DDL in BNF for HDF5-1.12 and above

DDL in BNF for HDF5

1. Introduction

This document contains the data description language (DDL) for an HDF5 file. The description is in Backus-Naur Form.

2. Explanation of Symbols

This section contains a brief explanation of the symbols used in the DDL.

```
::=                       defined as
    <tname>                   a token with the name tname
    <a> | <b>                 one of <a> or <b>
    <a>opt                     zero or one occurrence of <a>
    <a>*                      zero or more occurrence of <a>
    <a>+                      one or more occurrence of <a>
    [0-9]                     an element in the range between 0 and 9
    '['                       the token within the quotes (used for special characters)
    TBD                       To Be Decided
```

3. The DDL

```
<file> ::= HDF5 <file_name> { <file_super_block>opt <root_group> }

<file_name> ::= <identifier>

<file_super_block> ::= SUPER_BLOCK {
                           SUPERBLOCK_VERSION <int_value>
                           FREELIST_VERSION <int_value>
                           SYMBOLTABLE_VERSION <int_value>
                           OBJECTHEADER_VERSION <int_value>
                           OFFSET_SIZE <int_value>
                           LENGTH_SIZE <int_value>
                           BTREE_RANK <int_value>
                           BTREE_LEAF <int_value>
                           ISTORE_K <int_value>
                           <super_block_filespace>
                           USER_BLOCK {
                               USERBLOCK_SIZE <int_value>
                           }
                       }

<super_block_filespace> ::= FILE_SPACE_STRATEGY <super_block_strategy>
                            FREE_SPACE_PERSIST <boolean_value>
                            FREE_SPACE_SECTION_THRESHOLD <int_value>
                            FILE_SPACE_PAGE_SIZE <int_value>

<super_block_strategy> ::= H5F_FSPACE_STRATEGY_FSM_AGGR | H5F_FSPACE_STRATEGY_PAGE |
                             H5F_FSPACE_STRATEGY_AGGR | H5F_FSPACE_STRATEGY_NONE |
                             Unknown strategy

<root_group> ::= GROUP "/" {
                     <anon_named_datatype>*
                     <object_id>opt
                     <group_comment>opt
                     <group_attribute>*
                     <group_member>*
                 }

<datatype> ::= <atomic_type> | <compound_type> | <variable_length_type> | <array_type>

<anon_named_datatype> ::= DATATYPE <anon_named_type_name> {
```

```
                            <datatype>
                          }

<anon_named_type_name> ::= the assigned name for anonymous named type is
                           in the form of #oid, where oid is the object id
                           of the type

<atomic_type> ::= <integer> | <float> | <time> | <string> |
                  <bitfield> | <opaque> | <reference> | <enum>

<boolean_value> ::= FALSE | TRUE

<integer> ::= H5T_STD_I8BE     | H5T_STD_I8LE      |
              H5T_STD_I16BE    | H5T_STD_I16LE     |
              H5T_STD_I32BE    | H5T_STD_I32LE     |
              H5T_STD_I64BE    | H5T_STD_I64LE     |
              H5T_STD_U8BE     | H5T_STD_U8LE      |
              H5T_STD_U16BE    | H5T_STD_U16LE     |
              H5T_STD_U32BE    | H5T_STD_U32LE     |
              H5T_STD_U64BE    | H5T_STD_U64LE     |
              H5T_NATIVE_CHAR  | H5T_NATIVE_UCHAR  |
              H5T_NATIVE_SHORT | H5T_NATIVE_USHORT |
              H5T_NATIVE_INT   | H5T_NATIVE_UINT   |
              H5T_NATIVE_LONG  | H5T_NATIVE_ULONG  |
              H5T_NATIVE_LLONG | H5T_NATIVE_ULLONG

<float> ::= H5T_IEEE_F32BE   | H5T_IEEE_F32LE     |
            H5T_IEEE_F64BE   | H5T_IEEE_F64LE     |
            H5T_NATIVE_FLOAT |  H5T_NATIVE_DOUBLE |
            H5T_NATIVE_LDOUBLE

<time> ::= H5T_TIME: not yet implemented

<string> ::= H5T_STRING {
                 STRSIZE <strsize>;
                 STRPAD <strpad>;
                 CSET <cset>;
                 CTYPE <ctype>;
             }

<strsize> ::= <int_value>

<strpad> ::= H5T_STR_NULLTERM | H5T_STR_NULLPAD | H5T_STR_SPACEPAD

<cset> ::= H5T_CSET_ASCII | H5T_CSET_UTF8

<ctype> ::= H5T_C_S1 | H5T_FORTRAN_S1

<bitfield> ::= H5T_STD_B8BE     | H5T_STD_B8LE      |
               H5T_STD_B16BE    | H5T_STD_B16LE     |
               H5T_STD_B32BE    | H5T_STD_B32LE     |
               H5T_STD_B64BE    | H5T_STD_B64LE

<opaque> ::= H5T_OPAQUE {
                 OPAQUE_TAG <identifier>;
                 OPAQUE_SIZE <int_value>;opt
             }

<reference> ::= H5T_REFERENCE { <ref_type> }

<ref_type> ::= H5T_STD_REF_OBJECT | H5T_STD_REF_DSETREG | H5T_STD_REF | UNDEFINED

<compound_type> ::= H5T_COMPOUND {
                        <member_type_def>+
                    }

<member_type_def> ::= <datatype> <field_name>;

<field_name> ::= <identifier>

<variable_length_type> ::= H5T_VLEN { <datatype> }
```

```
<array_type> ::= H5T_ARRAY { <dim_sizes> <datatype> }

<dim_sizes> ::= '['<dimsize>']' | '['<dimsize>']'<dim_sizes>

<dimsize> ::= <int_value>

<attribute> ::= ATTRIBUTE <attr_name> {
                    <dataset_type>
                    <dataset_space>
                    <data>opt
                }

<attr_name> ::= <identifier>

<dataset_type> ::= DATATYPE <path_name> | <datatype>

<enum> ::= H5T_ENUM {
               <enum_base_type> <enum_def>+
           }

<enum_base_type> ::= <integer>
// Currently enums can only hold integer type data, but they may be expanded
// in the future to hold any datatype

<enum_def> ::= <enum_symbol> <enum_val>;

<enum_symbol> ::= <identifier>

<enum_val> ::= <int_value>

<path_name> ::= <path_part>+

<path_part> ::= /<identifier>

<dataspace> ::= <scalar_space> | <simple_space> | <complex_space> | <null_space>

<null_space> ::= NULL

<scalar_space> ::= SCALAR

<simple_space> ::= SIMPLE { <current_dims> / <max_dims> }

<complex_space> ::= COMPLEX { <complex_space_definition> }

<dataset_space> ::= DATASPACE <path_name> | <dataspace>

<current_dims> ::= <dims>

<max_dims> ::= '(' <max_dim_list> ')'

<max_dim_list> ::= <max_dim> | <max_dim>, <max_dim_list>

<max_dim> ::= <int_value> | H5S_UNLIMITED

<data> ::= <subset> | <data_values>

<data_values> ::= DATA {
               <scalar_space_data> | <simple_space_data>
           }

<scalar_space_data> ::= <any_element>

<any_element> ::= <atomic_element> | <compound_element> |
                  <variable_length_element> | <array_element>

<any_data_seq> ::= <any_element> | <any_element>, <any_data_seq>

<atomic_element> :: = <integer_data> | <float_data>    | <time_data>   |
                      <string_data>  | <bitfield_data> | <opaque_data> |
                      <enum_data>    | <reference_data>

<subset> ::= SUBSET {
```

```
                <start>;
                <stride>;
                <count>;
                <block>;
                DATA {
                    <simple_space_data>
                }
            }

<start> ::= START (<coor_list>)

<stride> ::= STRIDE (<pos_list>)

<count> ::= COUNT (<max_dim_list>)

<block> ::= BLOCK (<max_dim_list>)

<coor_list> ::= <coor_data>, <coor_list> | <coor_data>

<coor_data> ::= <integer_data> | H5S_UNLIMITED

<integer_data> ::= <int_value>

<float_data> ::= a floating point number

<time_data> ::= DATA{ not yet implemented.}

<string_data> ::= a string
// A string is enclosed in double quotes.
// If a string is displayed on more than one line, string concatenate
// operator '//'is used.

<bitfield_data> ::= <hex_value>

<opaque_data> ::= <hex_value>:<hex_value> | <hex_value>

<enum_data> ::= <enum_symbol>

<reference_data> ::= <object_ref_data> | <data_region_data> | <attribute_data> | NULL

<object_ref_data> ::= <object_type> <object_ref>

<object_type> ::= ATTRIBUTE | DATASET | GROUP | DATATYPE

<object_ref> ::= <object_id>

<object_id> ::= <path_name> | OBJECTID { <object_num> }

<object_num> ::= <int_value>:<int_value> | <int_value>

<attribute_data> ::= ATTRIBUTE <attr_name>opt
                          <data>opt

<data_region_data> ::= DATASET <dataset_name> {
                          <data_region_type>opt <data_region_data_list>
                          <dataset_type>opt <dataset_space>opt
                          <data>opt
                       }

<data_region_type> ::= REGION_TYPE <data_region_data_type>

<data_region_data_type> ::= POINT | BLOCK

<data_region_data_list> ::= <data_region_data_info>, <data_region_data_list> |
                            <data_region_data_info>

<data_region_data_info> ::= <region_info> | <point_info>

<region_info> ::= (<lower_region_vals>)-(<upper_region_vals>)

<lower_region_vals> ::= <lower_bound>, <lower_region_vals> | <lower_bound>
```

```
<upper_region_vals> ::= <upper_bound>, <upper_region_vals> | <upper_bound>

<lower_bound> ::= <int_value>

<upper_bound> ::= <int_value>

<point_info> ::= (<point_vals>)

<point_vals> ::= <int_value> | <int_value>, <point_vals>

<compound_element> ::= { <any_data_seq> }

<atomic_simple_data> :: = <atomic_element>, <atomic_simple_data> |
                          <atomic_element>

<simple_space_data> :: = <any_data_seq>

<variable_length_element> ::= ( <any_data_seq> )

<array_element> ::= '[' <any_data_seq> ']'

<named_datatype> ::= DATATYPE <type_name> { <datatype> }

<type_name> ::= <identifier>

<hardlink> ::= HARDLINK <path_name>

<group> ::= GROUP <group_name> { <hardlink> | <group_info> }

<group_comment> ::= COMMENT <string_data>

<group_name> ::= <identifier>

<group_info> ::= <object_id>opt <group_comment>opt <group_attribute>*
                 <group_member>*

<group_attribute> ::= <attribute>

<group_member> ::= <named_datatype> | <group> | <dataset> |
                   <softlink> | <external_link>

<dataset> ::= DATASET <dataset_name> { <hardlink> | <dataset_info> }

<dataset_info> ::= <dataset_type>
                   <dataset_space>
                   <dcpl_info>opt
                   <dataset_attribute>* <object_id>opt
                   <data>opt
// Tokens above can be in any order as long as <data> is
// after <dataset_type> and <dataset_space>.

<dcpl_info> ::= <storagelayout>
                <compression_filters>
                <fillvalue>
                <allocationtime>

<dataset_name> ::= <identifier>

<storagelayout> :: =  STORAGE_LAYOUT {
                         <contiguous_layout> | <chunked_layout> |
                         <compact_layout> | <virtual_layout>
                      }

<contiguous_layout> ::= CONTIGUOUS
                        <internal_layout> | <external_layout>

<chunked_layout> ::= CHUNKED <dims>
                     <filter_ratio>opt

<compact_layout> ::= COMPACT
                     <size>
```

```
<internal_layout> ::= <size>
                      <offset>

<external_layout> ::= EXTERNAL {
                          <external_file>+
                      }

<virtual_layout> ::= <vmaps>*opt

<vmaps> ::= MAPPING <int_value> {
                <virtual_map>
                <source_map>
            }

<virtual_map> ::= VIRTUAL {
                      <vmaps_selection>
                  }

<source_map> ::= SOURCE {
                     FILE <file_name>
                     DATASET <dataset_name>
                     <vmaps_selection>
                 }

<vmaps_selection> ::= <regular_hyperslab> | <irregular_hyperslab> |
                      <select_points> | <select_none> | <select_all>

<regular_hyperslab> ::= SELECTION REGULAR_HYPERSLAB {
                            <start>
                            <stride>
                            <count>
                            <block>
                        }

<irregular_hyperslab> ::= SELECTION IRREGULAR_HYPERSLAB {
                              <region_info>+
                            }

<select_points> ::= SELECTION POINT {
                        (<coor_list>)+
                    }

<select_none> ::= SELECTION NONE

<select_all> ::= SELECTION ALL

<dims> ::= (<dims_values>)

<dims_values> ::= <int_value> | <int_value>, <dims_values>

<external_file> ::= FILENAME <file_name> <size> <offset>

<offset> ::= OFFSET <int_value>

<size> ::= SIZE <int_value>

<filter_ratio> ::= <size> | <compressionratio>

<compressionratio> :: = <size> (<float_data>:1 COMPRESSION)

<compression_filters> :: = FILTERS {
                               <filter_type>+ | NONE
                           }

<filter_type> :: = <filter_deflate>   | <filter_shuffle>     |
                   <filter_flecther> | <filter_szip>        |
                   <filter_nbit>     | <filter_scaleoffset> |
                   <filter_default>

<filter_default> :: = <filter_user> {
                          FILTER_ID <int_value>
                          <filter_comment>opt
```

```
                        <filter_params>opt
                    }

<filter_user> :: = USER_DEFINED_FILTER

<filter_deflate> :: = COMPRESSION DEFLATE { LEVEL <int_value> }

<filter_shuffle> :: = PREPROCESSING SHUFFLE

<filter_flecther> :: = CHECKSUM FLETCHER32

<filter_szip> :: = COMPRESSION SZIP {
                       PIXELS_PER_BLOCK <int_value>
                       <filter_szip_mode>opt
                       <filter_szip_coding>opt
                       <filter_szip_order>opt
                       <filter_szip_header>opt
                  }

<filter_szip_mode> :: = MODE HARDWARE | K13

<filter_szip_coding> :: = CODING ENTROPY | NEAREST NEIGHBOUR

<filter_szip_order> :: = BYTE_ORDER LSB | MSB

<filter_szip_header> :: = HEADER RAW

<filter_nbit> :: = CHECKSUM NBIT

<filter_scaleoffset> :: = COMPRESSION SCALEOFFSET { MIN BITS <int_value> }

<filter_comment> :: = COMMENT  <identifier>

<filter_params> :: = PARAMS { <int_value>* }

<fillvalue> ::= FILLVALUE {
                    FILL_TIME H5D_FILL_TIME_ALLOC | H5D_FILL_TIME_NEVER | H5D_FILL_TIME_IFSET
                    VALUE H5D_FILL_VALUE_UNDEFINED | H5D_FILL_VALUE_DEFAULT | <any_element>
                }

<allocationtime> ::= ALLOCATION_TIME {
                         H5D_ALLOC_TIME_EARLY | H5D_ALLOC_TIME_INCR |
                         H5D_ALLOC_TIME_LATE
                     }

<dataset_attribute> ::= <attribute>

<softlink> ::= SOFTLINK <softlink_name> {
                   LINKTARGET <target>
               }

<softlink_name> ::= <identifier>

<target> ::= <identifier>

<external_link> ::= EXTERNAL_LINK <external_link_name> {
                        TARGETFILE <targetfile>
                        TARGETPATH <targetpath> <targetobj>opt
                    }

<external_link_name> ::= <identifier>

<user_defined_link> ::= USERDEFINED_LINK <external_link_name> {
                            LINKCLASS <user_link_type>
                        }

<user_link_type> ::= <int_value>

<targetfile> ::= <file_name>

<targetpath> ::= <identifier>
```

```
<targetobj> ::= <named_datatype> | <group> | <dataset>

<identifier> ::= "a string"
// character '/' should be used with care.

<pos_list> ::= <pos_int>, <pos_list> | <pos_int>

<int_value> ::= 0 | <pos_int>

<pos_int> ::= [1-9][0-9]*

<hex_value> ::= 0x[0-F][0-F]+ | [0-F][0-F]+
```

4. An Example of an HDF5 File in DDL

```
HDF5 "example.h5" {
GROUP "/" {
   ATTRIBUTE "attr1" {
      DATATYPE H5T_STRING {
           STRSIZE 17;
           STRPAD H5T_STR_NULLTERM;
           CSET H5T_CSET_ASCII;
           CTYPE H5T_C_S1;
         }
      DATASPACE SCALAR
      DATA {
         "string attribute"
      }
   }
   DATASET "dset1" {
      DATATYPE H5T_STD_I32BE
      DATASPACE SIMPLE { ( 10, 10 ) / ( 10, 10 ) }
      DATA {
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
         0, 1, 2, 3, 4, 5, 6, 7, 8, 9
      }
   }
   DATASET "dset2" {
      DATATYPE H5T_COMPOUND {
         H5T_STD_I32BE "a";
         H5T_IEEE_F32BE "b";
         H5T_IEEE_F64BE "c";
      }
      DATASPACE SIMPLE { ( 5 ) / ( 5 ) }
      DATA {
         {
            1,
            0.1,
            0.01
         },
         {
            2,
            0.2,
            0.02
         },
         {
            3,
            0.3,
            0.03
         },
         {
```

```
         4,
         0.4,
         0.04
      },
      {
         5,
         0.5,
         0.05
      }
   }
}
GROUP "group1" {
   COMMENT "This is a comment for group1";
   DATASET "dset3" {
      DATATYPE "/type1"
      DATASPACE SIMPLE { ( 5 ) / ( 5 ) }
      DATA {
         {
            [ 0, 1, 2, 3 ],
            [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
              0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
              0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
              0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
              0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
         },
         {
            [ 0, 1, 2, 3 ],
            [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
              0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
              0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
              0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
              0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
         },
         {
            [ 0, 1, 2, 3 ],
            [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
              0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
              0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
              0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
              0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
         },
         {
            [ 0, 1, 2, 3 ],
            [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
              0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
              0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
              0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
              0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
         },
         {
            [ 0, 1, 2, 3 ],
            [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
              0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
              0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
              0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
              0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
         }
      }
   }
}
DATASET "dset3" {
   DATATYPE H5T_VLEN { H5T_STD_I32LE }
   DATASPACE SIMPLE { ( 4 ) / ( 4 ) }
   DATA {
      (0), (10, 11), (20, 21, 22), (30, 31, 32, 33)
   }
}
GROUP "group2" {
   HARDLINK "/group1"
}
SOFTLINK "slink1" {
   LINKTARGET "somevalue"
```

```
   }
   DATATYPE "type1" H5T_COMPOUND {
      H5T_ARRAY { [4] H5T_STD_I32BE } "a";
      H5T_ARRAY { [5][6] H5T_IEEE_F32BE } "b";
   }
}
}
```