

HDF5LibraryInfo	Returns information about the HDF5 library used by Igor. This is of interest to advanced programmers only.
HDF5Control	Provides control of aspects of Igor's use of the HDF5 file format.
HDF5Dump	Returns a DDL-format dump of a group, dataset or attribute.
HDF5DumpErrors	Returns information about HDF5-related errors encountered by Igor. This is a diagnostic tool for experts that is needed only in rare cases.

HDF5 Procedure Files

Igor ships with two procedure files to support HDF5 use and programming. Both files are automatically loaded by Igor on launch and consequently are always available.

"HDF5 Browser.ipf" implements the HDF5 Browser. This procedure file is an independent module and consequently is normally hidden. If you are an Igor programmer who wants to inspect the procedure file, see Independent Modules for background information. However, there is no reason for you to call routines in "HDF5 Browser.ipf" from your own code.

"HDF5 Utilities.ipf" is a public procedure file (i.e., not an independent module) that defines HDF5-related constants and provides HDF5-related utility routines that may be of use if you write procedures that use HDF5 features.

If you write your own procedure file, you can use the constants and utility routines in "HDF5 Utilities.ipf" without #including anything. However, if you are creating your own independent module for HDF5 programming, you will need to #include "HDF5 Utilities.ipf" into your independent module - see "HDF5 Browser.ipf" for an example.

HDF5 Attributes

An attribute is a piece of data attached to an HDF5 group, dataset or named datatype.

To load an attribute, you need to use the HDF5LoadData operation with the /A=attributeNameStr flag and the /TYPE=objectType flag.

Loading attributes of type H5T_COMPOUND (compound - i.e., structure) is not supported.

Loading an HDF5 Numeric Attribute

This function illustrates how to load a numeric attribute of a group or dataset. The function result is an error code. The value of the attribute is returned via the pass-by-reference attributeValue numeric parameter.

```
Function LoadHDF5NumericAttribute(pathName, filePath, groupPath, objectName,  
                                objectType, attributeName, attributeValue)  
    String pathName           // Symbolic path name - or ""  
    String filePath           // File name, relative path or full path  
    String groupPath          // Path to group, such as "/", "/my_group"  
    String objectName         // Name of group or dataset  
    Variable objectType       // 1=group, 2=dataset  
    String attributeName      // Name of attribute  
    Variable& attributeValue  // Output - pass-by-reference parameter  
  
    attributeValue = NaN  
  
    Variable result = 0  
  
    // Open the HDF5 file  
    Variable fileID           // HDF5 file ID will be stored here  
    HDF5OpenFile /P=$pathName /R /Z fileID as filePath
```