H5TB_APPEND_RECORDS

Expand all Collapse all

- Jump to ...
- Summary
- Description
- Example
- Switch language ...
- C
- C++
- FORTRAN
- JAVA

H5TB_APPEND_RECORDS

Adds records to the end of the table

Procedure:

H5TB_APPEND_RECORDS (loc_id, dset_name, nrecords, type_size, field_offset, field_sizes, data)

Signature:

```
herr_t H5TBappend_records ( hid_t loc_id, const char *dset_name, hsize_t nrecords, size_t type_size, const
size_t*field_offset,
                const size_t *field_sizes, const void *data )
```

Parameters:

hid_t `loc_id`	IN: Identifier of the file or group where the table is located
const char `*dset_name`	IN: The name of the dataset to overwrite
hsize_t `nrecords`	IN: The number of records to append
size_t `type_size`	IN: The size of the structure type, as calculated by sizeof()
const size_t `*field_offset`	IN: An array containing the offsets of the fields. These offsets can be calculated with the HOFFSET macro
const size_t `*field_sizes`	IN: An array containing the sizes of the fields
const void `*data`	IN: Buffer with data

Description:

H5TB_APPEND_RECORDS adds records to the end of the table named `dset_name` attached to the object specified by the identifier `loc_id`. The dataset is extended to hold the new records.

Returns:

Returns a non-negative value if successful; otherwise returns a negative value.

Example:

hl / examples / ex_table_02.c [56:106] 1.10/master HDFFV/hdf5

```
/* Calculate the size and the offsets of our struct members in memory */
 size_t dst_size =  sizeof( Particle );
 size_t dst_offset[NFIELDS] = { HOFFSET( Particle, name ),
                                HOFFSET( Particle, lati ),
                                HOFFSET( Particle, longi ),
                                HOFFSET( Particle, pressure ),
                                HOFFSET( Particle, temperature )};

 size_t dst_sizes[NFIELDS] = { sizeof( p_data[0].name),
                               sizeof( p_data[0].lati),
                               sizeof( p_data[0].longi),
                               sizeof( p_data[0].pressure),
                               sizeof( p_data[0].temperature)};

 /* Define field information */
 const char *field_names[NFIELDS] =
 { "Name","Latitude", "Longitude", "Pressure", "Temperature" };
 hid_t      field_type[NFIELDS];
 hid_t      string_type;
 hid_t      file_id;
 hsize_t    chunk_size = 10;
 int        *fill_data = NULL;
 int        compress  = 0;
 int        i;

  /* Append particles */
 Particle particle_in[ NRECORDS_ADD ] =
 {{ "eight",80,80, 8.0f, 80.0},
 {"nine",90,90, 9.0f, 90.0} };

 /* Initialize the field field_type */
 string_type = H5Tcopy( H5T_C_S1 );
 H5Tset_size( string_type, 16 );
 field_type[0] = string_type;
 field_type[1] = H5T_NATIVE_INT;
 field_type[2] = H5T_NATIVE_INT;
 field_type[3] = H5T_NATIVE_FLOAT;
 field_type[4] = H5T_NATIVE_DOUBLE;

 /* Create a new file using default properties. */
 file_id = H5Fcreate( "ex_table_02.h5", H5F_ACC_TRUNC, H5P_DEFAULT, H5P_DEFAULT );

 /* make a table */
 H5TBmake_table( "Table Title",file_id,TABLE_NAME,NFIELDS,NRECORDS,
                       dst_size, field_names, dst_offset, field_type,
                       chunk_size, fill_data, compress, p_data  );

 /* append two records */
 H5TBappend_records(file_id, TABLE_NAME,NRECORDS_ADD, dst_size, dst_offset, dst_sizes,
  &particle_in );
```