RFC: Automated Testing for HDFView

Peter Cao
Allen Byrne

This document recommends a GUI testing framework, FEST Swing, for automated testing for HDFView. This document summaries the study we have done for the automated testing and estimates the work that needs to be done.

1 Introduction

The HDF libraries have a large collection of unit tests that assure confidence in their use. HDFJava libraries also have a large collection of unit tests that test the JNI interface and object libraries. These unit tests can be automatically executed during daily test jobs, and the results verified as correct.

However the HDFView program has only used a manual checklist (Appendix A: HDFView Test Checklist) to perform an acceptance test, usually before a release. There are tools which can automate not only the acceptance test, but also provide unit testing of HDFView's visual components and interactions.

A search of active open source gui unit testing frameworks indicated that the FEST framework would be well suited to test the HDFView program because of the current use of the JUnit4 framework in the JNI unit tests. Article references to the FEST framework has been found at a number of reputable sites (e.g. IBM Developer Works, IEEE Explore). It is actively being developed and expanded. There were a few others considered, which further research indicated they did not have the same level of active support. For details, visit the FEST site: http://fest.easytesting.org/swing/wiki/pmwiki.php (see Publications link) and IBM DW site: http://www.ibm.com/developerworks/java/library/j-swingtest/.

The FEST framework looks promising for our purpose. We have tested it with a simple case, checking the “HDF4” and “HDF5” library versions and more complicated cases, creating a new file and a new dataset. The framework does not require comprehensive changes to the current code. As the result of our initial study on FEST, we are presenting this RFC for more inputs before we go for the full implementation of the automated testing.

2 GUI Testing Approaches

There are two methods for testing a JAVA visual program, record/playback tools and unit testing frameworks. The record/playback tools are typically classified as "black box" testing. Record/playback tools record a user performing actions following a checklist of acceptance tests. The unit testing frameworks are "white box" testing because they require knowledge of how the visual components are constructed.

Both methods are useful test techniques and are usually complementary tools. The record/playback tools are best used by a person performing acceptance testing, while unit testing is better suited to unattended automated testing.

2.1 Record/playback

Acceptance tests, using the record/playback technique, allow a tester to execute a number of prerecorded scripts that exercise the program under test and verify that the expected results occur. A developer would record the scripts by starting the application and performing a sequence of steps that correspond to a previously developed scenario. The developer should document the expected states of each scenario, such that the testers can verify that the results they see match what is expected.

The typical record/playback tool is dependent upon the positioning of the screen elements being constant. If a button or dialog box is moved or replaced, the playback script would fail and require that this script and possibly others be rerecorded.

2.2 Unit testing

Swing is one of the more powerful GUI toolkits available; it's extensible, configurable, and cross-platform. But Swing's flexibility is both its major strength and a great weakness. With Swing, you can construct the same UI in many different ways. For example, you can use insets, empty borders, or fillers to put space between GUI components. Given Swing's extensive stock of options, understanding an existing GUI can be as daunting a task as writing a new one, and mapping its visual appearance to the underlying code is far from trivial.

As a developer, you explore the internals of the code while writing the tests. As a valuable side effect, you end up with a test suite that can help prevent the introduction of regressions when you maintain the code.

A good start is to write functional tests to understand how the GUI behaves in response to user input. Writing tests for GUIs is more complex than writing tests for non-visual code, because:

- Ideally, tests must be automated, but GUIs are designed for humans — not computer programs — to use.
- Conventional unit testing, involving tests of isolated classes, is unsuitable for GUI components. In GUI terms, a "unit" involves cooperation of more than one GUI component, which can itself consist of more than one class.
- GUIs respond to user-generated events. To test GUIs, you need a way to simulate user input, wait until the generated events have been broadcast to all listeners, and then check the result as the GUI would appear to the user. Writing code that simulates user interaction with GUIs can be tedious and error-prone.
- Changes in the GUI's layout should not affect robust functional tests.
- An additional issue is that you must already know the structure and behavior of the GUI you want to test, otherwise you don't know which components the automated test should use and what needs to be verified. In general, to write a GUI test you must know:
    - The components that are present in the GUI to test

- How you can uniquely identify such components in your tests
- The expected state (or properties) of the components in a particular use case

3 Testing HDFView with FEST

The FEST test framework is supplied as a compressed archive, which includes the jar files needed. Other extensions for different types of testing are available also. Exposure to the JUnit4 framework made FEST easy to integrate and create a quick test of the main window and buttons of HDFView. The first execution of the tests exposed a common error of SWING programming; SWING interaction must be performed in the Event Dispatch Thread, which HDFView failed. The proper fix was implemented as follows:

ORIGINAL

```
HDFView frame = new HDFView(rootDir, flist, W, H, X, Y);
frame.setVisible(true);
```

CORRECTED

```
final Vector the_flist = flist;
final String the_rootDir = rootDir;
final int the_X=X, the_Y=Y, the_W=W, the_H=H;
//Schedule a job for the event-dispatching thread:
//creating and showing this application's GUI.
javax.swing.SwingUtilities.invokeLater(new Runnable() {
  public void run() {
    HDFView frame = new HDFView(the_rootDir, the_flist, the_W,      the_H, the_X,
the_Y);
    frame.setVisible(true);
  }
});
```

The next test run uncovered a requirement of gui unit testing that was not supplied by HDFView; components which are targeted for testing must be uniquely identifiable to the test framework. A JAVA Swing program is usually built with a few components (Panel, Frame, Pane, etc ...) combined in multiple ways to get a visual result. The uniquely identifiable requirement is usually accomplished by

assigning the components an internal name at creation time. A quick fix of the components involved (Toolbar buttons) with the initial test suite allowed the test to run and complete successfully.

ORIGINAL

```
// open file button
JButton button = new JButton(ViewProperties.getFileopenIcon() );
tbar.add( button );
button.setToolTipText( "Open" );
button.addActionListener( this );
button.setActionCommand( "Open file" );
// close file button
button = new JButton(ViewProperties.getFilecloseIcon() );
tbar.add( button );
button.setToolTipText( "Close" );
button.addActionListener( this );
button.setActionCommand( "Close file" );
```

CORRECTED

```
// open file button
JButton button = new JButton(ViewProperties.getFileopenIcon() );
tbar.add( button );
button.setName("Open");
button.setToolTipText( "Open" );
button.addActionListener( this );
button.setActionCommand( "Open file" );
// close file button
button = new JButton(ViewProperties.getFilecloseIcon() );
tbar.add( button );
button.setName("Close");
```

```
button.setToolTipText( "Close" );
button.addActionListener( this );
button.setActionCommand( "Close file" );
```

The seven tests verified that the five toolbar buttons were enabled and the 'HDF4 Library' and 'HDF5 Library' buttons were activated and the test verified that correct message dialog appeared.

A more involved test to open a HDF file and interrogate a dataset will be developed and all the components involved will be updated to provide the needed identification. A test plan will be developed to identify the most critical functionality that should be tested. Then the HDFView source should be updated to provide the needed identification for the components involved. Once that is implemented and running during daily tests, less critical but necessary functionality can be added to the test suite. Future development should be required to use the FEST framework in a test driven development process.

4 Testing HDFView with JACARETO

In addition to the implementation of the unit testing framework, an acceptance testing plan, based on the existing test guide, will be developed and executed on a regular basis. Most likely the acceptance tests should be used after the implementation of a significant functionality and before any release. While an acceptance test can be automated with a record/playback tool, it will require a human to monitor and interpret the results.

The JACARETO record/playback test tool is a suitable tool for record/playback testing and is supplied as a compressed archive, which includes the jar files needed. The creation of a test script, based on the hdfview script, is the preferred solution to execute this tool. This script would set up the correct classpaths and parameters needed for both the test tool and hdfview. Once the test scripts are recorded and tested, these scripts can be included within a compressed archive that can be used by testers. We will need to identify the record/playback scripts from the existing test document (see Appendix A). We will need to identify by name which scripts should be executed and the expected results.

The creation of these record/playback scripts should take about 5X the normal time for a tester to perform just the test. This estimate assumes that the script must be recorded, playbacked on the target test machines, and documented. This does not take into account exceptions and debugging that might happen.

JACARETO was chosen because it is being actively developed, is usable on Windows, Linux and Mac machines, and is open source. Other tools can be considered in the future, once we have enough experience with this technique to properly evaluate a tool's capabilities.

5 Implementation Steps

Appendix A is a list of GUI components/actions for manual testing. There are over 250 items in the list. The list is not complete but it covers all the major features in HDFView. Each item should be reviewed

for record/playback suitability. Our finial goal is to implement GUI unit tests for all the items in the list. To accomplish this goal, we will take four steps:

A. Learning and researching: identifying tools for automatic GUI testing and testing feasibility of the tools on HDFView. The result will be reported in a RFC (this RFC)

B. Prototype implementation: completing GUI test for creating new HDF5 datasets (section 10.7 in the test list). The prototype will provide better estimation on the total work of the automatic GUI testing

C. Implementing items with high priorities: completing items with high priorities

D. Full implementation

   a. Completing all the items in the GUI unit tests

   b. Adding record/playback scripts

6 Project Plan for Prototype Implementation

The prototype implementation will focus on creating new HDF5 datasets. The goals of the prototype include having a more comprehensive understanding of the FEST test framework, providing details on how tests can be implemented, and giving a better estimation of the total work of the whole work.

The record/playback is not included in the prototype.

Task (details may vary as needed)	Work (hours)	Who	Start	Finish	Deliverable
Check and review deliverable	6	Peter	3/1/11	3/31/11	Status report
Modifying source for creating a file	4	Allen	3/1/11	3/2/11	Code
Creating a test file	4	Allen	3/3/11	3/4/11	Code
Modifying source for creating dataset	4	Allen	3/7/11	3/8/11	Code
Test case: int8, LE, 1D of 1000	4	Allen	3/9/11	3/10/11	Code
Test case: uint8, BE, 2D of 500x20	2	Allen	3/11/11	3/11/11	Code
Test case: int32, LE, 2D chunks of 50x20	2	Allen	3/14/11	3/14/11	Code
Test case: uint32, BE, level 5 gzip compression	2	Allen	3/15/11	3/15/11	Code
Test case: int64, LE, 2D chunks of 500x20	2	Allen	3/16/11	3/16/11	Code
Test case: float32, compressed 1D of 1000	2	Allen	3/17/11	3/17/11	Code
Test case: float64, 2D of 8000x8000	2	Allen	3/18/11	3/18/11	Code
Test case: 1D strings of length 80	2	Allen	3/21/11	3/21/11	Code
Test case: compressed 2D strings of length 80	2	Allen	3/22/11	3/22/11	Code
Test case: 1D references of size 2	4	Allen	3/23/11	3/24/11	Code
Test case: 30x20 2D enum of RGB colors	4	Allen	3/25/11	3/28/11	Code

Integrating to daily test	4	Allen	3/29/11	3/30/11	Code
Reporting and time estimation	6	Peter	3/30/11	3/31/11	Final report
Total (Peter 12, Allen 40)	56				

Revision History

July 27, 2010:	Version 1, first draft, circulated for comment within The HDF Group.
August 31, 2010	*Version 2, added gui functional testing information.*
December 17, 2010	*Version 3, added record/playback information.*

Appendix A: HDFView Test Checklist

Task #	Priority	Task	What to check
1.		***Installation***	
1.1.	High	Installing JRE version	
1.2.	High	Installing non-JRE version	
2.		***Starting HDFView***	
2.1.	Medium	from defaults	
2.2.	Medium	using –g 1200x800+100+200	The window size and offset
2.3.	Medium	with multiple files	If all the files show in the Tree
3.		***HDFView Tool Bar***	
3.1.	High	Open file icon	If the file popup window starts at default working Directory
3.2.	High	Close file icon	
3.3.	High	Help icon	If it points to the correct UG
3.4.	High	H4 version icon	If the HDF4 library if correct
3.5.	High	H5 version icon	If the HDF5 library if correct
3.6.	High	File/URL	Open file from a list of current files
4.		***HDFView Tabs***	
4.1.	High	Log Info	The bottom of the HDFView. General info.
4.2.	High	Metadata	Single click on datasets/groups to see if attributes and other metadata show here
5.		***HDFView File Menu***	**Test HDF5 file**

5.1.	High	Open	With Read/Write permission by default
5.2.	High	Open Read-Only	No modification to the file
5.3.	High	Close	Close single file
5.4.	High	Close All	Close all files
5.5.	High	Save	Save to the same file
5.6.	High	Save As	Save to a new file. Open the new file
5.7.	High	New→HDF4	Create an empty HDF4 file
5.8.	High	New→HDF5	Create an empty HDF5 file
6.		***HDFView Window Menu***	
6.1.	Medium	Cascade	Open a few datasets/images before test these menus
6.2.	Medium	Tile	
6.3.	Medium	Close	Close current table/image window
6.4.	Medium	Close All	Close all open tables/images
7.		***HDFView Tools Menu***	
7.1.	Medium	Convert JPEG→HDF4	If the HDF4 image is correct
7.2.	Medium	Convert JPEG→HDF5	If the HDF5 image is correct
7.3.	Medium	Register File Format	If a new file is in the supported list, e.g., FITS:ncsa.hdf.object.fits.FitsFile:fits
7.4.	Medium	Un-register File Format	If a file format is removed from the list
8.		***HDFView User Options***	
8.1.	Medium	Default WD to CWD	Restart HDFView to see if “file open “ popup starts at the CWD
8.2.	Medium	Default WD to other	Restart HDFView to see if “file open

			" popup starts at that directory, e.g. G:\temp
8.3.	Medium	User's Guide path	Test online and local copy
8.4.	Medium	Set font	Font size and type. Restart to see if the font change is saved
8.5.	Medium	Data delimiter	Test all five cases with read/write operations
8.6.	Medium	Max number to load	Set to 5 to see if it works. Set is back to 10,000 after testing is done. Try files with 20,000+ objs.
9.		***HDFView Help Menu***	
9.1.	High	User's Guide	
9.2.	High	H4 version icon	If the HDF4 library if correct
9.3.	High	H5 version icon	If the HDF5 library if correct
9.4.	High	Java Version	
9.5.	High	Supported File Formats	
9.6.	High	About…	If the HDFView version if correct
10.		***TreeView Popup Menu***	(right-mouse click on dataset/group)
10.1.	High	Double click on dataset/image or "Open" from Popup	If the content of dataset/image shows
10.2.	High	**"Open As" on an image**	
10.2.1.	High	Mouse Drag preview image to select a subset	If the content is correct
10.2.2.	High	Test different combinations of start/stride/count	If the content is correct
10.2.3.	High	Change dimension orders	If the content id correct
10.2.4.	High	Test different palettes	If the image is correct

10.2.5.	High	Display as Spreadsheet	If the content is correct
10.2.6.	High	Default display type	1.'Open as' on a image 2. Switch 'display as' between 'Spreadsheet' and 'Image' If the previous 'display as' choice is remembered , when 'Open as' again
10.2.7.	High	Large image	If it works for 8kx8k image. How long to load it.
10.2.8.	High	Move a large image	Mouse drag to move a large image
10.2.9.	High	Select part of image	(Shift+Mouse_drag) to select part of image. Can select an area as red box appears?
10.3.	High	**"Open As" on a dataset**	
10.3.1.	High	Drag preview image to select a subset	If the content is correct
10.3.2.	High	Test different combinations of start/stride/count	If the content is correct
10.3.3.	High	Change dimension orders	If the content id correct
10.3.4.	High	Display an integer dataset as image, try autogain/non	If the content is correct
10.3.5.	High	Display a float dataset as image, try autogain/non	If the content is correct
10.3.6.	High	Default display type	If the previous display type is remembered
10.3.7.	High	Large dataset	If it works for 8kx8k dataset. How long to load it.
10.4.	High	**"Open As" on Compound**	
10.4.1.	High	Test different combinations of start/stride/count	If the content is correct

10.4.2.	High	Change dimension orders	If the content id correct
10.4.3.	High	Select a single member	If the content is correct
10.4.4.	High	Select odd-members	If the content is correct
10.4.5.	High	Large dataset	If it works for 1M rows x 20 columns table. How long to load it.
10.5.	High	**"Open As" on Text**	
10.5.1.	High	Test different combinations of start/stride/count	If the content is correct
10.6.	High	**"New"→"Group"**	
10.6.1.	High	New group at the root	Close file; re-open file. Open group. Are members correct? Is group metadata correct?
10.6.2.	High	New group at other groups	Same as above
10.7.	High	**"New"→"Dataset"**	
10.7.1.	High	Different datatype class	Close file; re-open file. Open dataset as spreadsheet. Are values correct? Are datatypes correct?
10.7.2.	High	Different datatype size	Same idea as 10.7.1.
10.7.3.	High	Different datatype order	Etc.
10.7.4.	High	Different datatype sign	
10.7.5.	High	Different rank	
10.7.6.	High	Different dimension sizes	
10.7.7.	High	Different max sizes	
10.7.8.	High	Different chunking sizes	
10.7.9.	High	Different compression levels	
10.7.10.	High	Large dataset	8kx8k

10.8.	High	**"New"→"Image"**	Need to close/reopen file after image is created
10.8.1.	High	Different image size	
10.8.2.	High	Indexed image	
10.8.3.	High	24-bit true color	
10.8.4.	High	Pixel interlace	
10.8.5.	High	Plane interlace	
10.9.	High	**"New"→"Table"**	
10.9.1.	High	Different table size	
10.9.2.	High	Different chunking sizes	
10.9.3.	High	Different compression levels	
10.9.4.	High	Different types of members	
10.9.5.	High	Import member types	
10.10.	High	**"New"→"Datatype"**	
10.10.1.	High	Different datatype class	
10.10.2.	High	Different datatype size	
10.10.3.	High	Different datatype order	
10.10.4.	High	Different datatype sign	
10.11.	High	**"New"→"Link"**	
10.11.1.	High	Link to dataset	
10.11.2.	High	Link to group	
10.12.	High	Copy/Paste datasets in the same file	Re-open the destination file to see if the objects are copied. Are their attributes and other characteristics the same? Is content the same?

			Can I view them as image and spreadsheet?
10.13.	High	Copy/Paste datasets cross files	Re-open the destination file to see if the objects are copied. Ditto, etc.
10.14.	High	Copt/Paste group in the same file	Re-open the destination file to see if the objects are copied.
10.15.	High	Copy/Paste group cross files	Re-open the destination file to see if the objects are copied.
10.16.	High	Delete datasets	Re-open the destination file to see if the objects are removed
10.17.	High	Delete groups	Re-open the destination file to see if the objects are removed
10.18.	High	“Save To” on dataset	If the object is saved to a new file
10.19.	High	“Save To” on image	If the object is saved to a new file
10.20.	High	“Save To” on group	If the object is saved to a new file
10.21.	High	“Rename” on dataset	Re-open the file to see if the object is renamed
10.22.	High	“Rename” on image	Re-open the file to see if the object is renamed
10.23.	High	“Rename” on group	Re-open the file to see if the object is renamed
10.24.	High	“Show Properties” on groups	Check both attributes and general information
10.25.	High	“Show Properties” on datasets	Check both attributes and general information
10.26.	High	“Close File”	Close the file and all the open objects associated to the file
11.		***TableView***	
11.1.	High	Open different datasets (floats, integers,	If the content is correct

		compound)	
11.2.	Medium	**Line plot**	"Line Plot" icon or "Table"→"Show Lineplot" Exist?
11.2.1.	Medium	Line plot by rows	1)Select adjacent rows. 2)Select first and last rows. 3) Select non-adjacent rows in the middle. (use ctrl or shift key) Are all of the plots correct? (Check X and Y axis labels. Check shape of plot, Colors, number.)
11.2.2.	Medium	Line plot by columns	Same idea as above with columns.
11.2.3.	Medium	Line plot with different X-values	Same idea.
11.3.	Medium	Export Data to text File	'Table -> Export Data to File' and save to a file. Check the content of the text file
11.4.	Medium	Import Data from text File	1) Modify the above exported text file. 2) 'Table -> Import Data from File' and open the modified text file Check if the modification is correct, also unmodified values are correct?
11.5.	Medium	Import (?) Fixed Data Length	Set different fixed length and import data from text file
11.6.	Medium	Copy&Paste data values	1) Perform copy and paste in the same dataset Check if it worked? 2)Perform copy and paste to a dataset in different file. (make a copy of the file and test) If the destination file has the new

			data?
11.7.	Medium	Copy to new Dataset	Copy selected data to an new dataset in the same file. re-open the file the verify the new dataset is created in file
11.8.	Medium	Update file	1) modify any value. 2) 'Table -> Update File' 3) close the file 4) Re-open the file Check if the modification is correct?
11.9.	Medium	Select All	'Table -> Select All' Check to see all cells are high-lighted
11.10.	Medium	Show Statistics	'Table -> Show Statistics' Verify the values of min, max, mean and std.
11.11.	Medium	Math Conversion	'Table->Math Conversion' Verify that all math conversions are correct
11.12.	Medium	Scientific Notation	Show data in "0.0###E0#"
11.13.	Medium	**Frame selection**	3D datasets
11.13.1.	Medium	"Next"	
11.13.2.	Medium	"Previous"	
11.13.3.	Medium	"First"	
11.13.4.	Medium	"Last"	
11.13.5.	Medium	Enter frame number	
11.14.	Medium	Close table	
11.15.	Medium	TableView for a Compound dataset	Test for a compound dataset Open 'samples/TestH5Object.h5' , there is 'comp_dataset' object to

			test with.
11.15.1.	Medium	Line plot by rows	1)Select adjacent rows. 2)Select first and last rows. 3) Select non-adjacent rows in the middle. (use ctrl or shift key) Are all of the plots correct? (Check X and Y axis labels. Check shape of plot, Colors, number.)
11.15.2.	Medium	Line plot by columns	Same idea as above with columns.
11.15.3.	Medium	Line plot with different X-values	Same idea
11.16.	Medium	Export Data to text File	'Table -> Export Data to File' and save to a file. Check the content of the text file
11.17.	Medium	Import Data from text File	1) Modify the above exported text file. 2) 'Table -> Import Data from File' and open the modified text file Check if the modification is correct, also unmodified values are correct?
11.18.	Medium	Copy&Paste data values	1) Perform copy and paste in the same dataset Check if it worked? 2)Perform copy and paste to a dataset in different file. (make a copy of the file and test) If the destination file has the new data?
11.19.	Medium	Update file	1) modify any value. 2) 'Table -> Update File' 3) close the file 4) Re-open the file Check if the modification is correct?

11.20.	Medium	Select All	'Table -> Select All' Check to see all cells are high-lighted
11.21.	Medium	Show Statistics	'Table -> Show Statistics' Verify the values of min, max, mean and std.
11.22.	Medium	Math Conversion	'Table->Math Conversion' Verify that all math conversions are correct
12.		***ImageView***	
12.1.	High	Open a test file	'Open' 'samples/hdf5_test.h5' file under hdf-java dir. Open without fail?
12.2.	High	Drag mouse across an image	Open 'images/hst_lagoon_detail.jpg' object. Make the view window small so the image cannot fit. Drag the mouse on the image to see different part of the image. Do the correct image appear along dragging?
12.3.	High	Select an image area	Open 'images/iceberg' image. (SHIFT + mouse_drag) to select part of an open image Selected with red rectangle?
12.4.	Medium	Show Histogram	Click on "Histogram" icon or Image→Show Histogram. Select different part and see if Histogram reacts to it.
12.5.	Medium	Show Palette	Click "Palette" icon or

			Image→Change Palette See pop-up window 'image Palette for – xxx'?
12.5.1.	Medium	Change RED, GREEN, BLUE color line	
12.5.2.	Medium	Select different palette	Image color changes accordingly?
12.5.3.	Medium	Change palette value from "Show Value" table	Change 10 and 17 values to R:255 G:0 B:0 and OK. Do you see red dots on the image? (Zoom In to see better)
12.6.	Medium	Change Brightness/Contrast	Click on 'Briteness' icon or Image -> Brightness/Contrast. Check best Brightness/Contrast and extreme Brightness/Contrast Does the image change accordingly?
12.7.	Medium	Zoom in/out image	Clink on Zoom In or Zoom Out button. (Image -> Zoon In/Zoon Out) Zoom in and out Correctly?
12.8.	Medium	Save Image As JPEG	Image -> Save Image As -> JPEG Verify the jpeg file?
12.9.	Medium	Write Selection to Image	Select part of image and write it to a new image in the same file. Re-open the file to verify if the new image is written to file
12.10.	Medium	Set Value Range	Click Image-> Set Value Range. Change some value and see if the image changes accordingly? Expect to see color of the image changes.
12.11.	Medium	Horizontal Flip	Click Image->Flip->Horizontal
12.12.	Medium	Vertical Flip	Click Image->Flip→Vertical
12.13.	Medium	Rotate 90 CW	Click Image->Rotate->90 CW

12.14.	Medium	Rotate 90 CCW	Click Image->Rotate->90 CCW
12.15.	Medium	Contour	Try different contour levels
12.16.	Medium	Show Value	Check “Image->Show Value” and move the mouse on the image. At the bottom, x,y,value are showing. ‘Open As’ the same image and select ‘Spreadsheet’ and OK. Zoom-In the image to each pixel can be pointed. Compare some outline pixels with Spreadsheet number. Verify the image data values are correct
12.17.	Medium	Show Statistics	Click Image-> Show Statistics Verify the min, max, mean and std.
12.18.	Medium	Select all image	Click Image-> Select All Select the all area of the image?
12.19.	Medium	**Frame selection**	3D images Open ‘images/3D THG’ image.
12.19.1.	Medium	Traverse forward	Click Next arrow button
12.19.2.	Medium	Traverse backward	Click Previous arrow button
12.19.3.	Medium	View first image	Click First arrow butoon
12.19.4.	Medium	View last image	Click Last arrow button
12.19.5.	Medium	View a certain frame	Put frame number and hit enter key
12.20.	Medium	Animation	Click “Animation” icon for 3D images
12.21.	Medium	Animation speed	Image->Animation(fremas/second) Verify different speed.
12.22.	Medium	Close ImageView	
12.23.	Medium	Image on True Color image	Verify all above on 24-bit true color

			image
13.		***TextView***	
13.1.	Medium	Save data to text file	Create dataset with 'STRING' type, any length, 2dim, 10x5 -> OK. Open the dataset just created. Enter any string contents. Click 'Text -> Save To Text File' and save. Verify the content of the text file
13.2.	Medium	Modify and save text	Re-open the text dataset and modify contents. Click 'Text->Save Changes' Close and reopen the dataset. Verify the contents. Close entire file and reopen. verify the changes are saved in file
13.3.	Medium	Close TextView	Click 'Text->Close'. Can close?
14.		***Test HDF4 file***	Test all above for HDF4 file. Some of the features may not apply to HDF4
14.1.	Medium	TreeView	
14.2.	Medium	TableView	
14.3.	Medium	ImageView	
14.4.	Medium	TextView	
15.	Medium	***Test for Copy objects***	
15.1.	Medium	Copy a large dataset with contiguous layout,e.g. 3GB	
15.1.1	Medium	Without allocation	

15.1.2	Medium	With allocation, if data is written	
15.2.	Medium	Copy a chunked dataset	
15.2.1	Medium	Where storage in file is not allocated	
15.2.2	Medium	Where storage in file is partially allocated	
16.	Medium	***Test for data in 4Dimension Dataset***	
16.1.	Medium	Open a 4 dimension dataset with natural dimension order. Set start, end and stride.	
16.1.1	Medium	Check the value at a specific location [w] [x] [y] [z] in dataset	
	Medium		
16.2.	Medium	Open the dataset with the dimension order changed. Set start, end and stride values to what was set earlier, respective to the dimension order.	
16.2.1	Medium	Check the value at location [w] [x] [y] [z]	It should be the same as value obtained in 16.1.1
16.2.2	Medium	Open the dataset. Set start to a different value.	
16.2.3	Medium	Check the value at location [w] [x] [y] [z]	It should be the same as value obtained in 16.1.1
17.		***Plug-in modules***	
17.1.		HDF-EOS modules	***(download EOS modules)***

17.1.1.	Medium	Register file format	
17.1.2.	Medium	Un-register file format	
17.1.3.	Medium	Test TreeView	
17.1.4.	Medium	Test MetadataView	
17.1.5.	Medium	Test TableView	
17.1.6.	Medium	Test ImageView	
17.1.7.	Medium	Set default module	
17.1.8.	Medium	Set back default module	
17.2.		NetCDF file format	
17.2.1.	low	Register file format	
17.2.2.	low	Un-register file format	
17.2.3.	low	Test TreeView	
17.2.4.	low	Test MetadataView	
17.2.5.	low	Test TableView	
17.2.6.	low	Test ImageView	
17.2.7.	low	Set default module	
17.2.8.	low	Set back default module	