ahds Documentation

Release 0.2.3

Paul K. Korir, PhD

Jan 15, 2021

Contents

1 ahds **3**
- 1.1 Overview . . . 3
- 1.2 License . . . 4
- 1.3 Use Cases . . . 4
- 1.4 Installation . . . 4
- 1.5 Getting Started . . . 4
- 1.6 Future Plans . . . 8

2 ahds package **9**
- 2.1 ahds . . . 9
- 2.2 ahds.grammar module . . . 10
- 2.3 ahds.header module . . . 11
- 2.4 ahds.data_stream module . . . 14

3 Indices and tables **17**

Python Module Index **19**

Index **21**

CHAPTER 1

ahds

Contents

- ahds
  - *Overview*
  - *License*
  - *Use Cases*
  - *Installation*
  - *Getting Started*
  - *Future Plans*

1.1 Overview

ahds is a Python package to parse and handle Amira (R) files. It was developed to facilitate reading of Amira (R) files as part of the EMDB-SFF toolkit.

Note: Amira (R) is a trademark of Thermo Fisher Scientific. This package is in no way affiliated with with Thermo Fisher Scientific.

1.2 License

ahds is free software and is provided under the terms of the Apache License, Version 2.0.

```
Copyright 2017 EMBL - European Bioinformatics Institute

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either express or implied. See the License for the specific
language governing permissions and limitations under the License.
```

1.3 Use Cases

- Detect and parse Amira (R) headers and return structured data
- Decode data (HxRLEByte, HxZip)
- Easy extensibility to handle previously unencountered data streams

ahds was written and is maintained by Paul K. Korir but there is a list of contributors. Feel free to join this initiative.

1.4 Installation

ahds works with Python 2.7, 3.5, 3.6 and 3.7. It requires numpy to build.

```
pip install numpy
```

Afterwards you may run

```
pip install ahds
```

Note: Figure out a way to avoid the need for numpy as part of the build.

1.5 Getting Started

You can begin playing with ahds out of the box using the provided console command ahds.

```
me@home ~$ ahds ahds/data/FieldOnTetraMesh.am
*******************************************************************************************************
AMIRA (R) HEADER AND DATA STREAMS
-------------------------------------------------------------------------------------
 ↪-----------------------------------------------------
```

(continues on next page)

(continued from previous page)

```
+-ahds/data/FieldOnTetraMesh.am
↪           AmiraFile [is_parent? True ]
|  +-meta
↪               Block [is_parent? False]
|  |  +-file: ahds/data/FieldOnTetraMesh.am
|  |  +-header_length: 182
|  |  +-data_streams: 1
|  |  +-streams_loaded: False
|  +-header
↪         AmiraHeader [is_parent? True ]
|  |  +-filetype: AmiraMesh
|  |  +-dimension: 3D
|  |  +-format: BINARY
|  |  +-endian: BIG
|  |  +-version: 2.0
|  |  +-extra_format: None
|  |  +-Parameters
↪               Block [is_parent? False]
|  |  +-Tetrahedra
↪               Block [is_parent? False]
|  |  |  +-length: 23685
|  +-data_streams
↪               Block [is_parent? False]
********************************************************************************
```

The `ahds` command takes the following arguments

```
me@home ~$ ahds -h
usage: ahds [-h] [-s] [-d] [-l] file [file ...]

Python tool to read and display Amira files

positional arguments:
  file                a valid Amira file with an optional block path

optional arguments:
  -h, --help          show this help message and exit
  -s, --load-streams  whether to load data streams or not [default: False]
  -d, --debug         display debugging information [default: False]
  -l, --literal       display the literal header [default: False]
```

You can specify a **dotted path** after the filename to only render that the content of that field in the header:

```
me@home ~$ ahds ahds/data/FieldOnTetraMesh.am header
********************************************************************************
ahds: Displaying path 'header'
--------------------------------------------------------------------------------
↪-----------------------------------------
+-header
↪         AmiraHeader [is_parent? True ]
|  +-filetype: AmiraMesh
|  +-dimension: 3D
|  +-format: BINARY
|  +-endian: BIG
|  +-version: 2.0
|  +-extra_format: None
|  +-Parameters
↪               Block [is_parent? False]
```

(continues on next page)

(continued from previous page)

```
|  +-Tetrahedra                                                                                                  ␣
↪               Block [is_parent? False]
|  |  +-length: 23685
```

For debugging you can display the literal header (the exact header present in the file) using the `-l/--literal` flag. Also, you can display the parsed data structure using the `-d/--debug` flag.

```
me@home ~$ ahds --literal --debug ahds/data/FieldOnTetraMesh.am
********************************************************************************************************************
ahds: Displaying literal header
--------------------------------------------------------------------------------------------
↪-------------------------------------------
# AmiraMesh 3D BINARY 2.0
# CreationDate: Tue Nov  2 11:46:31 2004


nTetrahedra 23685

TetrahedronData { float[3] Data } @1
Field { float[3] f } Constant(@1)

# Data section follows
********************************************************************************************************************
ahds: Displaying parsed header data
--------------------------------------------------------------------------------------------
↪-------------------------------------------
[{'designation': {'dimension': '3D',
                  'filetype': 'AmiraMesh',
                  'format': 'BINARY',
                  'version': '2.0'}},
 {'comment': {'date': 'Tue Nov  2 11:46:31 2004'}},
 {'array_declarations': [{'array_dimension': 23685,
                          'array_name': 'Tetrahedra'}]},
 {'data_definitions': [{'array_reference': 'Tetrahedra',
                        'data_dimension': 3,
                        'data_index': 1,
                        'data_name': 'Data',
                        'data_type': 'float'},
                       {'array_reference': 'Field',
                        'data_dimension': 3,
                        'data_index': 1,
                        'data_name': 'f',
                        'data_type': 'float',
                        'interpolation_method': 'Constant'}]}]

********************************************************************************************************************
AMIRA (R) HEADER AND DATA STREAMS
--------------------------------------------------------------------------------------------
↪-------------------------------------------------
+-ahds/data/FieldOnTetraMesh.am                                                                                ␣
↪           AmiraFile [is_parent? True ]
|  +-meta                                                                                                      ␣
↪               Block [is_parent? False]
|  |  +-file: ahds/data/FieldOnTetraMesh.am
|  |  +-header_length: 182
|  |  +-data_streams: 1
```

(continues on next page)

(continued from previous page)

```
|  |  +-streams_loaded: False
|  +-header                                                                                                    ␣
↪         AmiraHeader [is_parent? True ]
|  |  +-filetype: AmiraMesh
|  |  +-dimension: 3D
|  |  +-format: BINARY
|  |  +-endian: BIG
|  |  +-version: 2.0
|  |  +-extra_format: None
|  |  +-Parameters                                                                                             ␣
↪              Block [is_parent? False]
|  |  +-Tetrahedra                                                                                             ␣
↪              Block [is_parent? False]
|  |  |  +-length: 23685
|  +-data_streams                                                                                              ␣
↪              Block [is_parent? False]
*******************************************************************************************************
```

By default, data streams are not read — only the header is parsed. You may obtain the data streams using the `-s/--load-streams` flag.

```
me@home ~$ ahds --load-streams ahds/data/FieldOnTetraMesh.am
*******************************************************************************************************
AMIRA (R) HEADER AND DATA STREAMS
-------------------------------------------------------------------------------------------------------
↪-----------------------------------------------------
+-ahds/data/FieldOnTetraMesh.am                                                                              ␣
↪           AmiraFile [is_parent? True ]
|  +-meta                                                                                                    ␣
↪              Block [is_parent? False]
|  |  +-file: ahds/data/FieldOnTetraMesh.am
|  |  +-header_length: 182
|  |  +-data_streams: 1
|  |  +-streams_loaded: True
|  +-header                                                                                                  ␣
↪         AmiraHeader [is_parent? True ]
|  |  +-filetype: AmiraMesh
|  |  +-dimension: 3D
|  |  +-format: BINARY
|  |  +-endian: BIG
|  |  +-version: 2.0
|  |  +-extra_format: None
|  |  +-Parameters                                                                                           ␣
↪              Block [is_parent? False]
|  |  +-Tetrahedra                                                                                           ␣
↪              Block [is_parent? False]
|  |  |  +-length: 23685
|  +-data_streams                                                                                            ␣
↪              Block [is_parent? True ]
|  |  +-Data                                                                                                 ␣
↪ AmiraMeshDataStream [is_parent? False]
|  |  |  +-data_index: 1
|  |  |  +-dimension: 3
|  |  |  +-type: float
|  |  |  +-interpolation_method: None
|  |  |  +-shape: 23685
```

(continues on next page)

(continued from previous page)

```
|  |  |  +-format: None
|  |  |  +-data: [  0.8917308   0.9711809 300.       ],...,[  1.4390504   1.1243758␣
↪300.       ]
*******************************************************************************************
```

1.6 Future Plans

- Write out valid Amira (R) files

CHAPTER 2

ahds package

2.1 ahds

This module provides a simple entry-point for using the underlying functionality through the *AmiraFile* class which automatically handles both *AmiraMesh* and *HxSurface* files. An *AmiraFile* is also a *Block* subclass with special attributes *meta* - for metadata not explicitly provided in the file (such as *header_length*), *header* - for the parse header and *data_streams* with the actual data stream data.

The only required argument is the name of the file to be read. By default, data streams are loaded but can be turned off (for quick reading) by setting *load_stream=False*. Additional *kwargs* are passed to the *AmiraHeader* class call.

There is a *read* method which (if data streams have not yet been read) will read the data streams.

An *AmiraFile* object may be printed to view the hierarchy of entities above or passed to *repr* to view the instatiation call that represents it.

class ahds.**AmiraFile**(*fn*, *load_streams=True*, **args*, ***kwargs*)

Bases: ahds.core.Block

Main entry point for working with Amira files

read()

Read the data streams if they are not read yet

class ahds.**AmiraHeader**(*fn*, *load_streams=True*, **args*, ***kwargs*)

Bases: ahds.core.Block

Class to encapsulate Amira metadata and accessors to Amira (R) data streams

data_pointers(***kwargs*)

The list of data pointers together with a name, data type, dimension, index, format and length

NOTE: deprecated access the data defnitions for each data array through the corresponding attributes eg.: ah.Nodes.Coordinates instead of ah.data_pointers.data_pointer_1 ah.Tetrahedra.Nodes instead of ah.data_pointers.data_pointer_2 etc.

definitions(***kwargs*)

Definitions consist of a key-value pair specified just after the designation preceded by the key-word 'define'

NOTE: this property is deprecated access the corresponding attributes directly eg. ah.Nodes instead of ah.definitions.Nodes or ah.Tetrahedra instead of ah.defintions.Tetrahedra

designation(***kwargs*)
Designation of the Amira file defined in the first row

Designations consist of some or all of the following data:

- filetype e.g. `AmiraMesh` or `HyperSurface`
- dimensions e.g. `3D`
- format e.g. `BINARY-LITTLE-ENDIAN`
- version e.g. `2.1`
- extra format e.g. `<hxsurface>`

NOTE: this property is deprecated use the corresponding attributes of the AmiraHeader instead to access the above informations

classmethod from_file(***kwargs*)
Deprecated classmethod

load()
Public loading method

2.2 ahds.grammar module

2.2.1 grammar

We define an EBNF grammar for Amira (R) headers to extract all metadata. In addition to that, we also define how *HxSurface* files are structured.

This module also includes several helper functions that use the grammar resources:

- the *get_header* function returns only the header up to the first data stream; data is returned as a decoded string (*UTF-8*);
- the *parse_header* function applies the grammar to return a nested set of Python primitives to be transformed into an *AmiraHeader* object;
- the *get_parsed_data* function transparently applied both above functions given the Amira (R) filename

ahds.grammar.**detect_format**(*fn*, *format_bytes=50*, *verbose=False*, **args*, ***kwargs*)
Detect Amira (R) file format (AmiraMesh/Avizo or HyperSurface)

Parameters

- **fn** (*str*) – file name
- **format_bytes** (*int*) – number of bytes in which to search for the format [default: 50]
- **verbose** (*bool*) – verbose (default) or not

Return str file_format either `AmiraMesh` or `HyperSurface`

ahds.grammar.**get_header**(*fn*, *file_format*, *header_bytes=20000*, *verbose=True*, **args*, ***kwargs*)
Apply rules for detecting the boundary of the header

Parameters

- **fn** (*str*) – file name

- **file_format** (*str*) – either `AmiraMesh` or `HyperSurface`
- **header_bytes** (*int*) – number of bytes in which to search for the header [default: 20000]

Return str data the header as per the `file_format`

`ahds.grammar.`**get_parsed_data**(*fn*, **args*, ***kwargs*)
All above functions as a single function

Parameters **fn** (*str*) – file name

Return tuple(list,int) parsed_data,header_length structured metadata and total number of header bytes

`ahds.grammar.`**parse_header**(*data*, *verbose=False*, **args*, ***kwargs*)
Parse the data using the grammar specified in this module

Parameters **data** (*str*) – delimited data to be parsed for metadata

Return list parsed_data structured metadata

2.3 ahds.header module

2.3.1 header

Module to convert parsed data from an Amira (R) header into a set of nested objects. The key class is *ahds.header.AmiraHeader*.

Usage:

```
>>> from ahds.header import AmiraHeader
>>> ah = AmiraHeader('<somfile>.am')
>>> print(ah)
+-header
↪         AmiraHeader [is_parent? True ]
|  +-filetype: AmiraMesh
|  +-dimension: None
|  +-format: BINARY
|  +-endian: LITTLE
|  +-version: 2.1
|  +-extra_format: None
|  +-Parameters
↪               Block [is_parent? True ]
|  |  +-Materials
↪           ListBlock [is_parent? True ]
|  |  |  +[0]-Exterior
↪               Block [is_parent? False]
|  |  |  |  +-Id: 1
|  |  |  +[1]-Inside
↪               Block [is_parent? False]
|  |  |  |  +-Color: [0.64, 0, 0.8]
|  |  |  |  +-Id: 2
|  |  |  +[2]-Mitochondria
↪               Block [is_parent? False]
|  |  |  |  +-Id: 3
|  |  |  |  +-Color: [0, 1, 0]
|  |  |  +[3]-Mitochondria_
↪               Block [is_parent? False]
```

(continues on next page)

(continued from previous page)

```
|  |  |  |  +-Id: 4
|  |  |  |  +-Color: [1, 1, 1, 0]
|  |  |  +[4]-mitochondria__                                                    ␣
↪               Block [is_parent? False]
|  |  |  |  +-Id: 5
|  |  |  |  +-Color: [0, 0.125, 1]
|  |  |  +[5]-NE                                                                ␣
↪               Block [is_parent? False]
|  |  |  |  +-Id: 6
|  |  |  |  +-Color: [1, 0, 0]
|  |  +-Content: 862x971x200 byte, uniform coordinates
|  |  +-BoundingBox: [0, 13410.7, 0, 15108.4, 1121.45, 4221.01]
|  |  +-CoordType: uniform
|  +-Lattice                                                                     ␣
↪               Block [is_parent? False]
|  |  +-length: [862 971 200]
```

Each nested object is constructed from the :py:class:`Block` class defined in the *core* module.

There are five top-level attributes that every `AmiraHeader` will have:

- *designation*
- *definitions*
- *Parameters*
- *Materials*
- *data_pointers*

All of them, except *Materials* and *Parameters* are marked deprecated and may be removed in future. The *designation* and *defintions* attributes just return the reference to the *AmiraHeader* instance which host all the attributes and data array declarations (*definitions*) found within the *AmiraMesh* and *HyperSurface* files. The *data_pointers* attribute collects from each attribute representing a valid array declaration the defined data definitions and returns the resulting list. New code should directly access the Block objects describing the available data directly from the corresponding array attribute of the *AmiraHeader* object.

The blocks defining the available data structures extend the basic *Block* to the corresponding *AmiraMeshDataStream* and *AmiraHxSurfaceDataStream* blocks which allow to load and access the correspinding data using the *data* property defined by theses blocks. The additional *stream_data*, *encoded_data* and *decoded_data* attributes are deprecated and may be removed in future versions.

The *attrs* attribute of all *Block* and subclass objects including *AmiraHeader* itself which allows to query the attributes added during loading is deprecated and may be removed in future version. To query which attributes are defined on each block use the builtin dir function instead. To access the defined attributes just use them lie any other static attribute.

The following attributes are moved from the *designation* attribute to the basic *header* block:

- *filetype* e.g. `AmiraMesh`, `Avizo` or `HyperSurface`
- *dimension* e.g. `3D`
- *format* e.g. `BINARY-LITTLE-ENDIAN`
- *version* e.g. `2.1`
- *extra_format*

Data pointers are identified by the name `data_pointer_<n>` where <n> is the number indicated by the block attibute in the above examples.

The above approach can also be used to load Amira HyperSurface files.

```
>>> ah = AmiraHeader('<somfile>.surf')
+-header
↪          AmiraHeader [is_parent? False]
|  +-filetype: HyperSurface
|  +-dimension: None
|  +-format: BINARY
|  +-endian: BIG
|  +-version: 0.1
|  +-extra_format: None
|  +-Parameters
↪                Block [is_parent? True ]
|  |  +-Materials
↪             ListBlock [is_parent? True ]
|  |  |  +[0]-Exterior
↪                Block [is_parent? False]
|  |  |  |  +-Id: 1
|  |  |  +[1]-Background
↪                Block [is_parent? False]
|  |  |  |  +-Id: 1
|  |  |  |  +-Color: [1, 0, 0]
|  |  |  +[2]-host_cell_2
↪                Block [is_parent? False]
|  |  |  |  +-Id: 2
|  |  |  |  +-Color: [1, 1, 0.167]
|  |  +-BoundaryIds
↪                Block [is_parent? False]
|  |  |  +-Name: BoundaryConditions
|  |  +-GridBox: [-1, 850, -1, 932, -1, 233]
|  |  +-GridSize: [852, 934, 235]
|  |  +-Filename: /Users/ubanan01/Desktop/cryo_Tomo_Polara/U2OS/08.04....
```

NOTE!!! NEW DESCRIPTION OF THE FILE FORMAT: https://assets.thermofisher.com/TFS-Assets/MSD/Product-Guides/user-guide-amira-software.pdf

class ahds.header.**AmiraHeader**(*fn*, *load_streams=True*, **args*, ***kwargs*)

Bases: ahds.core.Block

Class to encapsulate Amira metadata and accessors to Amira (R) data streams

add_attr(*attr*, *value=None*, *isparent=False*)

Add an attribute to this block object

data_pointers(***kwargs*)

The list of data pointers together with a name, data type, dimension, index, format and length

NOTE: deprecated access the data defnitions for each data array through the corresponding attributes eg.: ah.Nodes.Coordinates instead of ah.data_pointers.data_pointer_1 ah.Tetrahedra.Nodes instead of ah.data_pointers.data_pointer_2 etc.

definitions(***kwargs*)

Definitions consist of a key-value pair specified just after the designation preceded by the key-word 'define'

NOTE: this property is deprecated access the corresponding attributes directly eg. ah.Nodes instead of ah.definitions.Nodes or ah.Tetrahedra instead of ah.defintions.Tetrahedra

designation(***kwargs*)

Designation of the Amira file defined in the first row

Designations consist of some or all of the following data:

- filetype e.g. `AmiraMesh` or `HyperSurface`
- dimensions e.g. `3D`
- format e.g. `BINARY-LITTLE-ENDIAN`
- version e.g. `2.1`
- extra format e.g. `<hxsurface>`

NOTE: this property is deprecated use the corresponding attributes of the AmiraHeader instead to access the above informations

classmethod from_file(***kwargs*)
Deprecated classmethod

load()
Public loading method

move_attr(*new_name*, *name*)
Rename an attribute

2.4 ahds.data_stream module

2.4.1 data_stream

Classes that define data streams (DataList) in Amira (R) files

There are two main types of data streams:

- *AmiraMeshDataStream* is for *AmiraMesh* files
- *AmiraHxSurfaceDataStream* is for *HxSurface* files

Both classes inherit from *AmiraDataStream* class, which handles common functionality such as:

- initialisation with the header metadata
- the *get_data* method calls each subclass's *_decode* method

class ahds.data_stream.**AmiraHxSurfaceDataStream**(*name*, *header*)
Bases: ahds.data_stream.AmiraDataStream

Class that defines an Amira HxSurface data stream

add_attr(*attr*, *value=None*, *isparent=False*)
Add an attribute to this block object

get_data()
Decode and return the stream data in this stream

is_parent
A ListBlock is a parent if it has a Block attribute or if it has list items

load_stream
Reports whether data streams are loaded or not

material_dict
A convenience dictionary of materials indexed by material name

If this is not a Materials ListBlock (name = 'Material') then it should return None

move_attr(*new_name*, *name*)
Rename an attribute

read()
Extract the data streams from the HxSurface file

class ahds.data_stream.**AmiraMeshDataStream**(*name*, *header*)
Bases: ahds.data_stream.AmiraDataStream

Class that defines an AmiraMesh data stream

add_attr(*attr*, *value=None*, *isparent=False*)
Add an attribute to this block object

get_data()
Decode and return the stream data in this stream

is_parent
A ListBlock is a parent if it has a Block attribute or if it has list items

load_stream
Reports whether data streams are loaded or not

material_dict
A convenience dictionary of materials indexed by material name

If this is not a Materials ListBlock (name = 'Material') then it should return None

move_attr(*new_name*, *name*)
Rename an attribute

read()
Extract the data streams from the AmiraMesh file

ahds.data_stream.**byterle_decoder**(*data*, *output_size*)
If the C-ext. failed to compile or is unimportable use this slower Python equivalent

Parameters

- **data** (*str*) – a raw stream of data to be unpacked
- **output_size** (*int*) – the number of items when data is uncompressed

Return np.array output an array of np.uint8

ahds.data_stream.**hxbyterle_decode**(*data*, *output_size*)
If the C-ext. failed to compile or is unimportable use this slower Python equivalent

Parameters

- **data** (*str*) – a raw stream of data to be unpacked
- **output_size** (*int*) – the number of items when data is uncompressed

Return np.array output an array of np.uint8

ahds.data_stream.**hxzip_decode**(*data*, *output_size*)
Decode HxZip data stream

Parameters

- **data** (*str*) – a raw stream of data to be unpacked
- **output_size** (*int*) – the number of items when data is uncompressed

Return np.array output an array of np.uint8

ahds.data_stream.**set_data_stream**(*name*, *header*)
Factory function used by AmiraHeader to determine the type of data stream present

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

ahds, 9
ahds.data_stream, 14
ahds.grammar, 10
ahds.header, 11

Index

A

add_attr() (*ahds.data_stream.AmiraHxSurfaceDataStream method*), 14
add_attr() (*ahds.data_stream.AmiraMeshDataStream method*), 15
add_attr() (*ahds.header.AmiraHeader method*), 13
ahds (*module*), 9
ahds.data_stream (*module*), 14
ahds.grammar (*module*), 10
ahds.header (*module*), 11
AmiraFile (*class in ahds*), 9
AmiraHeader (*class in ahds*), 9
AmiraHeader (*class in ahds.header*), 13
AmiraHxSurfaceDataStream (*class in ahds.data_stream*), 14
AmiraMeshDataStream (*class in ahds.data_stream*), 15

B

byterle_decoder() (*in module ahds.data_stream*), 15

D

data_pointers() (*ahds.AmiraHeader method*), 9
data_pointers() (*ahds.header.AmiraHeader method*), 13
definitions() (*ahds.AmiraHeader method*), 9
definitions() (*ahds.header.AmiraHeader method*), 13
designation() (*ahds.AmiraHeader method*), 10
designation() (*ahds.header.AmiraHeader method*), 13
detect_format() (*in module ahds.grammar*), 10

F

from_file() (*ahds.AmiraHeader class method*), 10
from_file() (*ahds.header.AmiraHeader class method*), 14

G

get_data() (*ahds.data_stream.AmiraHxSurfaceDataStream method*), 14
get_data() (*ahds.data_stream.AmiraMeshDataStream method*), 15
get_header() (*in module ahds.grammar*), 10
get_parsed_data() (*in module ahds.grammar*), 11

H

hxbyterle_decode() (*in module ahds.data_stream*), 15
hxzip_decode() (*in module ahds.data_stream*), 15

I

is_parent (*ahds.data_stream.AmiraHxSurfaceDataStream attribute*), 14
is_parent (*ahds.data_stream.AmiraMeshDataStream attribute*), 15

L

load() (*ahds.AmiraHeader method*), 10
load() (*ahds.header.AmiraHeader method*), 14
load_stream (*ahds.data_stream.AmiraHxSurfaceDataStream attribute*), 14
load_stream (*ahds.data_stream.AmiraMeshDataStream attribute*), 15

M

material_dict (*ahds.data_stream.AmiraHxSurfaceDataStream attribute*), 14
material_dict (*ahds.data_stream.AmiraMeshDataStream attribute*), 15
move_attr() (*ahds.data_stream.AmiraHxSurfaceDataStream method*), 14
move_attr() (*ahds.data_stream.AmiraMeshDataStream method*), 15
move_attr() (*ahds.header.AmiraHeader method*), 14

P

parse_header() (*in module ahds.grammar*), 11

R

read() (*ahds.AmiraFile method*), 9
read() (*ahds.data_stream.AmiraHxSurfaceDataStream method*), 15
read() (*ahds.data_stream.AmiraMeshDataStream method*), 15

S

set_data_stream() (*in module ahds.data_stream*), 15