

# Installing and Using HDF Java Products

1	Introduction.....	1
2	Installing the HDF Java Products .....	2
2.1	Release Files .....	2
2.2	Installation.....	2
3	Java HDF5 Interface (JHI5).....	3
3.1	Description .....	3
3.2	Use.....	4
3.3	Considerations .....	7
4	HDF Object Package .....	7
4.1	Description .....	7
4.2	Use.....	8
5	HDFView .....	10
5.1	Description .....	10
5.2	Use.....	11
5.3	Windows .....	11
5.4	Unix .....	11
5.5	Mac.....	11

---

## 1 Introduction

This document provides basic instructions on the installation and use of HDF Java Products. More detailed information on the individual products is accessible via the links on the left side of the HDF Java Products web page: <http://www.hdfgroup.org/hdf-java.html>. HDF5 documentation can be found at <http://www.hdfgroup.org/HDF5/doc/doc-info.html>.

The HDF Java Products support both HDF4 and HDF5 formats. In this document, instructions and examples focus on HDF5.

The HDF Java Products include three distinct components:

- Java HDF Interface (JHI4 & JHI5): a Java Native Interface to the standard HDF (4&5) libraries
- Java HDF Object Package: a Java package that implements HDF data objects in an object-oriented form
- HDFView: a visual tool for browsing and editing HDF files

## 2 Installing HDF Java Products

### 2.1 Release Files

HDF Java Products are available for download from <http://www.hdfgroup.org/ftp/HDF5/hdf-java/>.

The HDF Java Products release includes prebuilt binaries and source code. Figure 1 shows the release files for the Windows 64-bit platform. Similar directories and files exist for other platforms.

```
COPYING
bin/
  COPYING
  win64/
    hdf-java-x.x-bin.tar
    hdf-java/
hdfview/
  COPYING
  hdfview_install_win64.exe
src/
  COPYING
  build_src_windows.html
  hdf-java-x.x-src.tar
  hdf-java/
```

**Figure 1. HDF Java Products release files**

In Figure 1, “x.x” is the version number of the HDF Java products (for example, if the version number were 2.6, then x.x would be 2.6 and read “hdf-java-2.6-bin.tar”). The “bin/win64” directory includes the prebuilt binaries for all of the HDF Java products for the Windows 64-bit platform. Although you can download specific binaries from the directory “bin/win64/hdf-java”, we encourage you to download and use the “bin/win64/hdf-java-x.x-bin.tar” file.

The “hdfview/” directory contains the HDFView installation program for the Windows 64-bit platform. This installation program also installs the HDF Object Package and the Java HDF5 Interface.

The “src/” directory has all of the source code for the Java HDF5 Interface, the Java HDF Object Package, and HDFView.

### 2.2 Installation

You can install the HDF Java Products from the HDFView installer or from the prebuilt binaries.

To install from the HDFView installer:

- 1) Download the installer for your platform, for example, hdfview\_install\_win64.exe, from the “hdfview/” directory for the Windows 64-bit platform.
- 2) Run the installer. The installer will guide you through the installation process.
- 3) The HDFView home directory will be the home directory for all of the HDF-Java products.

To install from the prebuilt binaries:

- 1) Download hdf-java-x.x-bin.tar from the appropriate directory for your platform. This will be the “bin/win64” directory for the Windows 64-bit platform.

- 2) Extract the files from the tar archive. The “hdf-java/” directory will be the home directory for all of the HDF-Java products.
- 3) If you plan to use HDFView, set the necessary variables in the hdfview.bat or hdfview.sh file for your installation:
  - For Windows:
    - a) Edit the file “hdf-java/bin/hdfview.bat”.
    - b) Fill in the values for the JAVAHOME and HDFJAVA variables for your installation. You would, for example, fill in the text shown in blue:  
SET JAVAHOME=C:\java\jdk1.6.0  
SET HDFJAVA=C:\hdf-java
  - For other platforms:
    - a) Edit the file “hdf-java/bin/hdfview.sh”.
    - b) Fill in the values for the JAVAHOME and HDFJAVA variables for your installation. You would, for example, fill in the text shown in blue:  
SET JAVAHOME=C:\java\jdk1.6.0  
SET HDFJAVA=C:\hdf-java

## 3 Java HDF5 Interface (JHI5)

### 3.1 Description

The Java HDF5 Interface is a Java package that “wraps” the HDF5 C library using the Java Native Method Interface (JNI). JHI5 provides Java developers with the greatest degree of flexibility and control in writing and reading HDF5 files.

It is extremely important to emphasize that the JHI5 package is not a pure Java implementation of the HDF5 library. The JHI5 package calls the same HDF5 library that is used by C programs to read and write HDF5 files. Using JHI5, Java applications are able to make calls that are implemented in the HDF5 C library. For details, visit <http://www.hdfgroup.org/hdf-java-html/JNI/>.

#### 3.1.1 H5.java

The Java code, also referred to as the Java wrapper class, is a list of Java methods that are one-to-one mappings to the HDF5 C APIs.

The code in Figure 2 shows part of the Java wrapper method H5.H5Dopen() as an example. This is the Java method used to call the HDF5 library routine H5Dopen(hid\_t loc\_id, const char \*name). The Java keyword “*native*” indicates that the method is implemented in a language other than Java.

```
public class H5 {  
...  
    public static native int H5Dopen(int loc_id, String name)  
        throws HDF5LibraryException, NullPointerException;  
...  
}
```

Figure 2. Java wrapper method

### 3.1.2 h5\*Imp.c

A series of files named h5\*Imp.c contains the C routines that implement the Java wrapper methods. These native C routines are invisible to Java applications and should not be called directly. Java applications must use the Java wrapper class, H5, to call the HDF5 library.

## 3.2 Use

The Java class ***ncsa.hdf.hdf5lib.H5*** is the central part of the JHI5 package. The methods in this class call the standard HDF5 C library. Using the H5 class, you can make HDF5 library calls from Java code.

The JHI5 package includes methods for *most* of the HDF5 functions. Some HDF5 functions that are rarely used or that have C-function pointer parameter types are also not available in JHI5.

The JHI5 methods have function signatures that are similar to the corresponding HDF5 library functions.

The example code in Figure 3 uses the JHI5 H5 class methods to create an HDF5 file and dataset. The Java Products distribution contains additional examples under the directory “src/hdf-java/examples”.

```

// Creating a file and dataset.

import ncsa.hdf.hdf5lib.H5;
import ncsa.hdf.hdf5lib.HDF5Constants;

public class H5_CreateDataset
{
    private static void CreateDataset() throws Exception
    {
        int fid=-1, sid=-1, did=-1;
        long[] dims = { 4, 6 };

        // Create a new file using default properties.
        fid = H5.H5Fcreate("dset.h5", HDF5Constants.H5F_ACC_TRUNC,
            HDF5Constants.H5P_DEFAULT, HDF5Constants.H5P_DEFAULT);

        // Create the data space for the dataset.
        sid = H5.H5Screate_simple(2, dims, null);

        // Create the dataset.
        if ((fid >= 0) && (sid >= 0))
            did = H5.H5Dcreate(fid, "/dset", HDF5Constants.H5T_STD_I32BE, sid,
                HDF5Constants.H5P_DEFAULT);

        // End access to the dataset and release resources used by it.
        if (did >= 0)
            H5.H5Dclose(did);

        // Terminate access to the data space.
        if (sid >= 0)
            H5.H5Sclose(sid);

        // Close the file.
        if (fid >= 0)
            H5.H5Fclose(fid);
    }

    public static void main(String[] args)
    {
        try {
            H5_CreateDataset.CreateDataset();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

**Figure 3. JHI5 H5 class methods**

To compile and run the example you need the JDK (1.6 or above) and the JHI5 binaries. If your system does not have JDK installed, download and install JDK from <http://java.sun.com/javase/downloads>. The JHI5 binaries are included in your HDF Java Products distribution.

The batch script shown in Figure 4 provides an example to use when building JHI5 Java programs on Windows.

```

@ECHO OFF

@REM the name of example class
SET EXAMPLE=H5_CreateDataset

@REM the JDK directory
SET JAVAHOME=C:\java\jdk.1.6.0

@REM the hdf-java binary release
SET HDFJAVA=C:\hdf-java

@REM #####
@REM # DO NOT MODIFY BELOW THIS LINE UNLESS YOU KNOW WHAT WANT TO DO
@REM #####

@REM the directory that contains the HDF5 dynamic link library, jhdf5.dll
SET PATH="%HDFJAVA%\lib\win;

@REM the class path that includes the JHI5 package
SET CLASSPATH=.;%HDFJAVA%\lib\jhdf5.jar

@REM compile the source code
%JAVAHOME%\bin\javac %EXAMPLE%.java

@REM run the example program
"%JAVAHOME%\bin\java" -Djava.library.path=%PATH% %EXAMPLE%

PAUSE

```

Figure 4. Batch script for building JHI5 on Windows

The shell script in Figure 5 provides an example for building JHI5 Java programs on machines running Unix-like operating systems.

```

#!/bin/sh

# the name of example class
export EXAMPLE=H5_CreateDataset

# the JDK directory
export JAVAHOME=/usr/java/jdk1.6.0_16

# the hdf-java binary release
export HDFJAVA=/mnt/hdf-java/linux32

# OS name: use "linux, macosx, or solaris" only
export OSNAME=linux

#####
# DO NOT MODIFY BELOW THIS LINE UNLESS YOU KNOW WHAT WANT TO DO
#####

# the directory that contains the HDF5 dynamic link library
export LD_LIBRARY_PATH=$HDFJAVA/lib/$OSNAME

# the class path that includes the JHI5 package
export CLASSPATH=.:$HDFJAVA/lib/jhdf5.jar

# compile the source code
$JAVAHOME/bin/javac $EXAMPLE.java

# run the example program
$JAVAHOME/bin/java -Djava.library.path=$LD_LIBRARY_PATH $EXAMPLE

```

Figure 5. Script for building JHI5 on Unix-like systems

### 3.3 Considerations

JHI5 gives Java applications maximum flexibility in working with HDF5 files. The flexibility in JHI5 carries with it a burden of complexity.

The code in Figure 6 opens an HDF5 file, reads the contents of a dataset whose dimensions and data type are not known in advance, and closes the file. The number of H5 methods used can be burdensome for the application developer, especially when doing “typical” operations that do not need HDF5’s advanced features. Operations such as writing the contents of a compound dataset require even more function calls.

```
int fid, did, sid, tid, rank;
long dims[], maxDims[];
Object theData;

fid = H5.H5Fopen( fname, HDF5Constants.H5F_ACC_RDWR, HDF5Constants.H5P_DEFAULT);
did = H5.H5Dopen(fid, dname);
sid = H5.H5Dget_space(did);
tid = H5.H5Dget_type(did);
rank = H5.H5Sget_simple_extent_ndims(sid);

dims = new long[rank];
maxDims = new long[rank];
H5.H5Sget_simple_extent_dims(sid, dims, maxDims);
theData = allocateArray(tid, dims);

H5.H5Dread( did, tid, HDF5Constants.H5S_ALL, HDF5Constants.H5S_ALL,
            HDF5Constants.H5P_DEFAULT, theData);

H5.H5Sclose(sid);
H5.H5Tclose(tid);
H5.H5Dclose(did);
H5.H5Fclose(fid);
```

**Figure 6. An example of code that performs multiple operations**

For these reasons, and to program in a more object-oriented manner, many developers make use of the HDF Object Package, which is described in the next section.

## 4 HDF Object Package

### 4.1 Description

The HDF Object Package is a Java package that provides an object-oriented interface to HDF data objects. The package offers a common API for simple access to HDF files.

The HDF Object Package, `ncsa.hdf.object`, provides classes that reflect the fundamental data objects of the HDF formats. For more information, please visit <http://www.hdfgroup.org/hdf-java/html/hdf-object/index.html>.

For application developers, the HDF Object Package offers three major benefits:

- Fewer method calls
- Easy to extend for specific needs
- Object-oriented interface to HDF data

## 4.2 Use

The HDF Object Package can be used by Java applications to access HDF5 files without directly calling the HDF5 library via JHI5. It encapsulates HDF5 library calls into classes that reflect the HDF5 data objects. A Java application calls the Object Package, and the Object Package calls JHI5 on behalf of the application.

The code in Figure 7 uses the HDF5 Object Package to create an HDF5 file and dataset. This code can be compared to the example code in section 3.2, which used the JHI5 package.

```
import ncsa.hdf.object.*;
import ncsa.hdf.object.h5.*;
import javax.swing.tree.DefaultMutableTreeNode;

public class H5Obj_CreateDataset
{
    public static void main( String args[] ) throws Exception
    {
        long[] dims = { 4, 6 };

        // create a new file
        H5File h5File = new H5File ("dset_obj.h5", H5File.CREATE);

        // open the file and retrieve the root group
        h5File.open();
        Group root = (Group)(( DefaultMutableTreeNode)h5File.getRootNode()).getUserObject();

        // create a datatype
        Datatype dtype = h5File.createDatatype(
            Datatype.CLASS_INTEGER, 4, Datatype.NATIVE, Datatype.NATIVE);

        // create a dataset
        Dataset dataset = h5File.createScalarDS
            ("dset", root, dtype, dims, null, null, 0, null);

        // close the file
        h5File.close();
    }
}
```

**Figure 7. Code using the HDF5 Object Package**

For more instructions and examples on the use of the HDF5 Object Package, please visit <http://www.hdfgroup.org/hdf-java-html/hdf-object/use.html>.

The batch file in Figure 8 provides an example for building Java programs that use the HDF Object Package on Windows. The batch file is almost the same as in section 3.2, except that it adds two more JAR files to the class path: jhdfobj.jar and jhdf5obj.jar.



```

@ECHO OFF

@REM the name of example class
SET EXAMPLE=H5Obj_CreateDataset

@REM the JDK directory
SET JAVAHOME=C:\Java\jdk1.6.0_16

@REM the hdf-java binary release
SET HDFJAVA=C:\hdf-java

@REM the directory that contains the HDF5 dynamic link library, jhdf5.dll
SET PATH="%HDFJAVA%\lib\win;

@REM the class path that includes the JHI5 package
SET CLASSPATH=.;%HDFJAVA%\lib\jhdf5.jar;%HDFJAVA%\lib\jhdfobj.jar;%HDFJAVA%\lib\jhdf5obj.jar

@REM compile the source code
%JAVAHOME%\bin\javac %EXAMPLE%.java

@REM run the example program
"%JAVAHOME%\bin\java" -Djava.library.path=%PATH% %EXAMPLE%

PAUSE

```

**Figure 8. Batch file for building Java programs using the HDF Object Package on Windows**

Figure 9 shows an example of a similar shell script for building Java programs that use the HDF Object Package on machines running Unix-based operating systems.

```

#!/bin/sh

# the name of example class
export EXAMPLE=H5Obj_CreateDataset

# the JDK directory
export JAVAHOME=/usr/java/jdk1.6.0_16

# the hdf-java binary release
export HDFJAVA=/mnt/hdf-java/linux32

# OS name: use "linux, macosx, or solaris" only
export OSNAME=linux

#####
# DO NOT MODIFY BELOW THIS LINE UNLESS YOU KNOW WHAT WANT TO DO
#####

# the directory that contains the HDF5 dynamic link library
export LD_LIBRARY_PATH=$HDFJAVA/lib/$OSNAME

# the class path that includes the JHI5 package
export CLASSPATH=.:$HDFJAVA/lib/jhdf5.jar: $HDFJAVA/lib/jhdfobj.jar:$HDFJAVA/lib/jhdf5obj.jar

# compile the source code
$JAVAHOME/bin/javac $EXAMPLE.java

# run the example program
$JAVAHOME/bin/java -Djava.library.path=$LD_LIBRARY_PATH $EXAMPLE

```

**Figure 9. Script for building Java programs using the HDF Object Package on Unix-based systems**

A Java application can make calls to both the HDF Object Package and to the JHI5 methods, relying on the JHI5 methods when more flexibility and control is needed. This advanced usage model can lead to unexpected consequences unless care is taken.

## 5 HDFView

### 5.1 Description

HDFView is a visual tool for browsing and editing HDF4 and HDF5 files. It provides an easy-to-use interface to create and explore HDF5 files. HDFview is frequently used to experiment with different data object structures and to inspect data values in HDF5 datasets.

HDFView allows you to:

- View a file hierarchy in a tree structure
- Create a new file
- Add or delete groups
- Add or delete datasets
- View and modify the content of a dataset
- Add, delete and modify attributes
- Replace I/O and GUI components such as table view, image view, and metadata view

Figure 10 shows a screenshot of HDFView. For more information, please visit <http://www.hdfgroup.org/hdf-java/html/hdfview/>.

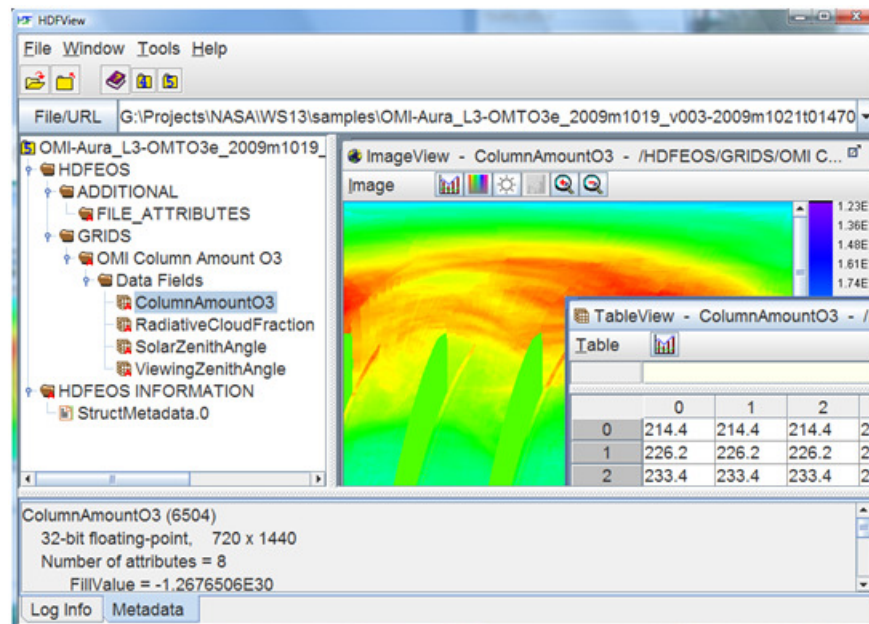


Figure 10. Screenshot of HDFView

## 5.2 Use

The following brief instructions explain how to run HDFView on various platforms. For more information on how to use HDFView, please see the *HDFView User's Guide* at

<http://www.hdfgroup.org/hdf-java-html/hdfview/UsersGuide/index.html>.

## 5.3 Windows

If you used the HDFView installer in section 2.2, you can run HDFView by selecting it from the Programs menu:

Start -> All Programs -> HDFView x.x -> HDFView

If you installed from the prebuilt binaries in section 2.2, you can run HDFView by double-clicking the file "hdfview.bat" in your "*HDFVIEW\_HOME/bin*" directory.

## 5.4 Unix

If you used the HDFView installer in section 2.2, you can run HDFView with the following command:

```
$ HDFVIEW_HOME/hdfview
```

If you installed from the prebuilt binaries in section 2.2, you can run HDFView from the shell script file "hdfview.sh" in your "*HDFVIEW\_HOME/bin*" directory:

```
$ HDFVIEW_HOME/bin/hdfview.sh
```

## 5.5 Mac

If you used the HDFView installer in section 2.2, you can run HDFView by double-clicking the HDFView icon on your desktop or running the following command:

```
$ open $ HDFVIEW_HOME /HDFView.app
```

If you installed from the prebuilt binaries in section 2.2, you can run HDFView from the shell script file "hdfview.sh" in your "*HDFVIEW\_HOME/bin*" directory:

```
$ HDFVIEW_HOME/bin/hdfview.sh
```