Cinema Simple Database Specification

v1.1, rev1.0

David Rogers (LANL)

David DeMarle (Kitware, Inc.)

James Ahrens (LANL)

John Patchett (LANL)

Cinema Database Specification v1.1 rev1.0

Updated: **March 5, 2014**

Authors: **David Rogers, Dave DeMarle, Jim Ahrens, John Patchett**

I. Overview

Extreme scale scientific simulations are leading a charge to exascale computation, and data analytics runs the risk of being a bottleneck to scientific discovery. Due to power and I/O constraints, we expect in situ visualization and analysis will be a critical component of these workflows. Options for extreme scale data analysis are often presented as a stark contrast: write large files to disk for interactive, exploratory analysis, or perform in situ analysis to save detailed data about phenomena that a scientists knows about in advance. We present a novel framework for a third option – a highly interactive, image-based approach that promotes exploration of simulation results, and is easily accessed through extensions to widely used open source tools. This in situ approach supports interactive exploration of a wide range of results, while still significantly reducing data movement and storage.

More information about the overall design of Cinema is available in the paper, *An Image-based Approach to Extreme Scale In Situ Visualization and Analysis*, which is available at the following link:

https://datascience.lanl.gov/data/papers/SC14.pdf

A Cinema database is a collection of data that supports this image-based approach to interactive data exploration. It is a set of images and associated metadata, and is defined and an example given in the following sections.

This specification is an update of Cinema Simple Specification v1.0, rev3.0, LA-UR-15-20572

Use cases

A Cinema Database supports the following three use cases:

1. Searching/querying of meta-data and samples. Samples can be searched purely on metadata, on image content, on position, on time, or on a combination of all of these.
2. Interactive visualization of sets of samples.
3. Playing interactive visualizations, allowing the user on/off control of elements in the visualization.

II. Cinema "simple" Database Specification v1.1

This document describes release v1.1 of the Cinema "simple" Database, and is intended to provide information about how to read or write data in this format.

The "simple" database is:

- a collection of images sampled by:
  - a single Cinema Camera. A Cinema Camera can be either *static* or *spherical*, indicating how the camera samples views. The Cinema database is a collection of images for all (time, position) pairings defined by the export script.
  - zero or one clipping plane operators, with an associated range of clipping values,
  - zero or one contour operators, and an associated range of contour values.
- a JSON file that is a specific instance of the Cinema "simple" Database schema shown in Section III.

The Cinema Database is implementation agnostic. This database specification separates the metadata description of a set of images from the implementation of how these images are stored. In particular, if the images for a specific instance of a database are stored on disk, *the design of the directory structure, metadata files, and image filenames on disk is entirely up to the person writing the data.* Instead, this specification expects a database of URIs that maps metadata to data products required by the specification.

Organization of this document.
This document includes the following sections:

Section III:	Outline of a Cinema "simple" Database schema in JSON data format
Section IV:	An example of a the JSON file for a specific database instance

III. Outline of a Cinema "simple" Database schema in JSON data format

This schema encodes the information for the above definitions, and contains the required information needed to create a specific instance of a Cinema "simple" Database.

```
{
"type"    : "simple",
"version" : "1.1",
"metadata": {
    "type": "parametric-image-stack"
},
"name_pattern": <valid URL with substitutions>
"arguments": {
    <name>: {
        "default": <value>,
        "label": <string>,
        "type": <one of [“list”, "range"]>,
        "values": [ list of unique values ]
    },
}
}
```

Details

Header information

This is required database header information, and values must be as defined below.

```
"type"    : "simple",
"version" : "1.1",
"metadata": {
    "type": "parametric-image-stack"
},
```

name_pattern

This string shows how to construct valid paths to cinema database images. It must be aa path relative to the location of the `<database>.json` file.

```
"name_pattern" : <valid file path with substitutions>

Ex:     "name_pattern" = "{time}/{phi}/{theta}/image.png" (several subdirectories)
        "name_pattern" = "{time}_{phi}_{theta}_image.png" (all images in one directory)
```

arguments

The rest of the file consists of arguments that define the valid values for the variables in the name_pattern value. They have the following form:

```
<string>: {
    "default": <value>,
    "label": <string>,
    "type": "range",
    "values": [ list of unique values ]
}
```

VI. Example

This example is based on above a JSON schema outline. Because there is no `'phi'` or `'theta'` value for an argument, this example shows a static Cinema camera.

```
{
"type"    : "simple",
"version" : "1.1",
"metadata": {
    "type": "parametric-image-stack"
},
"name_pattern": "{time}/{slice}.jpg",
"arguments": {
    "slice": {
        "default": -17.3205,
        "label": "slice",
        "type": "range",
        "values": [
            -17.3205,
            -13.4715,
            -9.6225,
            -5.7735,
            -1.9245,
            1.9245,
            5.7735,
            9.6225,
            13.4715,
            17.3205
        ]
    },
    "time": {
        "default": "0.000000",
        "label": "time",
        "type": "range",
        "values": [
            "0.000000",
            "0.500000"
        ]
    }
}
}
```