

Error Handling Interface

HDF-EOS Workshop IX

Quincey Koziol and Ray Lu

30 Nov 2005

Contents

- I. Basic concepts
- II. Basic error operations
- III. Advanced error operations

Basic Concepts

An example:

HDF5-DIAG: Error detected in HDF5 (1.7.52) thread 0:

```
#000: ../../hdf5_1.7/src/H5D.c line 1175 in H5Dcreate(): not a location ID
```

```
major: Invalid arguments to routine
```

```
minor: Inappropriate type
```

```
#001: ../../hdf5_1.7/src/H5Gloc.c line 162 in H5G_loc(): invalid object ID
```

```
major: Invalid arguments to routine
```

```
minor: Bad value
```

Library's error report:

- automatic
- reset for each API function call.

Concepts:

- Error stack
- Error record
- Error message: major and minor
- Error description

Basic Error Operations

1. To print error stack explicitly:

```
H5Eprint_stack(hid_t error_stack, FILE* stream)
```

2. To clear error stack:

```
H5Eclear_stack(hid_t error_stack)
```

3. To disable error stack printing:

```
H5Eset_auto_stack(hid_t error_stack,  
                 H5E_auto_stack_t func, void *client_data)
```

```
H5Eget_auto_stack(hid_t error_stack,  
                 H5E_auto_stack_t *func, void *client_data)
```

```
typedef herr_t (*H5E_auto_stack_t)(hid_t estack,  
                                   void *client_data)
```

Basic Error Operations

Example: Turn off error printing while “probing” a function.

```
/* Save old error handler */
H5E_auto_stack_t old_func;
void              *old_client_data;

H5Eget_auto_stack(error_stack, &old_func, &old_client_data);

/* Turn off error handling */
H5Eset_auto_stack(error_stack, NULL, NULL);

/* Probe. Likely to fail, but that's okay */
status = H5Fopen(.....);
      :

/* Restore previous error handler */
H5Eset_auto_stack(error_stack, old_func, old_client_data);
```

Another example: the library provides a cleaner way to do the same.

```
H5E_BEGIN_TRY {
    status = H5Fopen(.....);
} H5E_END_TRY;
```

Basic Error Operations

4. To customize printing of error stack:

Example:

```
herr_t my_hdf5_error_print(hid_t err_stack, void *unused){
    fprintf (stderr, "An HDF5 error was detected. Bye.\n");
    exit (1);
}
```

The function is installed as the error handler by saying

```
H5Eset_auto_stack(H5E_DEFAULT, my_hdf5_error_handler, NULL);
```

Basic Error Operations

5. To walk through error stack:

```
H5Ewalk_stack(hid_t err_stack, H5E_direction_t dir,  
             H5E_walk_t func, void *client_data)
```

```
typedef herr_t (*H5E_walk_t)(unsigned n, H5E_error_t *eptr,  
                             void *client_data)
```

```
typedef struct {  
    hid_t    cls_id;  
    hid_t    maj_num;  
    hid_t    min_num;  
    const char *func_name;  
    const char *file_name;  
    unsigned  line;  
    const char *desc;  
} H5E_error_t;
```

Advanced Error Operations

A new feature to allow application to push its own error records onto the HDF5 error stack.

Example:

```
Error Test-DIAG: Error detected in Error Program (1.0) thread 0:
#000: ../../hdf5_1.7/test/error_example.c line 164 in advanced_operations():
  H5Dcreate failed
  major: Error in API
  minor: Error in H5Dcreate
HDF5-DIAG: Error detected in HDF5 (1.7.52) thread 0:
#001: ../../hdf5_1.7/src/H5D.c line 1175 in H5Dcreate(): not a location ID
  major: Invalid arguments to routine
  minor: Inappropriate type
#002: ../../hdf5_1.7/src/H5Gloc.c line 162 in H5G_loc(): invalid object ID
  major: Invalid arguments to routine
  minor: Bad value
```

More concepts:

- Error class
- Library name
- Library version

Advanced Error Operations

1. To register application with error API:

```
H5Eregister_class(const char* cls_name, const char*  
                 lib_name, const char* version)
```

```
H5Ecreate_msg(hid_t class, H5E_type_t msg_type,  
             const char* msg)
```

```
H5Eget_class_name(hid_t class_id, char* name,  
                 size_t size)
```

```
H5Eget_msg(hid_t msg_id, H5E_type_t* msg_type,  
          char* msg, size_t size)
```

```
H5Eclose_msg(hid_t msg_id)
```

```
H5Eunregister_class(hid_t class_id)
```

Advance Error Operations

Example:

An application creates an error class and error messages.

```
/* Create an error class */
class_id = H5Eregister_class(ERR_CLS_NAME, PROG_NAME, PROG_VERS);

/* Create a major error message in the class */
maj_id = H5Ecreate_msg(class_id, H5E_MAJOR, "... ..");

/* Create a minor error message in the class */
min_id = H5Ecreate_msg(class_id, H5E_MINOR, "... ..");
```

Application closes error messages and un-registers error class.

```
H5Eclose_msg(maj_id);
H5Eclose_msg(min_id);
H5Eunregister_class(class_id);
```

Advanced Error Operations

2. To manipulate error stack:

H5Eget_current_stack(*void*)

H5Eset_current_stack(*hid_t* error_stack)

H5Epush_stack(*hid_t* error_stack, *const char** file,
*const char** func, *unsigned* line, *hid_t* cls_id,
hid_t major_id, *hid_t* minor_id, *const char**
desc, ...)

H5Epop(*hid_t* error_stack, *size_t* count)

H5Eget_num(*hid_t* error_stack)

H5Eclear_stack(*hid_t* error_stack)

H5Eclose_stack(*hid_t* error_stack)

Advanced Error Operations

Example:

```
/* Make call to HDF5 I/O routine */
if((dataset = H5Dcreate(FAKE_ID, DSET_NAME, H5T_STD_I32BE,
                      space, H5P_DEFAULT))<0)
{
    /* Push client error onto error stack */
    H5Epush_stack(H5E_DEFAULT, __FILE__, FUNC_adv, __LINE__,
                 ERR_CLS, ERR_MAJ_API, ERR_MIN_CREATE,
                 "H5Dcreate failed");

    /* Print the error stack explicitly */
    H5Eprint_stack(H5E_DEFAULT, stderr);
}
```

Advanced Error Operations

Another example:

```
H5E_BEGIN_TRY {
    ret = H5Dwrite(dset_id, mem_type_id, mem_space_id, file_space_id,
                  dset_xfer_plist_id, buf);
} H5E_END_TRY;

if(ret<0) {
    /* Push client error onto error stack */
    H5Epush_stack(H5E_DEFAULT,FILE,FUNC,LINE,cls_id,
                  CLIENT_ERR_MAJ_IO, CLIENT_ERR_MINOR_HDF5,"H5Dwrite failed");

    /* Preserve the error stack by assigning an object handle to it */
    error_stack = H5Eget_current_stack();

    /* Close dataset */
    H5Dclose(dset_id);

    /* Replace the current error stack with the preserved one */
    H5Eset_current_stack(error_stack);
    Return(1);
}
```

Contents

- I. Basic concepts
- II. Basic error operations
- III. Advanced error operations

Documents

User's Guide: http://hdf.ncsa.uiuc.edu/HDF5/doc_dev_snapshot/HE9-ErrorAPI,DataConversion/ErrorHandling-H5E-July05.pdf

Reference Manual:

http://hdf.ncsa.uiuc.edu/HDF5/doc_dev_snapshot/H5_dev/RM/RM_H5E.html

Acknowledgement:

This presentation is based upon work supported in part by a Cooperative Agreement with the National Aeronautics and Space Administration (NASA) under NASA grant NNG05GC60A. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NASA. Other support provided by NCSA and other sponsors and agencies (<http://hdf.ncsa.uiuc.edu/acknowledge.html>).