```
   HDF5CloseGroup groupID
   HDF5CloseFile fileID
End
```

Partial paths are relative to the file ID or group ID passed to HDF5SaveData.

Saving HDF5 Dataset Region Reference Data

Igor Pro does not currently support saving dataset region references.

HDF5 Enum Data

Most HDF5 files do not use enum datatypes so most users do not need to know this information.

Enum data values are stored in an HDF5 file as integers. The datatype associated with an enum dataset or attribute defines a mapping from an integer value to a name. HDF5LoadData can load either the integer data or the name for each element of enum data. You control this using the /ENUM=enumMode flag.

If /ENUM is omitted or enumMode is zero, HDF5LoadData creates a numeric wave with data type signed long and loads the integer enum values into it.

If enumMode is 1, HDF5LoadData creates a text wave and loads the name associated with each enum value into it. This is slower than loading the integer enum values but the speed penalty is significant only if you are loading a very large enum dataset or very many enum datasets.

HDF5 Opaque Data

Most HDF5 files do not use opaque datatypes so most users do not need to know this information.

Opaque data consists of elements that are treated as a string of bytes of a specified length. HDF5LoadData loads opaque data into an unsigned byte wave. If an element of opaque data is n bytes then each element occupies n contiguous rows in the wave.

HDF5 Bitfield Data

Most HDF5 files do not use bitfield datatypes so most users do not need to know this information.

Bitfield data consists of elements that are a sequence of bits treated as individual values. HDF5LoadData loads bitfield data of any length into an unsigned byte wave with as many bytes per element as are needed. For example, if the bitfield is two bytes then two rows of the unsigned byte wave are used for each element.

The data is loaded using the byte order as stored in the file. This is appropriate if you think of the data as a stream of bytes. If you think of the data as a short (2 bytes) or long (4 bytes), you can use the Redimension command. For example, if you just loaded a 2D data set containing 5 two-byte bitfields per row and 3 columns, you wind up with a 10x3 unsigned byte wave. You can change it to a 5x3 wave of shorts like this:

```
Redimension /N=(5,3) /W /E=1 bitfieldWave
```

If you need to change the byte order, use /E=2 instead of /E=1.

HDF5 Nested Datatypes

Most HDF5 files do not use nested datatypes so most users do not need to know the following information.

HDF5 supports "atomic" data classes, such as integers (called class H5T_INTEGER in the HDF5 library) and floats (class H5T_FLOAT), and "composite" data classes, such as structures (class H5T_COMPOUND), arrays (class H5T_ARRAY) and variable-length (class H5T_VLEN) types.

In a dataset whose datatype is atomic, each element of the dataset is a single value. In a dataset whose datatype is composite, each element is a collection of elements of one or more other datatypes.