
pin_areas	4	double	
pin_loadings	4	double	
apitch	0	double	
core_name	string	string	
core_loading	0	string	

* means that the dataset is required in some instances. See text for details.

Table 3: Datasets in Statepoint Group ./STATE_nnnn

Dataset	Dimension	Type	Required
boron	0	double	Required*
keff	0	double	
power	0	double	Required*
flow	0	double	
tinlet	0	double	
hours	0	double	Required*
exposure	0	double	Required*
exposure_efpd	0	double	
exposure_core	0	double	
pin_powers	4	double	
pin_exposures	4	double	
pin_fueltemps	4	double	
keff_sigma	0	double	
pin_powers_sigma	4	double	

* means that the dataset is required in some instances. See text for details.

Root Dataset Descriptions

INPUT_PARAMETER_LIST - (Special Group) - This is a special group in the root level that contains a copy of the XML input file. Subgroups should be created that are equivalent to the sublist levels in XML.

TITLE - String (Required) - Problem Title from the CASEID input parameter.

VERAOUT_VERSION - Integer (Required)— VERAOUT Specification version. This document refers to version 1. Note that the version number should only be incremented if the data format changes, or additional required data is added. The version number should not be incremented if additional optional data is added.

BUILD_DATE - String - Character string showing the date the code was built. This string is useful for quality assurance and reproducibility of results.

RUN_DATE - String (Required) - Character string showing the date the calculation was run. This string is useful for quality assurance and reproducibility of results.

RUN_TIME – String (Required) - Character string showing the time the code was run. This string is useful for quality assurance and reproducibility of results.

CORE Dataset Descriptions

The dataset names are shown in all caps below, but they are all lowercase on the HDF file.

CORE_MAP(NASSX,NASSY) – Integer (Required) – Square array indicating the 1-based index of the assembly from 1 to NASS. Non-fueled locations should have a 0. For quarter symmetry, the map should be expanded to fuel symmetry. For a 3x3 core with quarter reflective symmetry, this should look like:

```
4  3  4
2  1  2
4  3  4
```

CORE_SYM – Integer (Required) – Numeric flag to indicate the calculational symmetry of the case (consistent with CORE_MAP and NASS). Supported values are: 1=Full, 4=Quarter.

CORE_NAME – String – Name of reactor core from CORE input block

CORE_LOADING – Double – Total heavy metal mass in full-core (MTHM)

APITCH –Double Assembly pitch (cm)

AXIAL_MESH(NAX+1) - Double (Required) - The axial boundaries (in cm) of the axial fuel levels used for edits/output. The first array element may or may not be 0.0 cm depending on the problem.

RATED_POWER – Double (Required*) – Rated power in the full-core (MWt). This dataset is required if the POWER dataset is present.

RATED_FLOW – Double (Required*) –Rated flow in the full core (units TBD). This dataset is required if the FLOW dataset is present.

PIN_AREAS(NASS,NAX,NPIN,NPIN) – Double – Array of pin areas. If this dataset is not present, it should be assumed that all pins have the same radial areas. The pin areas are needed to convert linear power to volume-averaged power density. *This dataset should only be specified if the area of the fuel pins is not constant, i.e. annular fuel.*

PIN_LOADINGS(NASS,NAX,NPIN,NPIN) – Double – Fuel pin heavy metal density (g/cc). This array is used for averaging pin exposures.

STATE Dataset Descriptions

The dataset names are shown in all caps below, but they are all lowercase on the HDF file.

BORON – Double (Required*) – The reactor boron concentration (either calculated or specified).   This card is required if the boron concentration is not zero.

KEFF – Double – The reactor eigenvalue (either calculated or specified).

KEFF_SIGMA – Double – Eigenvalue uncertainty (from Monte Carlo codes)

POWER – Double (Required*) – Reactor power in percent. Actual power is determined by multiplying POWER by RATED_POWER. This value is used to normalize the pin power arrays. This dataset is required if PIN_POWERS are specified.

FLOW – Double – Reactor flow in percent.

TINLET – Double – Core inlet temperature. Units are degrees Celsius.

EXPOSURE – Double (Required*) – Cycle Exposure. Units GWD/MT(HM). Required if performing a depletion calculation, or starting from a depleted core.

EXPOSURE_EFPD – Double – Cycle Exposure. Units EFPD.

EXPOSURE_CORE – Double – Core Exposure. Units GWD/MT(HM)

HOURS –Double (Required*) – Transient time in hours. Required if the code is running an hours transient.

PIN_POWERS(NASS,NAX,NPIN,NPIN) – Double - Array of normalized pin powers . Pin powers are normalized such that the average linear power in the full-core is exactly one. (See discussion above on format of pin arrays.)

PIN_FUELTEMPS(NASS,NAX,NPIN,NPIN) – Double - Array of pin volume averaged fuel temperatures. (See discussion above on format of pin arrays.) Units are degrees Celsius.

PIN_EXPOSURES(NASS,NAX,NPIN,NPIN) – Double - Array of all pin exposures. (See discussion above on format of pin arrays.) Units GWD/MT.

PIN_POWERS_SIGMA(NASS,NAX,NPIN,NPIN) – Double – Array of pin uncertainty values (from Monte Carlo code)

Example Data
NASSX=2
NASSY=3
NAX=49
NASS=193
NPIN=17

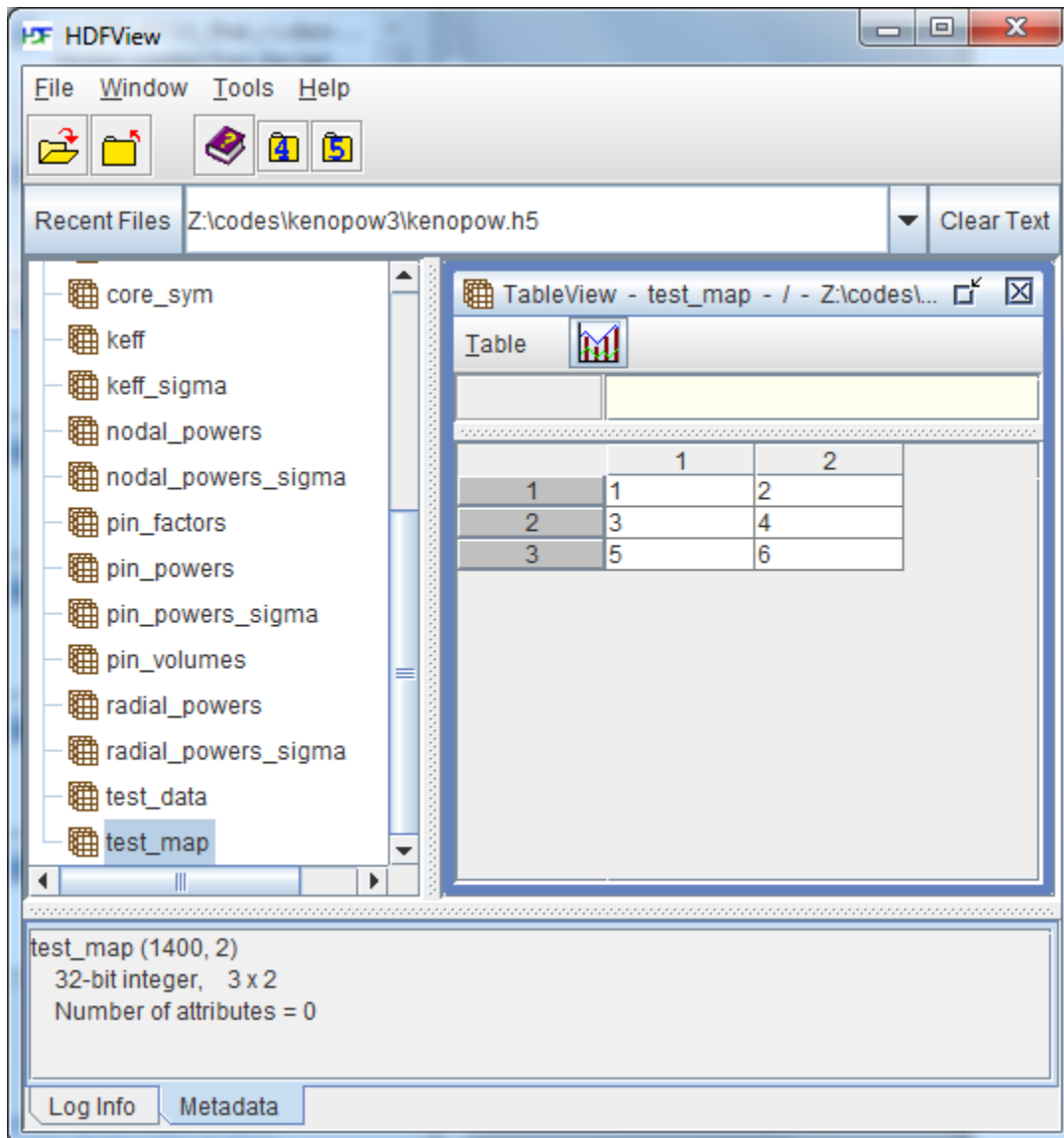core_map(1,1)=1
core_map (2,1)=2
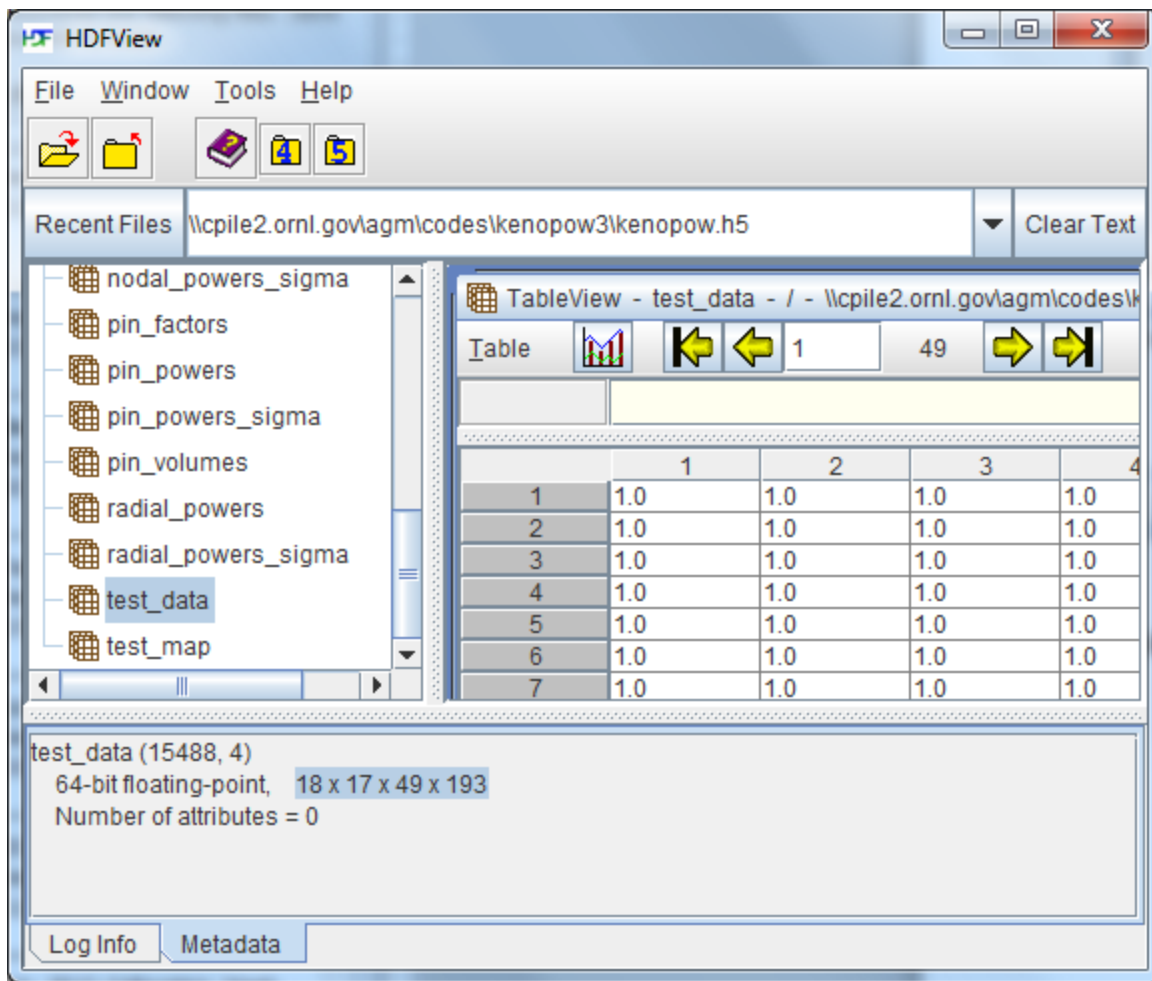core_map (1,2)=3
core_map (2,2)=4
core_map (1,3)=5

core_map (2,3)=6

core_map = 1 2
        3 4
        5 6

HDFView

The data arrays appear correctly arranged in HDFView, but the indices and array dimensions are reversed because HDFView shows the "C" ordering.

H5Dump

H5Dump produces output consistent with HDFView. H5Dump also uses "C" order when showing array indices.

```
[agm@orthanc ~]$ h5dump -d core_map kenopow.h5
HDF5 "kenopow.h5" {
DATASET "core_map" {
   DATATYPE  H5T_STD_I32LE
   DATASPACE  SIMPLE { ( 3, 2 ) / ( 3, 2 ) }
   DATA {
   (0,0): 1, 2,
   (1,0): 3, 4,
   (2,0): 5, 6
   }
}
}
```

```
[agm@orthanc ~]$ h5dump -d pin_powers kenopow.h5
HDF5 "kenopow.h5" {
DATASET "pin_powers" {
   DATATYPE  H5T_IEEE_F64LE
   DATASPACE  SIMPLE { ( 18, 17, 49, 193 ) / ( 18, 17, 49, 193 ) }
   DATA { … }
}
}
```

Core Symmetry

The core_map is always written in full-core geometry. If the problem is run with qtr-symmetry, only the pin powers in the active geometry will be written to the output file and the core_map will contain duplicate assembly numbers that reflect the symmetric partners.

For example, if a 3x3 assembly core was run with no symmetry, the core_map would look like this:

```
1 2 3
4 5 6
7 8 9
```

to show that there are 9 unique assemblies in the problem.

If the same problem was run in mirror qtr-symmetry, the core map would look like this:

```
4 3 4
2 1 2
4 3 4
```

to show that there are only 4 unique assemblies in the problem.

The pin powers will only be written in the correct/physical orientation for the assembly in the bottom right quadrant (SE quadrant) because this is the calculated quadrant. The assemblies in the other three quadrants (NE, NW, and SW) need to be rotated/flipped to place them in the correct physical orientation.

Sample Reader Program

A sample Fortran program has been written to read VERAOUT files. This sample problem reads two VERAOUT files and compares the boron and eigenvalue values to check to make sure they are within a

specified tolerance. This program is used in automated testing, but also serves as a useful example program for users attempting to do data post-processing.

This program is located in the VERA git repository in the following location:

```
…/VERA/VERAInExt/verain/src/veradiff.f90
```

Another set of example tools to read VERAOUT files is located in the GITHUB repository:

https://github.com/palmtag/VERAout-tools

Future Development

The VERAOUT specification is intended to be a "living" document. This document is intended to be updated as additional data is generated and/or needed by VERA-CS. A version number is written to the VERAOUT data file to allow backwards compatibility in case the format changes in future versions.

Some examples of datasets that may be added in the future include pin crud values, pin steaming rates, and pin PCI indicators.

Another set of data that may be added is channel-based data from subchannel codes. In order to add channel-based data, the storage format for channels between assemblies must be worked out.

Another area of future research is to determine how to visualize the HDF output in Visit or Paraview using an XDMF specification. See for example:
http://www.visitusers.org/index.php?title=Using_XDMF_to_read_HDF5